Wine Review Analysis using NLP methods

Chandini Gangadharan, Elizabeth Klier, and Nicola Ruggiero

2/22/2020

## Problem Statement

**We are setting up a new wine bottega in Seattle where there are many connoisseurs that enjoy tasting and consuming quality wine. Our problem is that there are so many choices of wines to choose from, and our shop will have limited space. We would like to determine what wines have the highest reviews looking at region/country, varieties, and wineries. Do the best wines come from the same region / country, or do the highest rated wines have similar key words in their descriptions? We will attempt to find out what wines we should stock in our new wine bottega by analyzing a wine review dataset from Twitter with description and point ratings.**

## About the Dataset used for this project:

**Rows**  This data set has 130,000, which is heavy on text type data.

**Columns / Fields**

**country**: The country that the wine is from  
**description**: Reviewer’s description of the wine  
**designation**: The vineyard within the winery where the grapes that made the wine are from  
**points**: The number of points WineEnthusiast rated the wine on a scale of 1-100 (though they say they only post reviews for wines that score >=80)  
**price**: The cost for a bottle of the wine  
**province**: The province or state that the wine is from  
**region\_1**: The wine growing area in a province or state (ie Napa)  
**region\_2**: Sometimes there are more specific regions specified within a wine growing area (ie Rutherford inside the Napa Valley)  
**taster\_name**: Reviewer’s name  
**taster\_twitter\_handle**: Reviewer’s Twitter handle  
**title**: The title of the wine review, which often contains the vintage if you’re interested in extracting that feature  
**variety**: The type of grapes used to make the wine (ie Pinot Noir)  
**winery**: The winery that made the wine

**Source**: <https://www.kaggle.com/zynicide/wine-reviews#winemag-data-130k-v2.csv>. Make sure to download the dataset with 130k rows *‘winemag-data-130k-v2.csv’* (there are other options available).

## Importing Packages and Libraries

# Below imports will be used and are needed to handle the data within this project.  
library(readr)  
library(dplyr)  
library(ggplot2)  
library(tidyverse)  
library(data.table)  
library(psych)  
library(xtable)  
  
# install package 'tm' for text mining.  
library('tm')  
  
# install package 'SnowballC'and 'wordcloud' for text analysis.  
library(SnowballC)  
library(wordcloud)

## Loading the dataset for analysis.

winemag\_df <- read.csv("winemag-data-130k-v2.csv")

## Data Cleaning and set-up

### Dealing with missing values/ empty values

**Checking NA values by column:**

colSums(is.na(winemag\_df))

## X country description   
## 0 0 0   
## designation points price   
## 0 0 8996   
## province region\_1 region\_2   
## 0 0 0   
## taster\_name taster\_twitter\_handle title   
## 0 0 0   
## variety winery   
## 0 0

* Only price has values of NA (8996 rows with NA).

**Checking for empty values by column:**

colSums(winemag\_df == "")

## X country description   
## 0 63 0   
## designation points price   
## 37465 0 NA   
## province region\_1 region\_2   
## 63 21247 79460   
## taster\_name taster\_twitter\_handle title   
## 26244 31213 0   
## variety winery   
## 1 0

* We see that there are empty values for fields country, designation, province, region\_1, region\_2, taster\_name, taster\_twitter\_handle, and variety.
* For exploratory analysis this is OK, we just need to make sure we take this into account when drawing conclusions from the data.

### Dealing with duplicate data

nrow(winemag\_df)

## [1] 129971

There are 129,971 rows in the dataset. We check to see how many are unique (exclude the index):

test <- unique(winemag\_df[ , 2:14])  
nrow(test) # verify no duplicate reviews/rows

## [1] 119988

* It looks like there are ~10,000 duplicate rows, so let’s get rid of them by redeclaring our main dataframe (we’ll drop the index)

winemag\_df <- unique(winemag\_df[ , 2:14])

## Exploratory Analysis

### Checking the data types

sapply(winemag\_df, class)

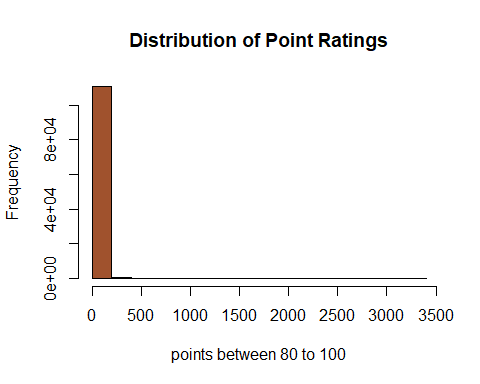
## country description designation   
## "factor" "factor" "factor"   
## points price province   
## "integer" "numeric" "factor"   
## region\_1 region\_2 taster\_name   
## "factor" "factor" "factor"   
## taster\_twitter\_handle title variety   
## "factor" "factor" "factor"   
## winery   
## "factor"

* As you can see in the above result that our dataset has only two numerical variable ‘points’ and ‘price’. All other variables are stated as ‘factor’ which means they are categorical type of data. From the attributes you can make out that this dataset is heavy on the text data because it has wine reviews which is basically a brief paragraph on each row.

### Exploring the Quantitative Measures

**Plotting the Price**

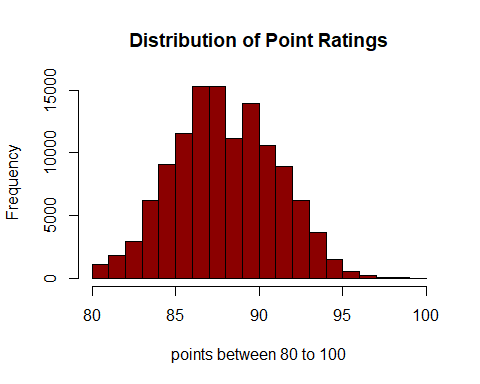
hist(winemag\_df$price, main='Distribution of Point Ratings', xlab='points between 80 to 100', col='sienna')



* As you can see the price of the wines show a very skewed distribution. Thi splot indicates that the wines ranging below approximately 200 makes up almost the total proportion whereas you can see a long tail and the price goes up till close to 3500. This could mean that there could be a few wine that falls under a higher price range.

**Plotting the Points**

hist(winemag\_df$points, main='Distribution of Point Ratings', xlab='points between 80 to 100', col='darkred')

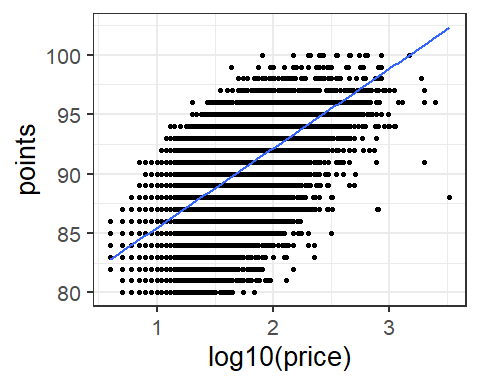


* We know that this dataset only includes wines with ratings of at least 80 points. The distribution of points seems to approximately symmetric about ~87 points, with *very* few wines receiving a rating greater than 95 points. We’ll investigate this more later.

**Plotting Price vs Points**

We know that these are the only two numerical variables in this dataset. Also, to get a fair idea of how price and points are realted lets plot the relationship using linear model.

ggplot(winemag\_df[complete.cases(winemag\_df),], aes(x = log10(price), y = points)) + geom\_point() + geom\_smooth(method = "lm") + theme\_bw(base\_size = 20)



* This visualization demonstrates a huge variation in price for any given rating. Using the logarithm of price makes it easy to see that wines with prices *several orders of magnitude* apart can still have the same rating. Based on these findings we can conclude that there is no meaningful relationship between price and points in this dataset, and that neither should be used to predict the other.

**Checking how influencing is the price on points**

mod <- lm(points~log10(price), data = winemag\_df[complete.cases(winemag\_df),])  
summary(mod)

##   
## Call:  
## lm(formula = points ~ log10(price), data = winemag\_df[complete.cases(winemag\_df),   
## ])  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -14.3031 -1.4937 0.1644 1.7447 9.2296   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 78.78574 0.03742 2105.6 <2e-16 \*\*\*  
## log10(price) 6.68388 0.02548 262.4 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 2.437 on 111591 degrees of freedom  
## Multiple R-squared: 0.3815, Adjusted R-squared: 0.3815   
## F-statistic: 6.883e+04 on 1 and 111591 DF, p-value: < 2.2e-16

* From the above summary statistics you can see that for a 1 point increase in ratings the price increases by $6. As we mentioned earlier that the relationship between price and points in this dataset is very ambiguous. A fine wine with lower price could also earn high ratings in its price category. So, it won’t be a good idea to base our predictions or recommendations around price and points only.

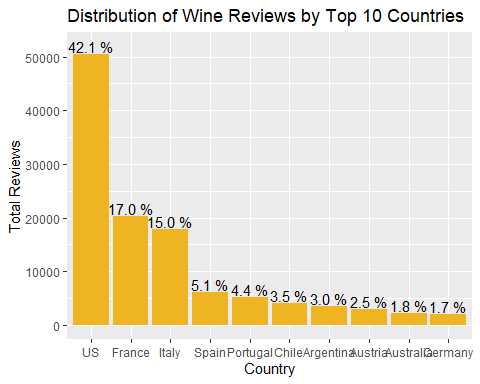
### Exploring the Qualitative Measures

**Quantity of Wines in Dataset by Country**

winereviewsCtry <- winemag\_df %>%   
 group\_by(country) %>%   
 summarise(total = n()) %>%   
 arrange(desc(total)) %>%   
 mutate(proportion\_of\_total = round(total/ sum(total), digits=7), cumulative\_proportion = cumsum(proportion\_of\_total))  
  
winereviewsCtry

## # A tibble: 44 x 4  
## country total proportion\_of\_total cumulative\_proportion  
## <fct> <int> <dbl> <dbl>  
## 1 US 50457 0.421 0.421  
## 2 France 20353 0.170 0.590  
## 3 Italy 17940 0.150 0.740  
## 4 Spain 6116 0.0510 0.791  
## 5 Portugal 5256 0.0438 0.834  
## 6 Chile 4184 0.0349 0.869  
## 7 Argentina 3544 0.0295 0.899  
## 8 Austria 3034 0.0253 0.924  
## 9 Australia 2197 0.0183 0.942  
## 10 Germany 1992 0.0166 0.959  
## # ... with 34 more rows

winereviewsCtry %>% head(10) %>%  
 ggplot( aes(x= factor(country, levels = winereviewsCtry$country[order(desc(winereviewsCtry$proportion\_of\_total))]), y = total)) +  
 geom\_col(fill='goldenrod2') +   
 geom\_text(aes(label = sprintf("%.1f %%", 100\*proportion\_of\_total), y = total + 1500)) +  
 labs(x="Country", y="Total Reviews", title="Distribution of Wine Reviews by Top 10 Countries")



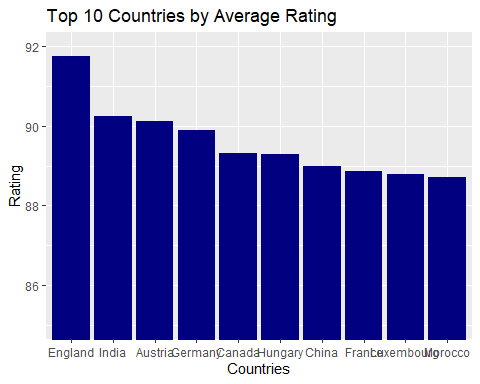
* From this visualization, we see that the U.S. produced nearly half of the wines reviewed in this dataset (42.1%), followed by France and Italy producing similar quantities at 17.0% and 15.0% respectively, and a big drop with Spain producing 5.1%.

**Countries with Highest Average Ratings**

wineRating = winemag\_df %>%   
 group\_by(country) %>%  
 summarise\_at(vars(points), funs(points = mean(., na.rm=T))) %>%  
 arrange(desc(points)) %>%  
 head(10)

## Warning: funs() is soft deprecated as of dplyr 0.8.0  
## Please use a list of either functions or lambdas:   
##   
## # Simple named list:   
## list(mean = mean, median = median)  
##   
## # Auto named with `tibble::lst()`:   
## tibble::lst(mean, median)  
##   
## # Using lambdas  
## list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))  
## This warning is displayed once per session.

ggplot(data=wineRating, aes(x=reorder(country,-points), y= points)) +   
 geom\_bar(stat="identity", fill = "navy") +   
 coord\_cartesian(ylim=c(85,92)) +   
 labs(x="Countries", y="Rating", title="Top 10 Countries by Average Rating")



* Interestingly, we see that England has the highest average wine rating of nearly 92 points, followed by India, Austria, and Germany with 90 points.
* The U.S. does not even make it in the top 10 countries by average rating, but perhaps this makes sense. We suspect that the reviewers in this dataset are predominantly American based on how many of the wines are from the U.S. despite their poorer performance compared to other wines. Americans may only hear about wines from some place like England or India if it comes highly recommended, therefore these foreign wines receive higher ratings.

**Most expensive wines by Country**

winemag\_df %>%  
 select(country, price) %>%  
 group\_by(country) %>%  
 summarise(maxprice = max(price, na.rm = TRUE)) %>%  
 arrange(desc(maxprice)) %>%  
 head(10)

## # A tibble: 10 x 2  
## country maxprice  
## <fct> <dbl>  
## 1 France 3300  
## 2 US 2013  
## 3 Austria 1100  
## 4 Portugal 1000  
## 5 Italy 900  
## 6 Australia 850  
## 7 Germany 775  
## 8 Spain 770  
## 9 Hungary 764  
## 10 Chile 400

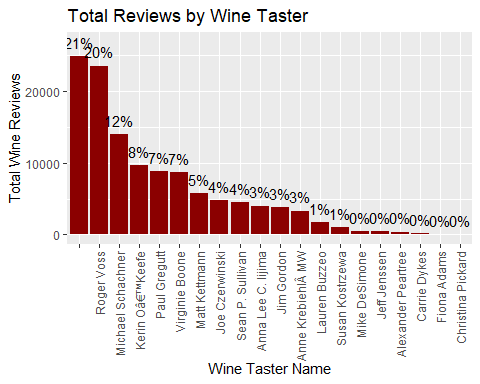
* Unsurpisingly, the most expensive wine reviewed comes from France at $3300. The country with second highest maximum wine price is the U.S. at $2013, followed by Austria and Portugal with ~$1000. All other max prices are under $1000, with Chile coming in at #10 with $400.

**Review Frequency by Taster**

#Group by taster and sort by total reviews. Analyze reviews distribution  
wineTstr <- winemag\_df %>% group\_by(taster\_name) %>% summarise(total=n()) %>%   
 arrange(desc(total)) %>%   
 mutate(totpcnt = round(total/ sum(total), 7), accum = cumsum(totpcnt))  
  
wineTstr

## # A tibble: 20 x 4  
## taster\_name total totpcnt accum  
## <fct> <int> <dbl> <dbl>  
## 1 "" 24917 0.208 0.208  
## 2 "Roger Voss" 23560 0.196 0.404  
## 3 "Michael Schachner" 14046 0.117 0.521  
## 4 "Kerin Oâ€™Keefe" 9697 0.0808 0.602  
## 5 "Paul Gregutt" 8868 0.0739 0.676  
## 6 "Virginie Boone" 8708 0.0726 0.748  
## 7 "Matt Kettmann" 5730 0.0478 0.796  
## 8 "Joe Czerwinski" 4766 0.0397 0.836  
## 9 "Sean P. Sullivan" 4461 0.0372 0.873  
## 10 "Anna Lee C. Iijima" 4017 0.0335 0.907  
## 11 "Jim Gordon" 3766 0.0314 0.938  
## 12 "Anne KrebiehlÂ MW" 3290 0.0274 0.965  
## 13 "Lauren Buzzeo" 1700 0.0142 0.979  
## 14 "Susan Kostrzewa" 1023 0.00853 0.988  
## 15 "Mike DeSimone" 461 0.00384 0.992  
## 16 "Jeff Jenssen" 436 0.00363 0.995  
## 17 "Alexander Peartree" 383 0.00319 0.999  
## 18 "Carrie Dykes" 129 0.00108 1.00   
## 19 "Fiona Adams" 24 0.0002 1.00   
## 20 "Christina Pickard" 6 0.00005 1

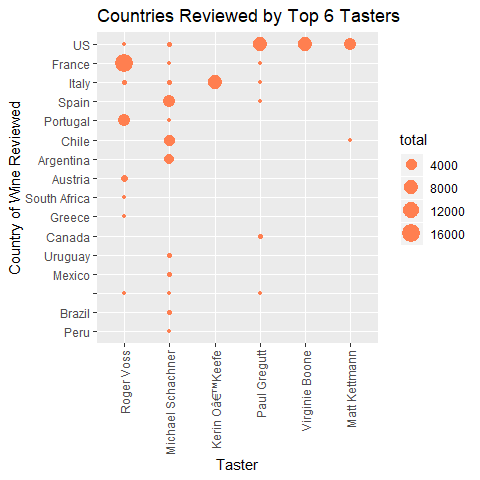
#Factor the taster name on desc order for organizing the bars on the next plot  
wineTstr$taster\_name <- factor(wineTstr$taster\_name, levels = wineTstr$taster\_name[order(-wineTstr$total)])  
  
#print a plot with the tasters and number of reviews  
wineTstr %>% ggplot(aes(x= taster\_name, y=total)) + geom\_col(fill='darkred') +  
 theme(axis.text.x = element\_text(angle = 90, vjust = 0.5, hjust=1)) +  
 geom\_text(aes(label = sprintf("%.f%%", 100\*totpcnt), y = total+2000)) +  
 labs(x="Wine Taster Name", y="Total Wine Reviews", title="Total Reviews by Wine Taster")



* This shows us that ~21% of our reviews come from unknown tasters, followed by Roger Voss holding the largest quantity of reviews (20% of all reviews in the dataset). We also see that the top 13 named reviewers make up the bulk (~80%) of the reviews (#15 makes up less than 1% of the reviews).

**Wine Countries Reviewed by Top 5 Named Tasters**

temp <- wineTstr %>% filter(taster\_name != "") %>% head(6)  
  
TopTstrCtry <- winemag\_df %>%   
 filter(taster\_name %in% temp$taster\_name) %>%  
 group\_by(taster\_name, country) %>%  
 summarise(total = n())  
  
library(ggplot2)  
TopTstrCtry %>%   
 ggplot( aes(x=factor(taster\_name, levels = wineTstr$taster\_name[order(-wineTstr$total)]),   
 y=factor(country, levels= winereviewsCtry$country[order(winereviewsCtry$total)]),   
 size = total)) +  
 geom\_point(color = 'coral') +  
 theme(axis.text.x = element\_text(angle = 90, vjust = 0.5, hjust=1)) +  
 labs(x="Taster", y="Country of Wine Reviewed",title="Countries Reviewed by Top 6 Tasters")



* This chart shows us that our most prolific named tasters are not super adventurous when it comes to trying wines from different countries. Our top taster, Roger Voss, (20% of all reviews in the dataset) primarily tastes wines from France, but also a number from Portugal and a small number from plaes like Austria, Chile, etc (7+ countries). Our #2 taster, Michael Schachner has the most diversified wine reviews out of the top 6 tasters (11+ countries), with the highest concentration on Spain, Chile, and Argentina. The next reviewers are much less diverse, with #3 (Kerin O’Keefe) *only* reviewing wines from Italy.
* This chart may suggest some underlying preferences towards certain wine countries among the most prestigious reviewers. For example Roger Voss is by far the most prolific reviewer and most of his reviews are of French wines, but nearly half of all wines in the dataset are from the U.S. In other words, the distribution of reviews by country for all reviews is not the same as the distributions for the top reviewers.

**Ratings and Points Analysis**

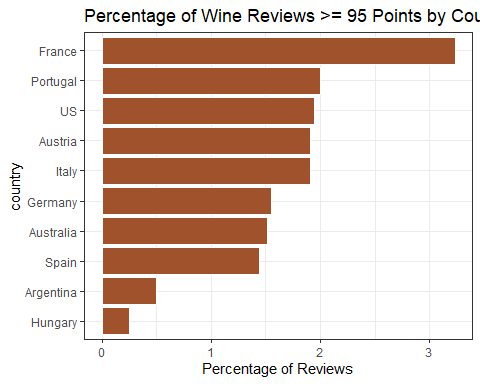
summary(winemag\_df$points)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 80.00 86.00 88.00 88.44 91.00 100.00

* Our previous estimate of 87 points for the average wine rating wasn’t far off–the average wine rating is 88.44 points. We also see that there is at least one wine with the maximum 100 points. We also see that 50% of the reviews are below 88 points, 75% below 91, and 25% below 86.
* We were interested in where the *best* wines come from, so let’s look at the countries with the greatest proportion of their wines having a rating of 95 points or above.

**Countries with the Highest Percentage of Reviews Having 95 Points or Above**

temp\_points<-winemag\_df %>%  
 filter(points >= 95) %>%  
 group\_by(country)%>%  
 summarise(n = n()) %>%  
 arrange(desc(n)) %>%  
 head(n = 10)  
  
p <-winemag\_df %>%  
 group\_by(country)%>%  
 summarise(n = n()) %>%  
 arrange(desc(n)) %>%  
 head(n = 10)  
  
temp\_points$p = p$n  
temp\_points$n = 100 \* temp\_points$n / temp\_points$p  
  
library(ggplot2)  
temp\_points %>%   
 ggplot(aes(x =reorder(country,n), y = n )) +  
 geom\_bar(stat='identity',colour="white", fill = c("sienna")) +  
 labs(x = 'country', y = 'Percentage of Reviews', title = 'Percentage of Wine Reviews >= 95 Points by Country') +  
 coord\_flip() +   
 theme\_bw()



* From this visualtion, we see that France has the highest percentage of its wines receiving a review of at least 95 points, with over 3% of the wines reviewed receiving 95 or above in our dataset.
* France is followed by Portugal, US, and Italy respectively in the highest percentages of wines receiving a rating of at least 95 / 100.

### Detailed Price analysis

summary(winemag\_df$price)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 4.00 17.00 25.00 35.62 42.00 3300.00 8395

* We see that the average wine in the dataset costs ~$35. The priciest wine is $3300 and the cheapest wine is $4. The median tells us that half of the reviews are for wines with a price of no more than $25. The 3rd quartile tells us that 75% of wines reviewed cost no more than $42.
* We will explore where the most expensive wines come from:

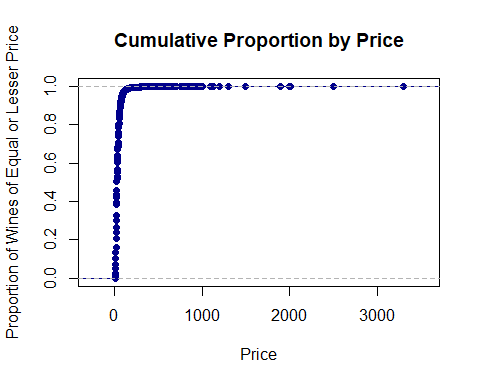
temp <-winemag\_df %>%  
 select(country,points,price,province,winery, variety) %>%  
 arrange(desc(price)) %>%  
 head(n = 20)  
  
temp

## country points price province winery  
## 1 France 88 3300 Bordeaux ChÃ¢teau les Ormes Sorbet  
## 2 France 96 2500 Bordeaux ChÃ¢teau PÃ©trus  
## 3 France 96 2500 Burgundy Domaine du Comte Liger-Belair  
## 4 US 91 2013 California Blair  
## 5 France 97 2000 Bordeaux ChÃ¢teau PÃ©trus  
## 6 France 96 2000 Burgundy Domaine du Comte Liger-Belair  
## 7 France 98 1900 Bordeaux ChÃ¢teau Margaux  
## 8 France 100 1500 Bordeaux ChÃ¢teau Lafite Rothschild  
## 9 France 100 1500 Bordeaux ChÃ¢teau Cheval Blanc  
## 10 France 96 1300 Bordeaux ChÃ¢teau Mouton Rothschild  
## 11 France 96 1200 Bordeaux ChÃ¢teau Haut-Brion  
## 12 France 94 1125 Burgundy Domaine du Comte Liger-Belair  
## 13 France 97 1100 Bordeaux ChÃ¢teau La Mission Haut-Brion  
## 14 Austria 94 1100 Wachau Emmerich Knoll  
## 15 Portugal 97 1000 Port W. & J. Graham's  
## 16 France 94 1000 Bordeaux ChÃ¢teau La Mission Haut-Brion  
## 17 Portugal 94 980 Port Kopke  
## 18 France 95 973 Burgundy Domaine Jacques Prieur  
## 19 France 97 932 Bordeaux ChÃ¢teau Haut-Brion  
## 20 Italy 94 900 Tuscany Biondi Santi  
## variety  
## 1 Bordeaux-style Red Blend  
## 2 Bordeaux-style Red Blend  
## 3 Pinot Noir  
## 4 Chardonnay  
## 5 Bordeaux-style Red Blend  
## 6 Pinot Noir  
## 7 Bordeaux-style Red Blend  
## 8 Bordeaux-style Red Blend  
## 9 Bordeaux-style Red Blend  
## 10 Bordeaux-style Red Blend  
## 11 Bordeaux-style Red Blend  
## 12 Pinot Noir  
## 13 Bordeaux-style Red Blend  
## 14 GrÃ¼ner Veltliner  
## 15 Port  
## 16 Bordeaux-style White Blend  
## 17 Port  
## 18 Pinot Noir  
## 19 Bordeaux-style White Blend  
## 20 Sangiovese Grosso

* 9/10 of the most expensive wines come from France (the exception from the U.S. at #4).
* 7/10 come from Bordeaux which are also Bordeaux-style red blends
* 2 from Burgundy are Pinot Noir.
* 1 from California is a Chardonnay.
* No two wines on this list come from the same winery except the 2 from Domaine du Comte Liger-Belair.

**Let’s look at the cumulative proportion of wine prices for all reviews:**

df <- winemag\_df %>%   
 filter(!is.na(price))%>%   
 select(price) %>%   
 table()  
  
cumulative\_proportion <- ecdf(winemag\_df$price)  
plot(cumulative\_proportion, main='Cumulative Proportion by Price', xlab='Price', ylab='Proportion of Wines of Equal or Lesser Price', col='dark blue')



**We see that nearly all wines cost less than ~ $100, but to be sure let’s look at the exact values:**

df1 <- prop.table(df)   
  
cat("Portion of wines under $300: ", cumsum(df1)['300'], '\n')

## Portion of wines under $300: 0.9972041

cat("Portion of wines under $200: ", cumsum(df1)['200'], '\n')

## Portion of wines under $200: 0.9940408

cat("Portion of wines under $100: ", cumsum(df1)['100'], '\n')

## Portion of wines under $100: 0.9710645

cat("Portion of wines under $50: ", cumsum(df1)['50'], '\n')

## Portion of wines under $50: 0.8343982

cat("Portion of wines under $25: ", cumsum(df1)['25'], '\n')

## Portion of wines under $25: 0.5056858

cat("Portion of wines under $10: ", cumsum(df1)['10'], '\n')

## Portion of wines under $10: 0.05301408

* This confirms that ~97% of wines reviewed cost less than $100. Based on this, we believe we can deliver on nearly all of our customers’ demands with just wines costing less than $100.

## Text Analysis using NLP techniques

Now you might have got a fair idea of how and what types of wine are present in our dataset and and which all types a wine can be classified under. We’re now interested in mining the reviews to understand how the experts feel about the wine. Expert’s reviews on public magazines or social media may influence the buyers or tasters. WWe Would like to know what words are typically found in the wine descriptions for our reviews. To do this We start by aggregating all the words in our descriptions using the ‘tm’ package.

Now, before we jump into applying text analysis methods let us give you a brief intro on Text Analysis and Text Mining methods.

**Quick background:** text analytics (also known as text mining) refers to a discipline of computer science that combines machine learning and natural language processing (NLP) to draw meaning from unstructured text documents.

There are a few basic steps involved in preparing an unstructured text document for deeper analysis:

* Language Identification - The first step in text analytics is identifying what language the text is written in. In this project since we know that the reviews are in English its becomes easy.
* Tokenization - it is the process of breaking down text document apart into those pieces. Tokens can be Words, Punctuation (exclamation points intensify sentiment), Hyperlinks ([https://](NULL)…) or Possessive markers (apostrophes).
* Sentence Breaking - to know exactly where the sentence ended or to be able to tell where the boundaries are on a sentence.
* Part of Speech Tagging (PoS)- is the process of determining the part of speech of every token in a document, and then tagging it whether a given token represents a proper noun or a common noun, or if it’s a verb, an adjective, or something else entirely.
* Chunking - means assigning PoS-tagged tokens to phrases.
* Syntax Parsing - is a way to determine the structure of a sentence. In truth, syntax parsing is really just fancy talk for sentence diagraming.
* Sentence Chaining - The final step in preparing unstructured text for deeper analysis is sentence chaining, sometimes known as sentence relation. Grouping of words and phrases into semantically similar groups.

**About the ‘tm’ package** We are using ‘tm’ package for text mining. It is a framework for text mining applications within R. You will come across many other functions that we use to execute certain NLP tasks. Short descriptions are added to guide you regarding the usage of that function.

Since Text Mining is the process of summarizing a large amount of text into usable statistics. In text mining, the phrase vectorizing a document may come up. This means turning a document into a vector, which could be a list of the words in the document and a numeric value signifying the number of times that particular word appears in the document.

**Accessing the description column and creating a new vector called ‘review’.**

review<-winemag\_df$description

**Interpreting review vector with ‘tm’ package**

#VectorSource(): Takes a grouping of texts and makes each element of the resulting vector a document within your R Workspace. There are many types of sources, but VectorSource() is made for working with character objects in R.  
library(tm)  
review<-VectorSource(review)  
  
#\*\*Creating VCorpus object\*\*  
#VCorpus(): Takes a source object and makes a volatile corpora. A VCorpus object is created from a source object. In essence, a corpus is a collection of documents. Since the object is volatile, all changes only affect the corresponding R object. Also for volatile objects, once the variable is destroyed, the corpus object is also destroyed.  
rev<-VCorpus(review)  
  
#\*\*Removing punctuation from reviews\*\*  
#tm\_map(x, FUN, ...): allows for the application of transformation functions such as removing punctuation from each document within a corpus object, x.  
rev<-tm\_map(rev, removePunctuation)  
  
  
#\*\*Converting to lowercase to make future cleaning easier\*\*  
#tm\_map(x, FUN, ...): allows for the application of transformation functions such as converting CAPS to lower for each document within a corpus object, x.  
rev<-tm\_map(rev, content\_transformer(tolower))  
  
#\*\*Removing numbers if any from the reviews to keep it simple to analyse\*\*  
#tm\_map(x, FUN, ...): allows for the application of transformation functions such as removing numbers from each document within a corpus object, x.  
rev<-tm\_map(rev, removeNumbers)  
  
#\*\*Remove excess white spaces\*\*  
# Using tm\_map, we clean the description data by removing all excess whitespaces and the words that are common to the English language (called 'stop words')  
rev<-tm\_map(rev, stripWhitespace)  
  
#\*\*Remove stop words (takes around 1 min. since its processing large amount of text data)\*\*  
# Stopwords are common words that are filtered out in a document so statistics concerning them will not be calculated. This allows for a clearer inspection of the more of the interesting words. We are using a pre made Stopwords list on 'en' English language.  
rev<-tm\_map(rev, removeWords, stopwords("en"))  
  
#\*\*Stemming the tokens from the reviews\*\*  
# To prevent seeing words with the same root listed multiple times (ex. 'accent', 'accented', 'accents'), we remove everything but the 'stem' of a word (remove ending 's', 'ed', etc.). These are also called 'suffixes'.  
rev<- tm\_map(rev, stemDocument)  
  
#\*\*Creating DTM matrix (takes around 1 min)\*\*  
# In text mining, it is important to create the document-term matrix (DTM) of the corpus we are interested in. A DTM is basically a matrix, with documents designated by rows and words by columns, that the elements are the counts or the weights.  
  
#TermDocumentMatrix(): takes a VCorpus object, x, and creates a matrix as a list object where the document names are the column names and the terms are the row names. Controls such as stopwords, tolower, etc. can be used to clean up the documents.  
rev\_dtm<-DocumentTermMatrix(rev)  
  
#\*\*Finding the most popular stems\*\*  
# Here we are trying to look at the terms/ words that has appeared atleast 10000 times in our reviews. i.e finding frequently occuring terms and we are setting our treshold for basic frequency.   
rev\_freq<-findFreqTerms(rev\_dtm, lowfreq=10000)  
rev\_freq

## [1] "acid" "age" "appl" "aroma" "balanc"   
## [6] "berri" "black" "blackberri" "blend" "bright"   
## [11] "cherri" "citrus" "crisp" "dark" "dri"   
## [16] "drink" "finish" "flavor" "fresh" "fruit"   
## [21] "fruiti" "give" "hint" "light" "miner"   
## [26] "nose" "note" "now" "oak" "offer"   
## [31] "palat" "plum" "raspberri" "red" "rich"   
## [36] "ripe" "show" "soft" "spice" "structur"   
## [41] "sweet" "tannin" "textur" "white" "wine"

* We end up with a list of all ‘stem’ words appearing more than 10,000 times throughout the dataset. This is good, but these words are just sorted alphabetically, not by frequency. It’d also be nice to visualize the prominence of some words over others, so we should try a word cloud.
* Let’s look at the most common words in reviews with at least a 95 point rating. We’ll look at the countries that have the largest proportion of wines with these reviews of at least 95 points: **France, Portugal, U.S., Austria, and Italy**.

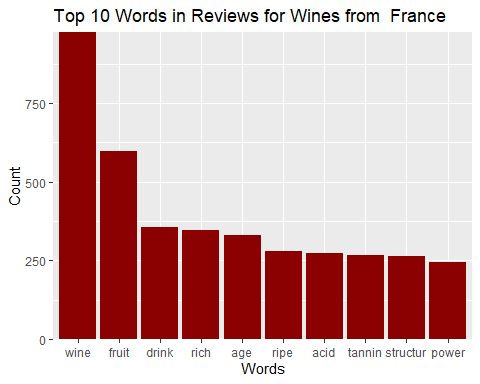
## In Depth Analysis

### Exploring the most common words in reviews with atleast 95 point rating for top 5 countries.

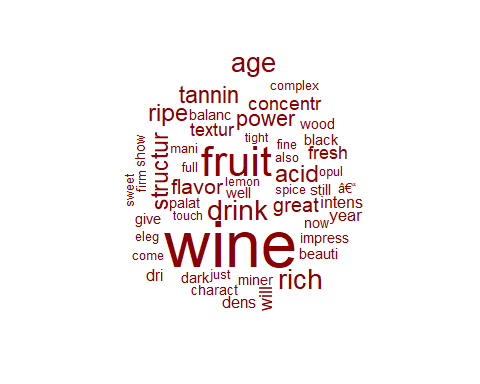
**Let’s look at #1 for proportion of reviews of 95 or above, we start with France:**

library(tm)  
plug\_in='France'  
  
s <- subset(winemag\_df, country==plug\_in)  
temp <- subset(s, points >= 95)  
  
sw<-temp$description  
  
sw<-VectorSource(sw)  
sw<-VCorpus(sw)  
  
clean\_corpus <- function(corpus){  
 corpus <- tm\_map(corpus, removePunctuation)  
 corpus <- tm\_map(corpus, removeNumbers)  
 corpus <- tm\_map(corpus, content\_transformer(tolower))  
 corpus <- tm\_map(corpus, removeWords, stopwords("en"))  
 corpus <- tm\_map(corpus, stemDocument)  
 corpus <-tm\_map(corpus, stripWhitespace)  
 return(corpus)  
}  
  
swc<-clean\_corpus(sw)  
  
swctdm<-TermDocumentMatrix(swc)

#creating matrix from TDM  
swcm<-as.matrix(swctdm)  
sf<-rowSums(swcm)  
sf<-sort(sf, decreasing=T)  
  
#creating a data frame  
swf<-data.frame(term=names(sf), num=sf)  
  
t = paste('Top 10 Words in Reviews for Wines from ', plug\_in)  
  
  
#creating a plot of 10 top stems  
ggplot(data=head(swf, 10), aes(x=factor(term, levels = swf$term[order(-swf$num)]), y=num)) +   
 geom\_col(fill="darkred") +   
 labs(x="Words", y="Count", title=t) +  
 scale\_y\_continuous(expand = c(0, 0))



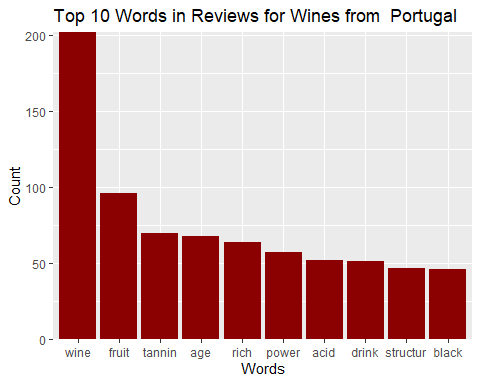
#creating a wordcloud  
wordcloud(swf$term, swf$num, max.words=50, color="darkred")



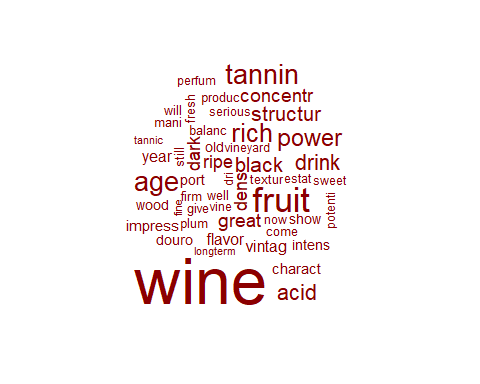
**Now let’s look at #2 for proportion of reviews of 95 or above, Portugal:**

plug\_in='Portugal'  
  
s <- subset(winemag\_df, country==plug\_in)  
temp <- subset(s, points >= 95)  
  
sw<-temp$description  
  
sw<-VectorSource(sw)  
sw<-VCorpus(sw)  
  
clean\_corpus <- function(corpus){  
 corpus <- tm\_map(corpus, removePunctuation)  
 corpus <- tm\_map(corpus, removeNumbers)  
 corpus <- tm\_map(corpus, content\_transformer(tolower))  
 corpus <- tm\_map(corpus, removeWords, stopwords("en"))  
 corpus <- tm\_map(corpus, stemDocument)  
 corpus <-tm\_map(corpus, stripWhitespace)  
 return(corpus)  
}  
  
swc<-clean\_corpus(sw)  
  
swctdm<-TermDocumentMatrix(swc)

#creating matrix from TDM  
swcm<-as.matrix(swctdm)  
sf<-rowSums(swcm)  
sf<-sort(sf, decreasing=T)  
  
#creating a data frame  
swf<-data.frame(term=names(sf), num=sf)  
  
t = paste('Top 10 Words in Reviews for Wines from ', plug\_in)  
  
#creating a plot of 10 top stems  
ggplot(data=head(swf, 10), aes(x=factor(term, levels = swf$term[order(-swf$num)]), y=num)) +   
 geom\_col(fill="darkred") +   
 labs(x="Words", y="Count", title=t) +  
 scale\_y\_continuous(expand = c(0, 0))



#creating a wordcloud  
wordcloud(swf$term, swf$num, max.words=50, color="darkred")

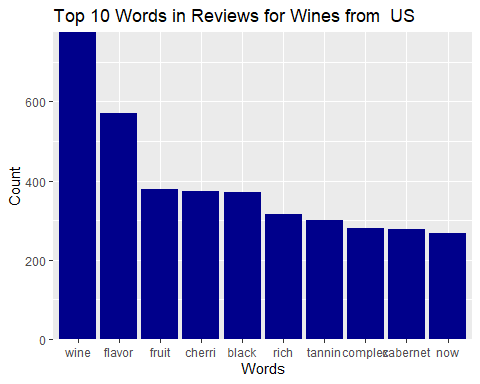


* These descriptions are similar, but one difference is ‘ripe’ being the #3 word for the highly-rated French wines, but wines from Portugal have ‘dark’ and ‘black’ in the top 10 words. Perhaps the highly-rated wines from Portugal are a different variety than those from France.

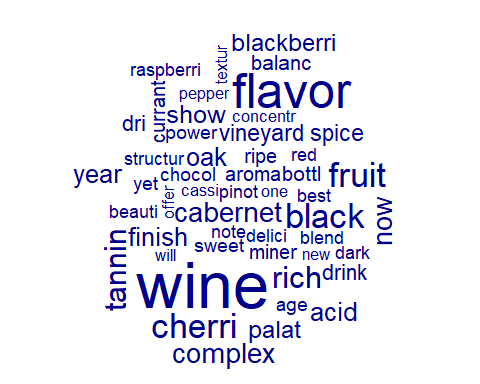
**let’s look at #3 for proportion of reviews of 95 or above for U.S.:**

plug\_in='US'  
  
s <- subset(winemag\_df, country==plug\_in)  
temp <- subset(s, points >= 95)  
  
sw<-temp$description  
  
sw<-VectorSource(sw)  
sw<-VCorpus(sw)  
  
clean\_corpus <- function(corpus){  
 corpus <- tm\_map(corpus, removePunctuation)  
 corpus <- tm\_map(corpus, removeNumbers)  
 corpus <- tm\_map(corpus, content\_transformer(tolower))  
 corpus <- tm\_map(corpus, removeWords, stopwords("en"))  
 corpus <- tm\_map(corpus, stemDocument)  
 corpus <-tm\_map(corpus, stripWhitespace)  
 return(corpus)  
}  
  
swc<-clean\_corpus(sw)  
  
swctdm<-TermDocumentMatrix(swc)

#creating matrix from TDM  
swcm<-as.matrix(swctdm)  
sf<-rowSums(swcm)  
sf<-sort(sf, decreasing=T)  
  
#creating a data frame  
swf<-data.frame(term=names(sf), num=sf)  
  
t = paste('Top 10 Words in Reviews for Wines from ', plug\_in)  
  
#creating a plot of 10 top stems  
ggplot(data=head(swf, 10), aes(x=factor(term, levels = swf$term[order(-swf$num)]), y=num)) +   
 geom\_col(fill="darkblue") +   
 labs(x="Words", y="Count", title=t) +  
 scale\_y\_continuous(expand = c(0, 0))



#creating a wordcloud  
wordcloud(swf$term, swf$num, max.words=50, color="darkblue")

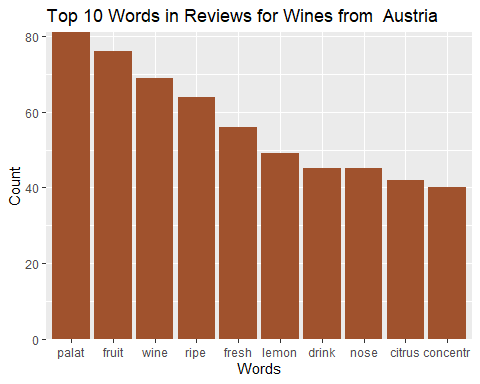


* Again, ‘fruit’ is the top word, but we also see a few things that are new: ‘cherri(es)’, ‘oak’, ‘complex’, ‘cabernet’, and ‘blackberri(es)’.

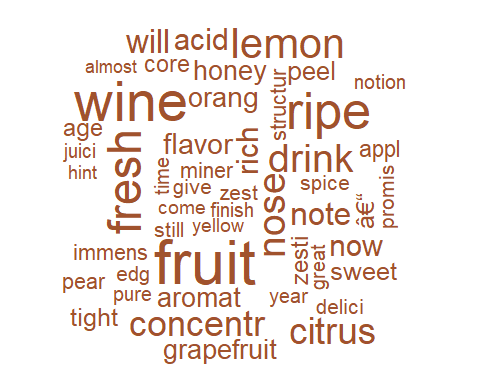
**Let’s look at #4 for proportion of reviews of 95 or above for Austria:**

plug\_in='Austria'  
  
s <- subset(winemag\_df, country==plug\_in)  
temp <- subset(s, points >= 95)  
  
sw<-temp$description  
  
sw<-VectorSource(sw)  
sw<-VCorpus(sw)  
  
clean\_corpus <- function(corpus){  
 corpus <- tm\_map(corpus, removePunctuation)  
 corpus <- tm\_map(corpus, removeNumbers)  
 corpus <- tm\_map(corpus, content\_transformer(tolower))  
 corpus <- tm\_map(corpus, removeWords, stopwords("en"))  
 corpus <- tm\_map(corpus, stemDocument)  
 corpus <-tm\_map(corpus, stripWhitespace)  
 return(corpus)  
}  
  
swc<-clean\_corpus(sw)  
  
swctdm<-TermDocumentMatrix(swc)

#creating matrix from TDM  
swcm<-as.matrix(swctdm)  
sf<-rowSums(swcm)  
sf<-sort(sf, decreasing=T)  
  
#creating a data frame  
swf<-data.frame(term=names(sf), num=sf)  
  
t = paste('Top 10 Words in Reviews for Wines from ', plug\_in)  
  
#creating a plot of 10 top stems  
ggplot(data=head(swf, 10), aes(x=factor(term, levels = swf$term[order(-swf$num)]), y=num)) +   
 geom\_col(fill="sienna") +   
 labs(x="Words", y="Count", title=t) +  
 scale\_y\_continuous(expand = c(0, 0))



#creating a wordcloud  
wordcloud(swf$term, swf$num, max.words=50, color="sienna")

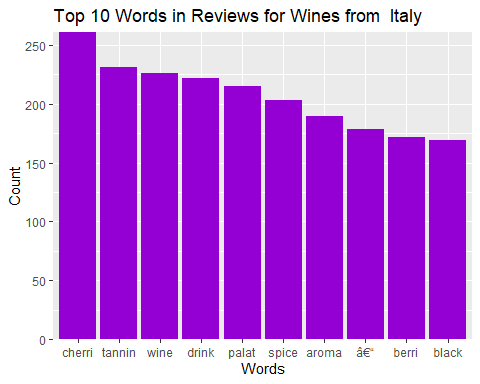


* We have a new #1 word: **‘palat(e)’**. We also have some new words: **‘fresh’, ‘lemon’, and ‘citrus’.** Perhaps **Austrian wines are more commonly lighter, zestier, and citrus-y.**

**let’s look at #5 for proportion of reviews of 95 or above for Italy:**

plug\_in='Italy'  
  
s <- subset(winemag\_df, country==plug\_in)  
temp <- subset(s, points >= 95)  
  
sw<-temp$description  
  
sw<-VectorSource(sw)  
sw<-VCorpus(sw)  
  
clean\_corpus <- function(corpus){  
 corpus <- tm\_map(corpus, removePunctuation)  
 corpus <- tm\_map(corpus, removeNumbers)  
 corpus <- tm\_map(corpus, content\_transformer(tolower))  
 corpus <- tm\_map(corpus, removeWords, stopwords("en"))  
 corpus <- tm\_map(corpus, stemDocument)  
 corpus <-tm\_map(corpus, stripWhitespace)  
 return(corpus)  
}  
  
swc<-clean\_corpus(sw)  
  
swctdm<-TermDocumentMatrix(swc)

#creating matrix from TDM  
swcm<-as.matrix(swctdm)  
sf<-rowSums(swcm)  
sf<-sort(sf, decreasing=T)  
  
#creating a data frame  
swf<-data.frame(term=names(sf), num=sf)  
  
t = paste('Top 10 Words in Reviews for Wines from ', plug\_in)  
  
#creating a plot of 10 top stems  
ggplot(data=head(swf, 10), aes(x=factor(term, levels = swf$term[order(-swf$num)]), y=num)) +   
 geom\_col(fill="darkviolet") +   
 labs(x="Words", y="Count", title=t) +  
 scale\_y\_continuous(expand = c(0, 0))



#creating a wordcloud  
wordcloud(swf$term, swf$num, max.words=50, color="darkviolet")



The most common word for nearly all countries was ‘fruit’, but ‘fruit’ isn’t even in the top 10 words for highly-rated Italian wines. Instead **‘cherri(es)’** is the most common word, with the new words like **‘leather’ and ‘spice’ also in the top 10.**

## Which Wines to Stock?

* Our exploratory analysis above showed us that:
  1. We don’t need to carry any wines over $100 to satisfy our customers since there is a wide range and fine quality wines available under that price.
  2. There aren’t many wines that have ratings of at least 95 points.
  3. France, Austria, US, Portugal, and Italy carry the highest proportion of those wines that have a rating of 95 or above.
* We think we should carry some wines from each of the 5 countries with the largest proportion of reviews of 95 or above because these must be recognized as places with great wines. The wines stocked should also cost less than $100 because our store isn’t super fancy.

**Let’s see how many wines have ratings of 95 and up, cost $100 or less, and come from one of those 5 key countries:**

# Here we are creating a subset of the original dataset containg wines filtered on the basis of points, price and belongs to top 5 countries selected.  
stock <- subset(winemag\_df, points >= 95)  
stock <- subset(stock, price <= 100)  
stock <- subset(stock, country %in% c('France', 'Austria', 'US', 'Portugal', 'Italy'))  
  
head(stock[c('country', 'variety', 'title', 'price', 'points')],10) #review the dataset

## country variety  
## 353 US Chardonnay  
## 356 US Chardonnay  
## 362 Italy Nebbiolo  
## 365 US Pinot Noir  
## 1558 US Pinot Noir  
## 1561 US Bordeaux-style Red Blend  
## 1568 US Pinot Noir  
## 1571 Italy Red Blend  
## 1578 US Pinot Noir  
## 1579 France Bordeaux-style Red Blend  
## title  
## 353 Rochioli 2014 South River Chardonnay (Russian River Valley)  
## 356 Rochioli 2014 Sweetwater Chardonnay (Russian River Valley)  
## 362 Bel Colle 2012 Simposio (Barolo)  
## 365 Winderlea 2014 Weber Vineyard Pinot Noir (Dundee Hills)  
## 1558 Williams Selyem 2009 Precious Mountain Vineyard Pinot Noir (Sonoma Coast)  
## 1561 Pirouette 2008 Red Wine Red (Columbia Valley (WA))  
## 1568 Williams Selyem 2009 Allen Vineyard Pinot Noir (Russian River Valley)  
## 1571 Tenuta Argentiera 2008 Argentiera (Bolgheri Superiore)  
## 1578 Williams Selyem 2009 Rochioli Riverblock Vineyard Pinot Noir (Russian River Valley)  
## 1579 ChÃ¢teau Beau-SÃ©jour BÃ©cot 2009 Saint-Ã‰milion  
## price points  
## 353 68 96  
## 356 68 96  
## 362 60 95  
## 365 48 95  
## 1558 94 99  
## 1561 50 98  
## 1568 82 97  
## 1571 75 96  
## 1578 78 96  
## 1579 75 96

* We see that there are 1,200+ wines meeting these criteria. This is a wide range of wines to deal. So lets drill down and also take into consideration a normal price range for price sensitive customers too…

**Exploring wines with ratings of 85 or above, but cost $30 or less:**

cheapstock <- subset(winemag\_df, points >= 85)  
cheapstock <- subset(cheapstock, price <= 30)  
cheapstock <- subset(cheapstock, country %in% c('France', 'Austria', 'US', 'Portugal', 'Italy'))  
  
head(cheapstock[c('country', 'variety', 'title', 'price', 'points')],10) #review the dataset

## country variety  
## 2 Portugal Portuguese Red  
## 3 US Pinot Gris  
## 4 US Riesling  
## 7 Italy Frappato  
## 8 France GewÃ¼rztraminer  
## 10 France Pinot Gris  
## 11 US Cabernet Sauvignon  
## 12 France GewÃ¼rztraminer  
## 15 US Chardonnay  
## 21 US Red Blend  
## title  
## 2 Quinta dos Avidagos 2011 Avidagos Red (Douro)  
## 3 Rainstorm 2013 Pinot Gris (Willamette Valley)  
## 4 St. Julian 2013 Reserve Late Harvest Riesling (Lake Michigan Shore)  
## 7 Terre di Giurfo 2013 Belsito Frappato (Vittoria)  
## 8 Trimbach 2012 Gewurztraminer (Alsace)  
## 10 Jean-Baptiste Adam 2012 Les Natures Pinot Gris (Alsace)  
## 11 Kirkland Signature 2011 Mountain CuvÃ©e Cabernet Sauvignon (Napa Valley)  
## 12 Leon Beyer 2012 Gewurztraminer (Alsace)  
## 15 Mirassou 2012 Chardonnay (Central Coast)  
## 21 QuiÃ©vremont 2012 Vin de Maison Red (Virginia)  
## price points  
## 2 15 87  
## 3 14 87  
## 4 13 87  
## 7 16 87  
## 8 24 87  
## 10 27 87  
## 11 19 87  
## 12 30 87  
## 15 12 87  
## 21 23 87

* This is a staggering 43,000+ rows, so we have to cut this down. How many different country/variety pairs are in our more expensive stock?

unique\_country\_variety <- unique(stock[, c('country', 'variety')])  
head(unique\_country\_variety, 10)

## country variety  
## 353 US Chardonnay  
## 362 Italy Nebbiolo  
## 365 US Pinot Noir  
## 1561 US Bordeaux-style Red Blend  
## 1571 Italy Red Blend  
## 1579 France Bordeaux-style Red Blend  
## 1584 US Syrah  
## 2514 US Riesling  
## 3062 US Cabernet Sauvignon  
## 3065 US Bordeaux-style White Blend

* We only have 92 unique country/variety pairs in this set. **Let’s assume that the varieties of wines from the 95+ points stock are widely recognized and regarded as good choices, so we could perhaps carry a few wines of each country/variety.**

Let’s carry **1) the wine of that country and variety with the highest rating** and 2) **the wine of that country and variety with the lowest price but a rating of at least 85 points:**

title <- c()  
country <- c()  
variety <- c()  
price <- c()  
rating <- c()  
   
for(i in 1:nrow(unique\_country\_variety)) {  
 c <- toString(unique\_country\_variety[i, 1])  
 v <- toString(unique\_country\_variety[i, 2])  
 t <- subset(stock, country == c & variety ==v)  
 s <- subset(cheapstock, country == c & variety ==v)  
 first = (i \* 2) - 1  
 # get the highest rated wine  
 title[[first]] <- toString(t[which.max(t$points), 'title'])  
 country[[first]] <- toString(t[which.max(t$points), 'country'])  
 variety[[first]] <- toString(t[which.max(t$points), 'variety'])  
 price[[first]] <- toString(t[which.max(t$points), 'price'])  
 rating[[first]] <- toString(t[which.max(t$points), 'points'])  
 # also get the cheapest wine of this country/variety  
 second = first + 1  
 title[[second]] <- toString(s[which.min(t$price), 'title'])  
 country[[second]] <- toString(s[which.min(t$price), 'country'])  
 variety[[second]] <- toString(s[which.min(t$price), 'variety'])  
 price[[second]] <- toString(s[which.min(t$price), 'price'])  
 rating[[second]] <- toString(s[which.min(t$price), 'points'])  
}  
  
result <- data.frame(title, country, variety, price, rating)  
head(result, 10)

## title  
## 1 Failla 2010 Estate Vineyard Chardonnay (Sonoma Coast)  
## 2 Roanoke Vineyards 2011 The Wild Chardonnay (North Fork of Long Island)  
## 3 Brezza 2013 Cannubi (Barolo)  
## 4 Serradenari 2006 Barolo  
## 5 Williams Selyem 2009 Precious Mountain Vineyard Pinot Noir (Sonoma Coast)  
## 6 Block Nine 2010 Caiden's Vineyards Pinot Noir (California)  
## 7 Pirouette 2008 Red Wine Red (Columbia Valley (WA))  
## 8 Baiting Hollow Farm Vineyard 2010 Horse Rescue Mirage Red (North Fork of Long Island)  
## 9 Castello Banfi 2007 Excelsus Red (Toscana)  
## 10 Terre de Trinci 2003 Riserva (Montefalco Rosso)  
## country variety price rating  
## 1 US Chardonnay 44 99  
## 2 US Chardonnay 20 86  
## 3 Italy Nebbiolo 60 98  
## 4 Italy Nebbiolo 26 91  
## 5 US Pinot Noir 94 99  
## 6 US Pinot Noir 13 85  
## 7 US Bordeaux-style Red Blend 50 98  
## 8 US Bordeaux-style Red Blend 25 87  
## 9 Italy Red Blend 81 97  
## 10 Italy Red Blend 18 86

## Recommendations

We recommend stocking 184 varieties of wine for this wine bottega. We expect our customers to be affluent wine-lovers, so we want to provide a great variety of wines that come highly recommended. We recommend focusing on those wines that come from the 5 countries with the highest proportion of wines given a review of 95 points or above: France, Austria, US, Portugal, and Italy. We expect that our customers will recognize these countries as being places that produce quality wines. As a casual bottega, we also want to offer wines with affordable prices. There are 92 unique country-variety combinations amongst those wines with at least a 95-point rating, costing less than $100, and originating from one of the 5 countries discussed above. We recommend carrying the highest-rated for each of these 92 country-varieties because our customers should recognize them as the best of the best and be excited to purchase. However, we notice that few of these wines are less than $60 a bottle. In order to serve those customers who love wine but are not ready to spend this much on a bottle, we recommend stocking another 92 wines of the same country-varieties as those for the best of the best. This additional stock of wines are the cheapest wines of the country-variety with a rating of at least 85 points. We believe that carrying pairs of well-known country-variety wines, the best wine and the best value alternative, will satisfy our customers’ demand for highly-recommended wines.