wine_analysis_markdown

Brendon Kaufman

September 10, 2018

Wine Analysis Markdown Script

This script analyzes the data set of 130k wine reviews posted by user "zynicide" on Kaggle.com This project is intended as a first foray into show-casing my R capabilities, focusing on basic data analysis using style conventions from Hadley Wickham's tidyverse. I will be replicating this code as a markdown file as well.

————Loading in packages and the data————————

```
library(tidyverse)
library(ggmap)
library(DT)
wine130k <- "data_repository\\winereviews130k.csv"
geocoded_fr_regions_path <- "data_repository\\geocoded_french_regions.csv"
wine_reviews <- read_csv(wine130k)
geocoded_fr_regions <- read_csv(geocoded_fr_regions_path)
```

————Start Data Exploration————————

To see what the data set looks like, we'll look at a few exploratory features

```
head(wine_reviews)
```

```
## # A tibble: 6 x 14
##      X1 country  description    designation points price province region_1
##   <int> <chr>    <chr>          <chr>        <int> <dbl> <chr>    <chr>
## 1     0 Italy    Aromas includ~ Vulkà Bian~     87    NA Sicily ~ Etna
## 2     1 Portugal This is ripe ~ Avidagos        87    15 Douro    <NA>
## 3     2 US       Tart and snap~ <NA>            87    14 Oregon   Willame~
## 4     3 US       Pineapple rin~ Reserve La~     87    13 Michigan Lake Mi~
## 5     4 US       Much like the~ Vintner's ~     87    65 Oregon   Willame~
## 6     5 Spain    Blackberry an~ Ars In Vit~     87    15 Norther~ Navarra
## # ... with 6 more variables: region_2 <chr>, taster_name <chr>,
## #   taster_twitter_handle <chr>, title <chr>, variety <chr>, winery <chr>
```

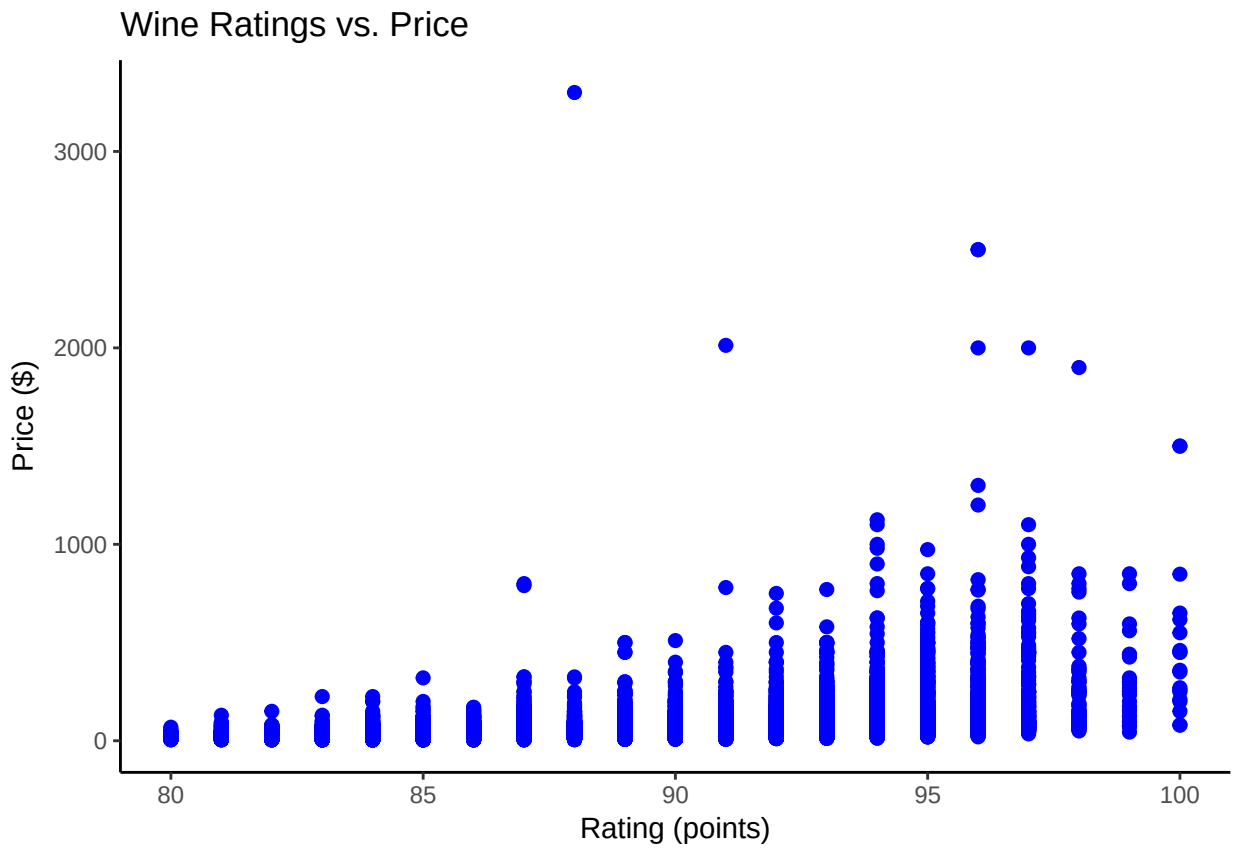
Each row is a bottle of wine with features such as its country of origin and privince/region as well as the name of the taster, the variety of wine, its rating, a description, and its price.

One thing that seems most interesting is seeing whether a wine's rating correlates to its price. Do expensive wines tend to be rated higher?

```
price_vs_rating <- ggplot(wine_reviews, aes(points, price))

price_vs_rating + geom_point(color = "blue", size = 2) +
  theme_classic() +
  labs(x = "Rating (points)", y = "Price ($)", title = "Wine Ratings vs. Price")
```

```
## Warning: Removed 8996 rows containing missing values (geom_point).
```



At first glance, there seems to be some relationship between the two variables. However, there are some odd data points, such as the most expensive wine bottle which only has a rating in the high 80s. We can see the correlation coefficient between these two variables with the following, using the `complete.obs` argument to tell R to disregard bottles with NA in price or points

```
cor(wine_reviews$points, wine_reviews$price, use = "complete.obs")
```

```
## [1] 0.4161667
```

Correlation of r = 0.42 shows a medium-strength positive correlation between Rating and Price

————Start Ratings By Country Exploration————

Something else that seems interesting is knowing which countries have the highest rating wines. We can figure this out easily using the dplyr packages to sort the data. Relevant to this question also is how many wines were rated from each country

```
ratings_by_country <- wine_reviews %>%
  group_by(country) %>%
  summarise(number_of_wines = n(), mean_rating = mean(points)) %>%
  arrange(desc(mean_rating))
```

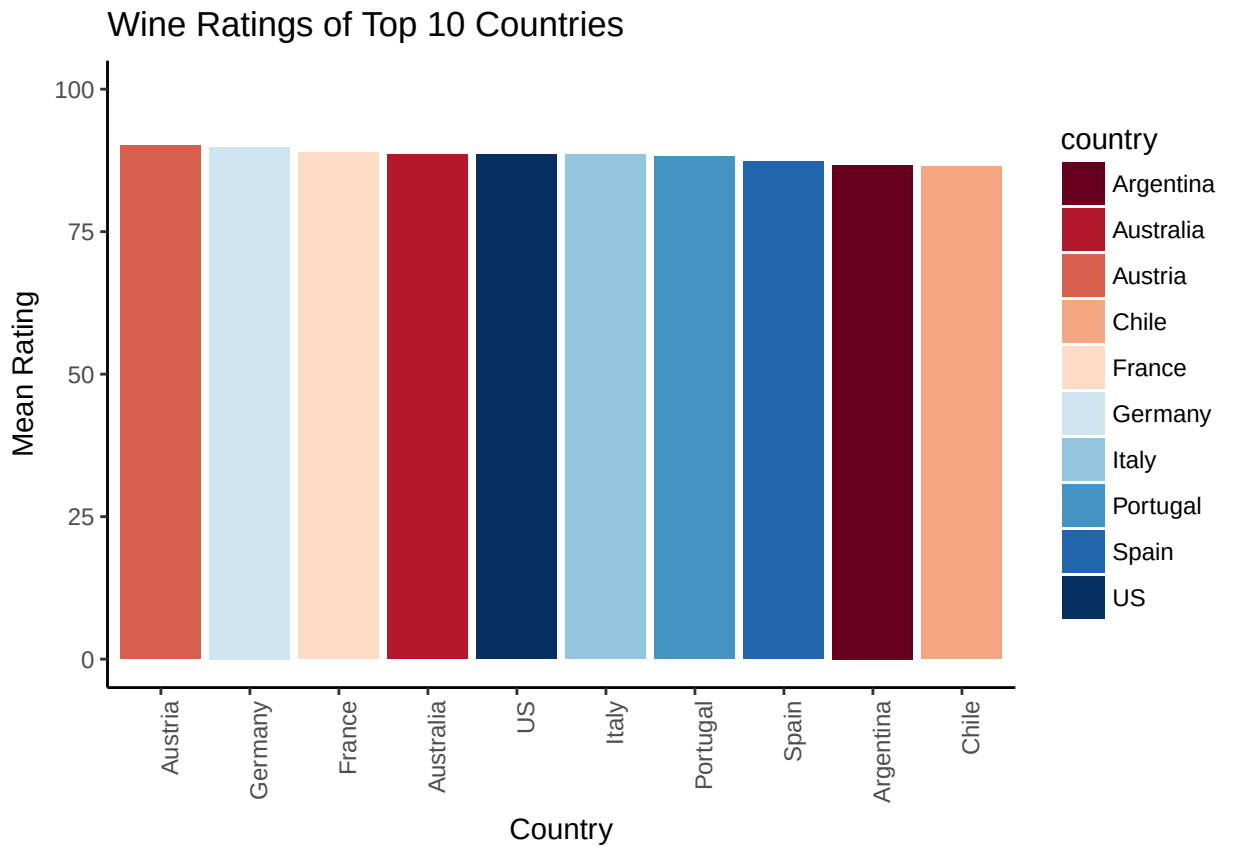
Perhaps shockingly, French wines are rated behind those from England, Austria, and Germany, although England only had 74 wines rated. These differences are, however, small. Let's visualize the ratings of the countries with the 10 most wines rated.

```r
most_wines_by_country_top_ten <- wine_reviews %>%
  group_by(country) %>%
  summarise(number_of_wines = n(), mean_rating = mean(points)) %>%
  arrange(desc(number_of_wines)) %>%
  head(10) %>%
  arrange(desc(mean_rating))
```

Among these countries we find that Austria wins out, followed by Germany and France. The differences are small though so let's see them visualized.

```r
wines_by_country_bar <- ggplot(most_wines_by_country_top_ten, aes(x = reorder(country, desc(mean_rating)

wines_by_country_bar + geom_bar(stat = "identity") +
  labs(x = "Country", y = "Mean Rating", title = "Wine Ratings of Top 10 Countries") +
  theme_classic() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  scale_fill_brewer(palette="RdBu") +
  ylim(0, 100)
```



————Start Ratings By Wine Type Exploration————

We might also want to look at the types of wines being tested. Perhaps we can figure out which country produces the best bottles of each wine.

Looking at the types of wine by sheer number:

```
wine_types <- wine_reviews %>%
  group_by(variety) %>%
  summarise(count = n()) %>%
  arrange(desc(count))
```

With over 650 types of wine, this exercise is going to become difficult. Ideally, we could find which country produces the best wine for each category. We can do this with summarise by making a vector of mean ratings for each type by country, then choosing the top country.

```
wine_types_country_ratings <- wine_reviews %>%
  group_by(variety, country) %>%
  summarise(mean_rating = round(mean(points),2)) %>%
  arrange(variety)

wine_types_country_ratings <- wine_types_country_ratings %>%
  group_by(variety) %>%
  top_n(1, mean_rating)

#If two countries are tied it will show both

wine_types_country_joined <- wine_types_country_ratings %>%
  left_join(wine_types, by = "variety") %>%
  arrange(desc(count)) %>%
  drop_na() %>%
  rename(top_country = country)
```

`wine_types_country_joined` has all varieties of wines listed with the number of wines for that variety. Also, it has the country ;isted which scored the best mean rating for all of its bottles of that variety. That top mean rating is included as well. Thus, there were 13,272 Pinot Noirs judged, and England had on average the best Pinot Noirs with a mean score of 91.86. Again shockingly, the more popular wine varieties are not dominated by France!

————Start Visualization of Wines in the World————

Sampling 10 rows (two different sizes) from the wine_reviews dataframe and plotting the location of the wines with colors of points depending on the rating. Only sampling a small amount because we can only find a limited amount of coordinates with the Google API and it takes a while to find them. You can do more, but this is just to show what's possible.

```
wine_reviews_10sample <- wine_reviews %>%
  sample_n(10) %>%
  select(country, province, points) %>%
  drop_na() %>%
  mutate_geocode(province) %>%
  drop_na() %>%
  mutate(rating_category = case_when(points <= 87 ~ "low", points > 87 & points < 94 ~ "medium", points
```

Starting the map portion

```
map <- NULL
mapWorld <- borders("world", colour="gray50", fill="gray50") # create a layer of borders
map <- ggplot() +   mapWorld

colors <- c("low" = "red", "medium" = "yellow", "high" = "green")
```

Now Layer the wines on top

```
World_Wines_Map <- map+
  geom_point(aes(x=wine_reviews_10sample$lon, y=wine_reviews_10sample$lat, colour =  factor(wine_reviews
  scale_color_manual(values = colors, name = "Rating Category", breaks=c("high","medium","low"), labels
  labs(x = "Longitude", y= "Latitude", title = "Wines Mapped Across the World")
```

————Start Visualization of Pinot Noirs in France————

Mapping the best places to get a Pinot Noir in France

1. We're going to filter the data frame to just look at Pinot Noirs from France
2. We're going to create a map of France
3. We're going to plot those wines on the map, with color depending on how high they are rated

```
reviews_pinot_france <- wine_reviews %>%
  select(country, province, region_1, variety, points) %>%
  filter(country == "France" & variety == "Pinot Noir") %>%
  mutate(rating_category = case_when(points <= 87 ~ "low", points > 87 & points < 94 ~ "medium", points
```

Don't want to have to `geocode` for each of the regions, so going to create a DF with just the unique regions and `geocode` and `join`. We can code by provinces or regions. Provinces are nice because there aren't that many of them, so we can just use `geocode()` as follows:

```
french_pinot_provinces <- reviews_pinot_france %>%
  select(province) %>%
  unique() %>%
  mutate_geocode(province) %>%
  drop_na()
```

We now have a list of the provinces and their lat/longitude. We can merge the lat/longitude with the reviews_pinot_france dataframe to create a map.

However, if we want to plot the points based on their region, it's more complex. Namely, there are more regions, so we get Query Limit Errors from the Google API when querying a lot. First we will isolate the regions as we did with the provinces, also adding an index column.

```
french_pinot_regions <- reviews_pinot_france %>%
  select(region_1) %>%
  unique()

french_pinot_regions <- mutate(french_pinot_regions, index = seq(1,nrow(french_pinot_regions)))
```

Then, we will use Shane Lynn's script which helps us get around Query Limit Errors by automatically waiting to retry geocode requests such that the Google API errors are not triggered. This code, included in the repository, outputs a "geocoded" df with latitude/longitude values (among other values) for each location based on its index. Once we run the script, we can save the geocoded csv for later so that we don't need to pull lat/longitudes from the Google server every time (it's a pain in the ass).

The following join gives us an updated region dataframe where we have all of the regions that Google was able to pinpoint (hence `dropna()`) and their coordinates

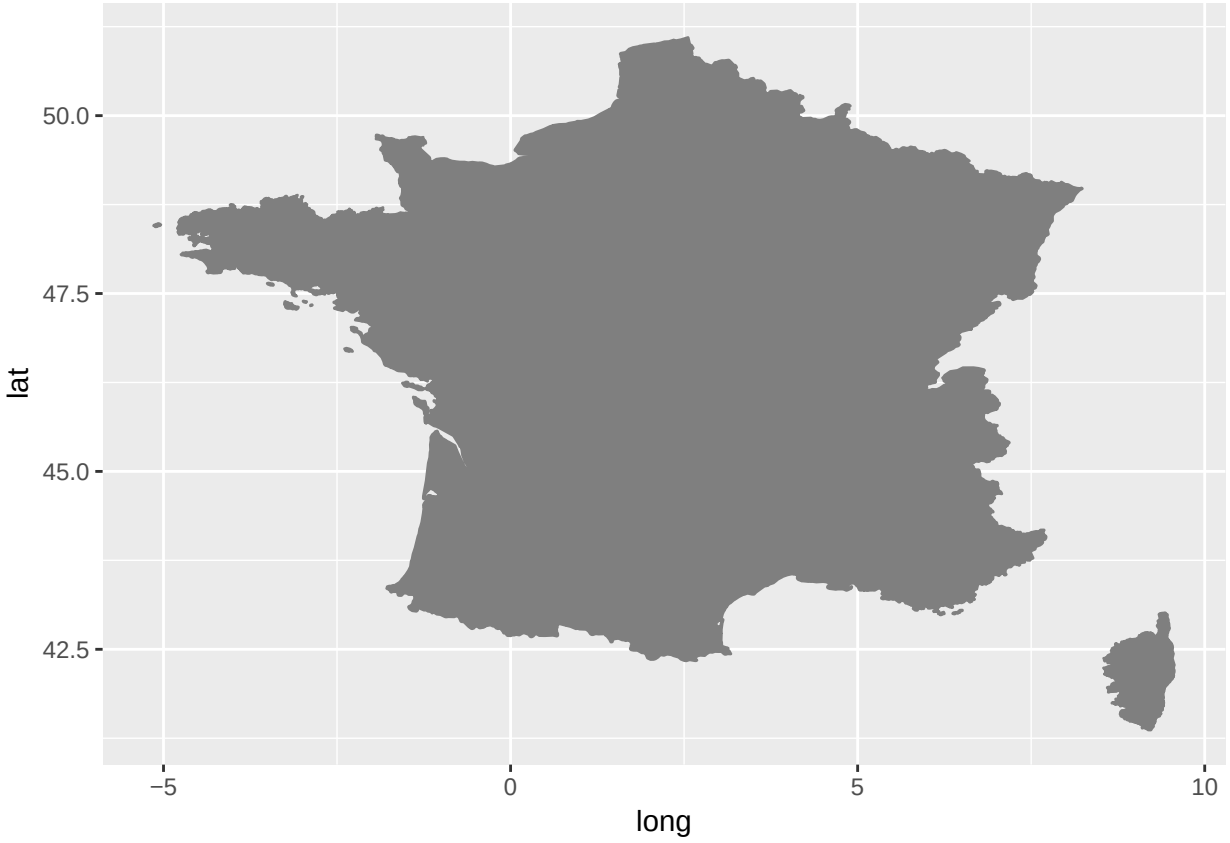
```
french_pinot_regions_withcoords <- drop_na(left_join(french_pinot_regions, geocoded_fr_regions, by = "in
```

Now we are going to take the list of all French Pinot Noirs and merge that with the coordinates for the region, retaining only the columns that we really need to make a plot.

```
reviews_pinot_france_joined <- drop_na(left_join(reviews_pinot_france, french_pinot_regions_withcoords,
reviews_pinot_france_joined <- reviews_pinot_france_joined %>%
  select(country, province, region_1, variety, points, index, lat, long, rating_category)
```

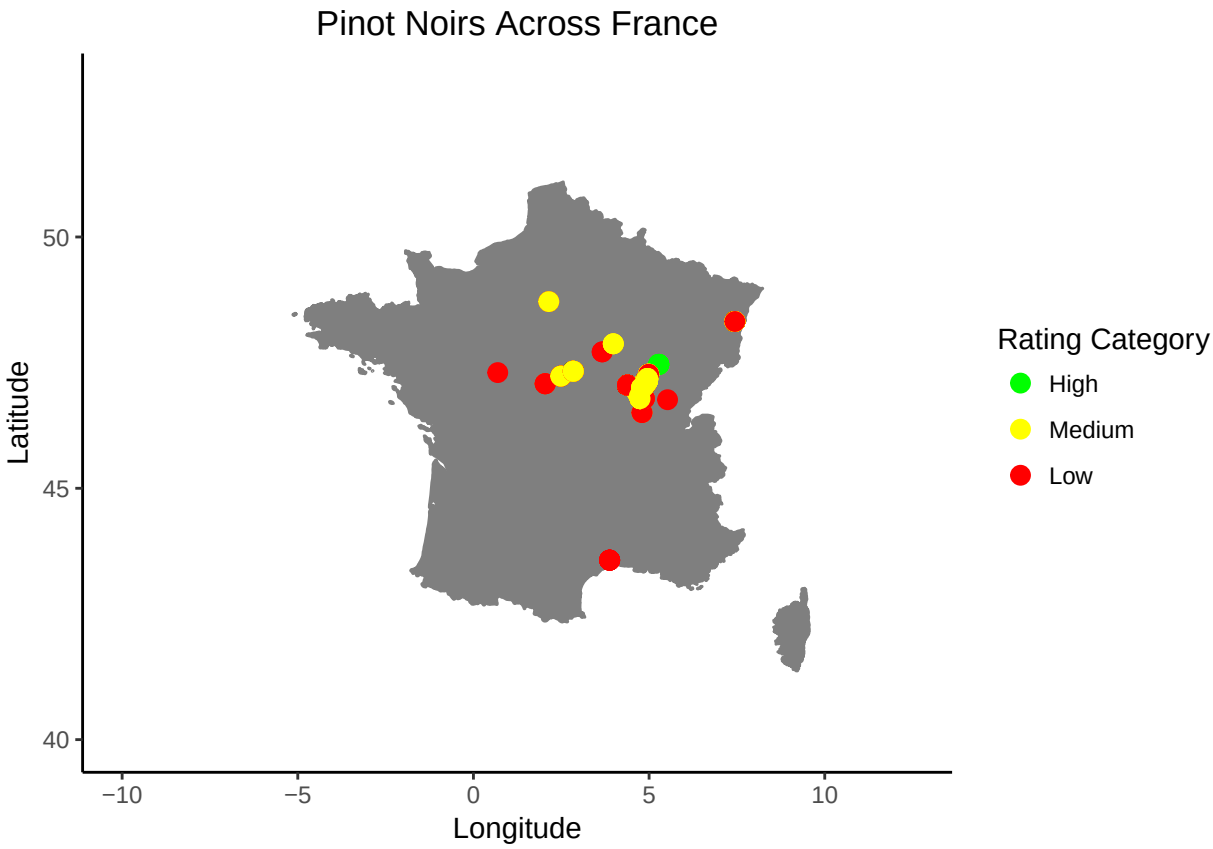
Now we plot with ggplot2 like before!

```
#literally a map of France
francemap <- NULL
map_france <- borders("france", colour="gray50", fill="gray50") # create a layer of borders
francemap <- ggplot() +   map_france
francemap
```




```
colors <- c("low" = "red", "medium" = "yellow", "high" = "green")
francemapwithpoints <- francemap +
  geom_point(aes(x=reviews_pinot_france_joined$long, y=reviews_pinot_france_joined$lat, colour =  facto
  scale_color_manual(values = colors, name = "Rating Category", breaks=c("high","medium","low"), labels
  labs(x = "Longitude", y= "Latitude", title = "Pinot Noirs Across France") +
  xlim(-10,12.5) +
  ylim(40, 53) +
  theme_classic() +
  theme(plot.title = element_text(hjust = 0.5))

francemapwithpoints
```


Pinot Noirs Across France
Latitude
50
45
40
−10
−5
0
5
10
Longitude
Rating Category
High
Medium
Low