Wine Prices Report

Corentin Lobet

9/10/2020

Introduction : data and objectives

In this research we use data of wine reviews that can be found here on Kaggle. It consists of about 130,000 wine reviews with information about the wine and the reviewer.

This dataset has largely been used to explain and predict the grade given to the wine (ranging from 80 to 100). Here we chose to model prices because they take values from a few dollars to several thousands dollars. We expect to get low precision using wine and reviewer specs though we wonder if the written reviews ('descrption') can help to explain unusually high prices.

```
head(wine_price)
```

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

```
names(wine_price)
```

```
##  [1] "X1"                    "country"               "description"
##  [4] "designation"           "points"                "price"
##  [7] "province"              "region_1"              "region_2"
## [10] "taster_name"           "taster_twitter_handle" "title"
## [13] "variety"               "winery"
```

```
summary(wine_price[, c(5,6)]) %>% tb
```

points	price
Min. : 80.0	Min. : 4
1st Qu.: 86.0	1st Qu.: 17
Median : 88.0	Median : 25

points	price
Mean : 88.5	Mean : 35
3rd Qu.: 91.0	3rd Qu.: 42
Max. :100.0	Max. :3300
NA	NA's :8996

Methodology

As we will see in the next section, this dataset needs some cleaning before we can fit models on it.

We then pick regression ML models we'll be using to predict prices. Actually we do not include the code we used to select models since it's simple training. The process took some days as we trained all regression models included in the caret package and we picked models based on 2 criteria : performance (RMSE) and computation time. As a result we kept 3 main models : MARS, CIT and GBM. We also present results of 2 other models (linear regression and decision tree) as they are fast to run and it gives us the opportunity to compare them with MARS and CIT that are examples of their adaptive versions.

Firstly, we fit prices against wine and taster specs only and analyze errors of the predictions. We then add word patterns as predictors in order to see if reviews add value using our models.

EDA and data cleaning

Basic Cleaning

The table below shows the number of distinct values taken by each variable. We can see that 'title' and 'description' have less than 129,971 different values. Reviews are reported from 43 countries for up to 1,229 regions. Grades ('points') have only 21 levels (80-100) while prices have nearly 400 levels (we will assume it is a continuous variable). The table also shows that there are as few as 19 tasters (actually it is maybe more because the 19th value is NA).

Some cleaning rules arise from these observations : remove 'designation' and 'winery' as there are too many levels and we think don't bring much information.Remove 'province' but keep 'country' and 'region_1' as we want to observe if there is a country effect and we don't want repetitive informations (indeed we assume there is a multicollinearity bias for province and country).

```
# levels = apply(wine_price, 2, function(x) { factor(x) %>% levels() %>% length() })
# save(levels, file = "data/levels.RData")
load("data/levels.RData"); levels %>% tb
```

	x
X1	129971
country	43
description	119955
designation	37979
points	21
price	390
province	425
region_1	1229
region_2	17
taster_name	19

	x
taster_twitter_handle	15
title	118840
variety	707
winery	16757

Then wa take a look at NA values. We can remove rows for NA countries and varieties (less than 100 observations), for prices (it's our independent variable), for taster name (we assume it is an important variable and it's hard to deal with NAs for this one).

```
apply(wine_price, 2, function(x) { sum(is.na(x)) }) %>% tb
```

	x
X1	0
country	63
description	0
designation	37465
points	0
price	8996
province	63
region_1	21247
region_2	79460
taster_name	26244
taster_twitter_handle	31213
title	0
variety	1
winery	0

It appears that region_1 has many NA values (>20,000). However we can see in the title variable that it often provides the region inside parentheses. We will then try to get it back from title (a variable that we won't use then by the way).

```
wine_price[1:20, c(8,12)] %>% tb
```

region_1	title
Etna	Nicosia 2013 Vulkà Bianco (Etna)
NA	Quinta dos Avidagos 2011 Avidagos Red (Douro)
Willamette Valley	Rainstorm 2013 Pinot Gris (Willamette Valley)
Lake Michigan Shore	St. Julian 2013 Reserve Late Harvest Riesling (Lake Michigan Shore)
Willamette Valley	Sweet Cheeks 2012 Vintner's Reserve Wild Child Block Pinot Noir (Willamette Valley)
Navarra	Tandem 2011 Ars In Vitro Tempranillo-Merlot (Navarra)
Vittoria	Terre di Giurfo 2013 Belsito Frappato (Vittoria)
Alsace	Trimbach 2012 Gewurztraminer (Alsace)
NA	Heinz Eifel 2013 Shine Gewürztraminer (Rheinhessen)
Alsace	Jean-Baptiste Adam 2012 Les Natures Pinot Gris (Alsace)
Napa Valley	Kirkland Signature 2011 Mountain Cuvée Cabernet Sauvignon (Napa Valley)

region_1	title
Alsace	Leon Beyer 2012 Gewurztraminer (Alsace)
Alexander Valley	Louis M. Martini 2012 Cabernet Sauvignon (Alexander Valley)
Etna	Masseria Setteporte 2012 Rosso (Etna)
Central Coast	Mirassou 2012 Chardonnay (Central Coast)
NA	Richard Böcking 2013 Devon Riesling (Mosel)
Cafayate	Felix Lavaque 2010 Felix Malbec (Cafayate)
Mendoza	Gaucho Andino 2011 Winemaker Selection Malbec (Mendoza)
Ribera del Duero	Pradorey 2010 Vendimia Seleccionada Finca Valdelayegua Single Vineyard Crianza (Ribera del Duero)
Virginia	Quiévremont 2012 Meritage (Virginia)

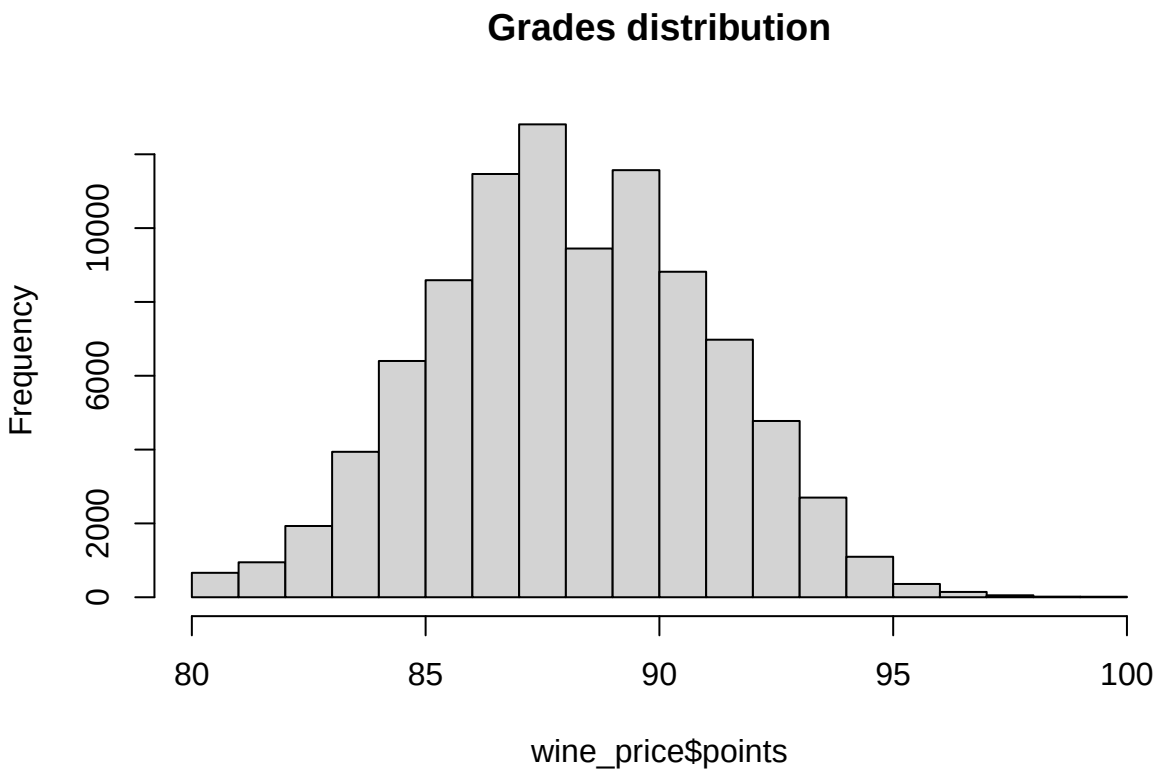
We are now ready for this first cleaning and variable picking step. The code below provides this data manipulation

```
# Select columns and remove NAs
wine_price = wine_price %>%
   select(country, description, points, price,
          region_1, taster_name, variety, title) %>%
   filter(!is.na(price), !is.na(variety), !is.na(country), !is.na(taster_name)) %>%
   rename(region = region_1)
# Add regions from title
wine_price = wine_price %>%
   filter(!is.na(region) | str_detect(title, "\\(([:print:]+)\\)")) %>%
   mutate(region = ifelse(is.na(region), str_extract(title, "\\(([:print:]+)\\)"), region)) %>%
   mutate(region = str_remove_all(region, "\\(|\\)")) %>%
   filter(str_count(region) < 25)
wine_price = select(wine_price, -title)
```

EDA and further data pre-processing

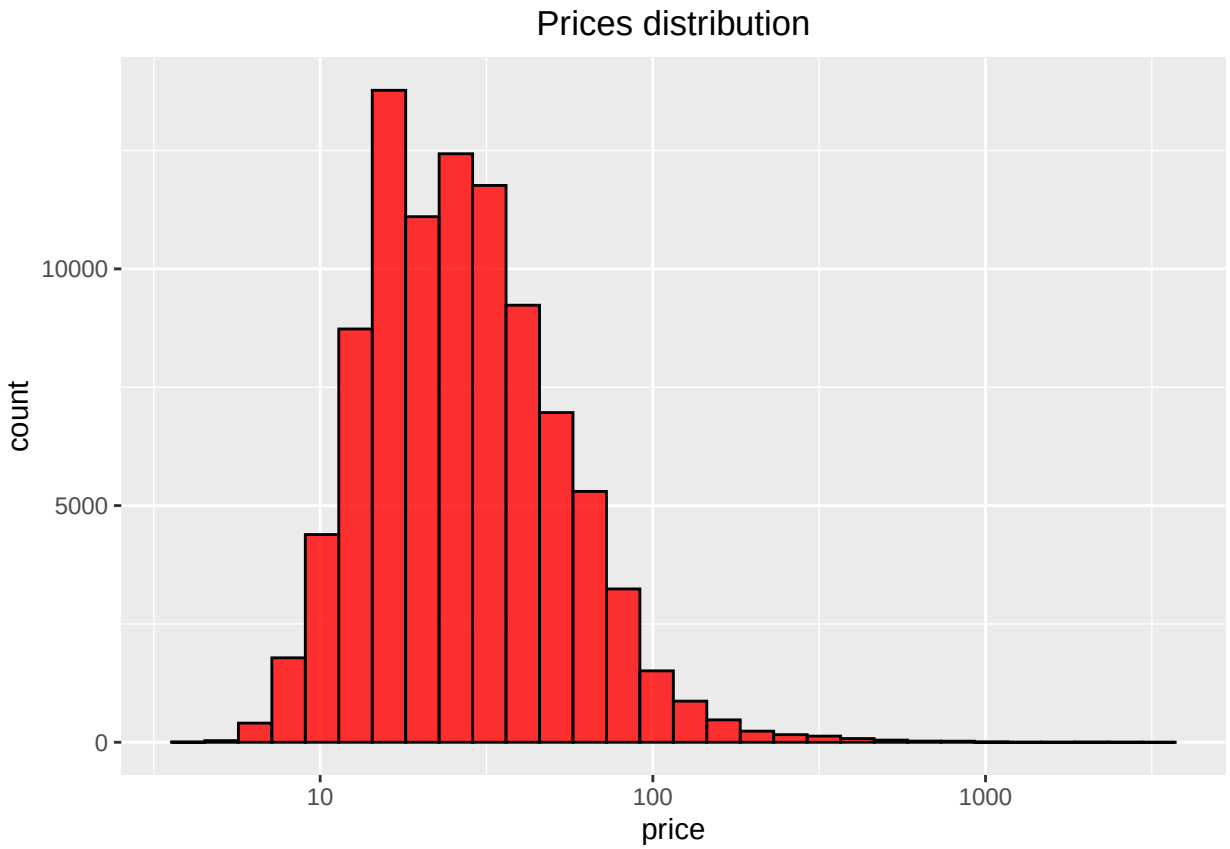
We can notice that prices distribution seems to be skewed (because of outliers) while ratings' scores distribution looks Gaussian. This fact makes us expect that the score won't explain well the prices. That's why we look for other variables impact.

```
# Ratings distribution
hist(wine_price$points, main = "Grades distribution")
```



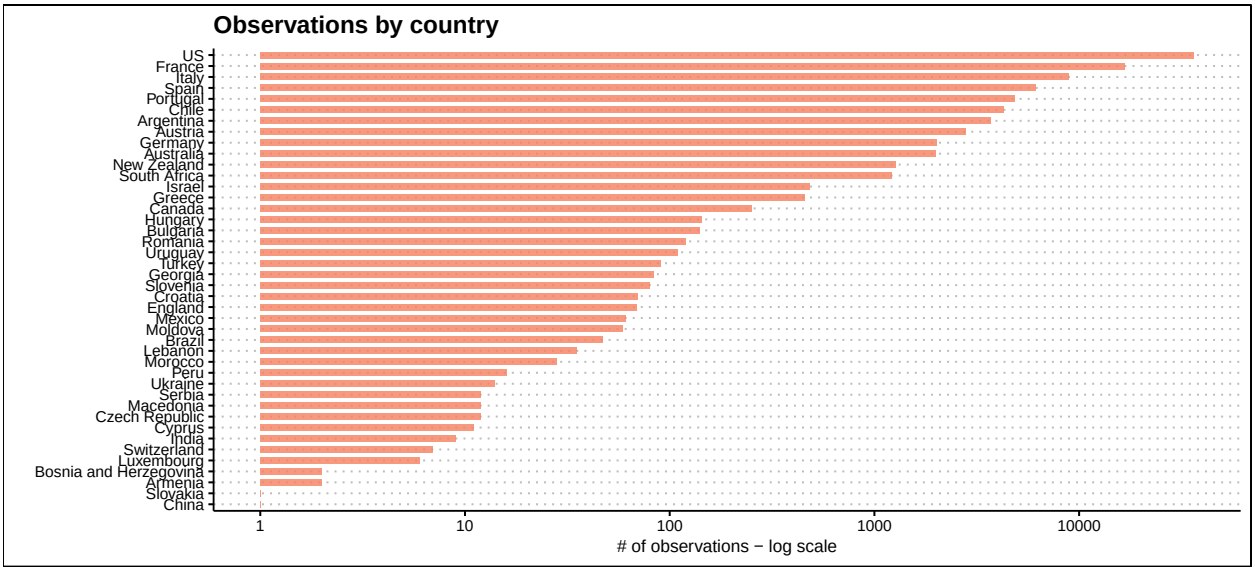

```
# Price distribution
ggplot(wine_price, aes(price)) +
   geom_histogram(color = "black", fill = "red", alpha = 0.8) +
   scale_x_log10() +
   ggtitle("Prices distribution") +
   theme(plot.title = element_text(hjust = 0.5))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
wine_price %>%
   group_by(country) %>%
   summarise(n = n()) %>%
   mutate(country = reorder(country, n)) %>%
   ggplot(aes(y = country, x = n)) +
   geom_bar(stat = "identity", fill = "#f68060", alpha = 0.8, width = 0.6) +
   scale_x_log10() +
   xlab("# of observations - log scale") +
   ylab("") + ggtitle("Observations by country") +
   ggthemes::theme_clean()
```

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



Fitting models on basic predictors

Implementing descriptions as a predictor

Fitting models with word patterns

Conclusion