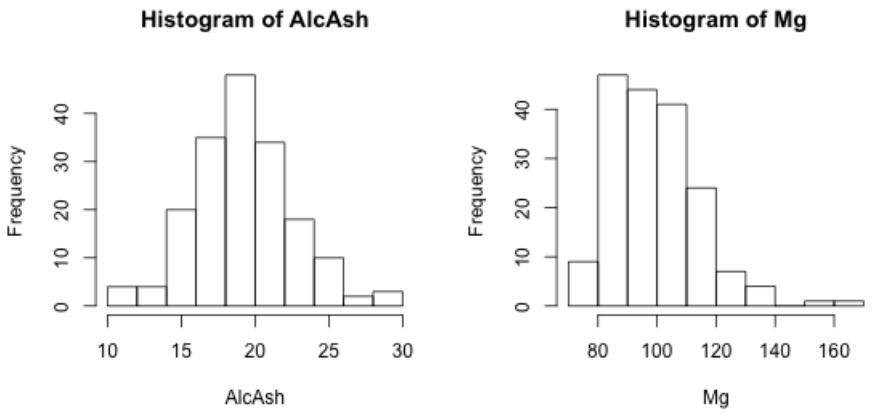
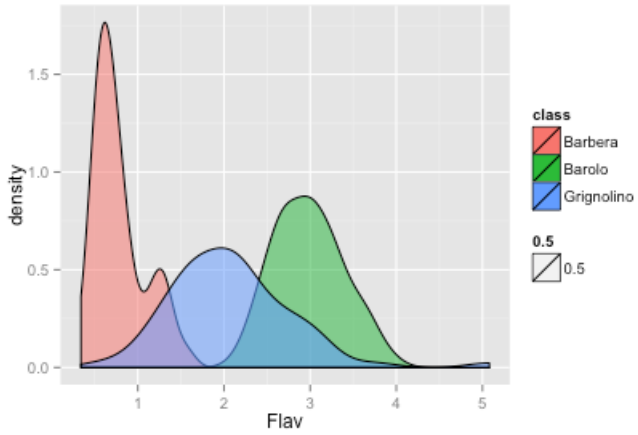
Assignment 3 Executive Summary

With the Wine dataset we will be applying a supervised learning to predict the class of the unknown wine. In this initial project we will perform an EDA on the dataset to understand the distributions of the variables with the context of classification in mind. Within our EDA we have two tasks: **univariate and bivariate analysis**.

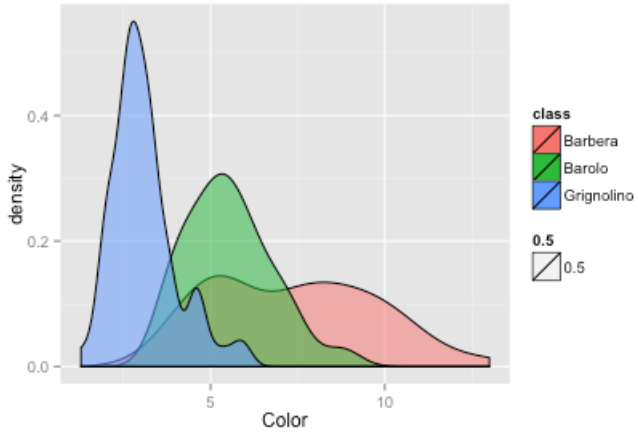
The univariate EDA was performed using histograms of each variable. The results show that there are some variables that are normally distributed, while others appear to be log-normal in distribution. For example, "AlcAsh" shows a normal distribution whereas "Mg" could potentially benefit from a log transformation if necessary:



The bivariate EDA was then performed to see the relationship between the explanatory and target variables. This was completed using Kernel Density Estimates, with each density estimate broken down by the target classes. The results were positive, with some variables showing good breakdown between the classes. For example "Flav" shows excellent separation between the "Barbera" and "Barolo" classes, as show below:



In addition, the “Color” variable could be used to separate the “Grignolino” class from the others:



In summary, the bivariate results show that an algorithm should be able to separate the three classes easily using this variable.

Based on these results, we feel comfortable moving on with the modeling process. The EDA has already shown some variables to be viable in a classification context. While there could be better separation between the “Grignolino” class and others, it may be worthwhile to apply classification algorithms and see what happens. For full writeup, visit http://jakechen.github.io/pred_412/programming1/code.html

Attached Code

````
1. Introduction
===============
With the Wine dataset we will be applying a supervised learning to
predict the class of the unknown wine. In this initial project we will
perform an EDA on the dataset to understand the distributions of the
variables with the context of classification in mind.

2. Procedure & Analysis
=======================
2.1 Data import
---------------
First we begin by loading the data.
```{r}
library(MMST)
data(wine)
```

2.2 Summary EDA
---------------
Let's begin the EDA process by performing a quick viewing and summary
reports on the entire dataset.
```{r}
head(wine)
str(wine)
summary(wine)
```
**Analysis**
What we see is quite straight-forward and expected. Here are some
important highlights we can gather about the data:
- Str() function:
  - There are 15 variables in the dataset
  - The last 2 variable are the categorical classifiers that we are
modeling for. R has already set them to the "factor" datatype for us.
  - There are
- Summary() function:
  - There are no missing values in our data.
  - The magnitudes in the variables vary from single digits to
thousands.
  - The dataset is relatively well distributed between the classes.

2.3 Graphical EDA
-----------------
Now that we have completed a cursory summary EDA of our dataset, we
move on with a graphical EDA on the datasets. This step of the EDA will
hopefully provide us a better understanding of the variables in terms
of their individual distributions as well as their mutual
relationships.

### 2.3.1 Univariate EDA
First let's perform a univariate EDA to understand the variables as a
whole. We can do this by **looking at a histogram of each variable**.
```{r, fig.width=4, fig.height=4}
for(i in 1:(ncol(wine)-2)){
  hist(wine[,i], xlab=names(wine)[i], main=paste('Histogram of',
names(wine)[i]))
````
````

```
}
```
Analysis
From this quick initial analysis we can see that there are **some variables that seem to be normally distributed as well as some variables that could be log-normally distributed**. For example, Alcohol, AlcAsh, and Proa seem to be relatively normally distributed. On the other hand, MalicAcid, Mg, and Color may benefit from a log transformation.

2.3.2 Bivariate EDA
With a univariate EDA completed, we move on to multivariate EDA to look at the relationship between variables. Because the purpose of this project is classification, we begin by looking at each explanatory variable in relation to the target variable. Instead of histograms, this time we will use **Kernel Density Estimation broken down by wine class**. We'll also adjust the opacity of the estimates to make the overlapping regions more visible. To create these graphics we will turn to the **ggplot2** package.
```{r, fig.width=6, fig.height=4}
library(ggplot2)
for(i in 1:(ncol(wine)-2)){
  p <- ggplot(wine, aes(x=wine[,i], fill=class)) +
    geom_density(aes(alpha=0.5)) +
    labs(x=names(wine)[i])
  print(p)
}
```
Analysis
Already we can see some great breakdowns in the data that could be useful in classifying wines. For example:
- "Flav" and "OD" have great separation between "Barbera" and "Barolo" classes.
- "Color" has decent separation between "Grignolino" and the others.

Some variables will likely be useless in classification. For example "Ash" and "Mg" could be potentially bad candidates.

3. Conclusion & Next Steps
Some primary takeaways from this EDA include:
- Univariate EDA shows variables that could benefit from log transformations.
  - *I did try this later and it didn't do much. See appendices.*
- Bivariate EDA points out some variables with clear breakdowns between the classes.
  - "Barbera" and "Barolo" classes are can be easily separated via the "Flav" and "OD" variables.
  - The "Grignolino" class can be separated out via the "Color" variable, but the split is not nearly as clean as the others.

Based on these results, we feel that it is worthwhile to move forward with classification algorithms. There are clear enough breakdowns just in this initial EDA that more sophisticated algorithms should be able to perform the classification task successfully.

4. Appendices

```
=============
4.1 Additional analysis
----------------------
### Does a log transformation on Color do anything?
```{r, fig.width=6, fig.height=4}
wine$l_Color <- log(wine$Color)
p <- ggplot(wine, aes(x=l_Color, fill=class)) +
  geom_density(aes(alpha=0.5)) +
  labs(x=names(wine)[i])
print(p)
```
Apparently it does not.
```