EDA + Visualziation + Text Mining for Wine Products

Tianze Hua

3/5/2022

```
library(pacman)
p_load(tidyverse, kableExtra, magrittr,
       knitr, ggrepel, ggwordcloud, readr, tm,
       SnowballC, wordcloud,RColorBrewer, wordcloud2, sjPlot)
search()
```

```
##  [1] ".GlobalEnv"           "package:sjPlot"       "package:wordcloud2"
##  [4] "package:wordcloud"    "package:RColorBrewer" "package:SnowballC"
##  [7] "package:tm"           "package:NLP"          "package:ggwordcloud"
## [10] "package:ggrepel"      "package:knitr"        "package:magrittr"
## [13] "package:kableExtra"   "package:forcats"      "package:stringr"
## [16] "package:dplyr"        "package:purrr"        "package:readr"
## [19] "package:tidyr"        "package:tibble"       "package:ggplot2"
## [22] "package:tidyverse"    "package:pacman"       "package:stats"
## [25] "package:graphics"     "package:grDevices"    "package:utils"
## [28] "package:datasets"     "package:methods"      "Autoloads"
## [31] "package:base"
```

```
md <-read.csv("Wine_tasting.csv",sep=",", na.strings ="" )
```

```
glimpse(md)
```

```
## Rows: 1,000
## Columns: 14
## $ X                     <int> 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14~
## $ country               <chr> "Italy", "Portugal", "US", "US", "US", "Spain", ~
## $ description           <chr> "Aromas include tropical fruit, broom, brimstone~
## $ designation           <chr> "Vulk√† Bianco", "Avidagos", NA, "Reserve Late H~
## $ points                <int> 87, 87, 87, 87, 87, 87, 87, 87, 87, 87, 87, 87, ~
## $ price                 <int> NA, 15, 14, 13, 65, 15, 16, 24, 12, 27, 19, 30, ~
## $ province              <chr> "Sicily & Sardinia", "Douro", "Oregon", "Michiga~
## $ region_1              <chr> "Etna", NA, "Willamette Valley", "Lake Michigan ~
## $ region_2              <chr> NA, NA, "Willamette Valley", NA, "Willamette Val~
## $ taster_name           <chr> "Kerin O,ÄôKeefe", "Roger Voss", "Paul Gregutt",~
## $ taster_twitter_handle <chr> "@kerinokeefe", "@vossroger", "@paulgwine¬†", NA~
## $ title                 <chr> "Nicosia 2013 Vulk√† Bianco  (Etna)", "Quinta do~
## $ variety               <chr> "White Blend", "Portuguese Red", "Pinot Gris", "~
## $ winery                <chr> "Nicosia", "Quinta dos Avidagos", "Rainstorm", "~
```

Count the levels of character columns, range of interger columns, count number of NAs

```
# count levels of all the character columns
md %>%
```

```
  select(-"X") %>%
  select_if(is.character) %>%
  mutate_all(as.factor) %>%
  map(levels) %>%
  map(length)
```

```
## $country
## [1] 18
##
## $description
## [1] 1000
##
## $designation
## [1] 669
##
## $province
## [1] 99
##
## $region_1
## [1] 269
##
## $region_2
## [1] 17
##
## $taster_name
## [1] 16
##
## $taster_twitter_handle
## [1] 13
##
## $title
## [1] 999
##
## $variety
## [1] 137
##
## $winery
## [1] 868
```

```
md %>%
  select(-X) %>%
  select_if(is.character) %>%
  summarize_all(funs(sum(is.na(.)))) -> lvl2

# count the range of each integer columns
md %>%
  select_if(is.integer) %>%
  na.omit() %>%
  lapply(range)
```

```
## $X
## [1]   1 999
##
## $points
```

Table 1: NA proportion of Each Character Columns

	Number_of_NA	NA_prop
country	1	0.1%
designation	247	24.7%
province	1	0.1%
region_1	165	16.5%
region_2	633	63.3%
taster_name	206	20.6%
taster_twitter_handle	252	25.2%

```
## [1]  80 100
##
## $price
## [1]   7 775
```

```
# count the NA and blank fields in each column
md %>%
  select(-X) %>%
  summarize_all(funs(sum(is.na(.)))) %>%
  t() %>%
  as.data.frame() %>%
  filter(V1>0) %>%
  rename(Num_of_NA = V1) ->s2

# showing missing proportion of each column of our dataset
lvl2 %>%
  t() %>%
  as.data.frame() %>%
  filter(V1 >0) %>%
  mutate(NA_prop = paste0(100*round(V1/1000, 5), "%", sep='')) %>%
  rename(Number_of_NA = "V1") %>%
  kbl(caption = "NA proportion of Each Character Columns") %>%
  kable_classic_2(full_width = F,
                  html_font = "Cambria")
```

Data Cleanning

```
# We only going to remove the observation without country, price field.

md %<>%
  drop_na(country, price)

md %<>%
  select(-X)

# remove all the foreign characters of the entire dataset for better understanding
md %<>%
  mutate_all(funs(gsub("[[:punct:]]", "", .)))


dim(md)
```

```
## [1] 942  13
```

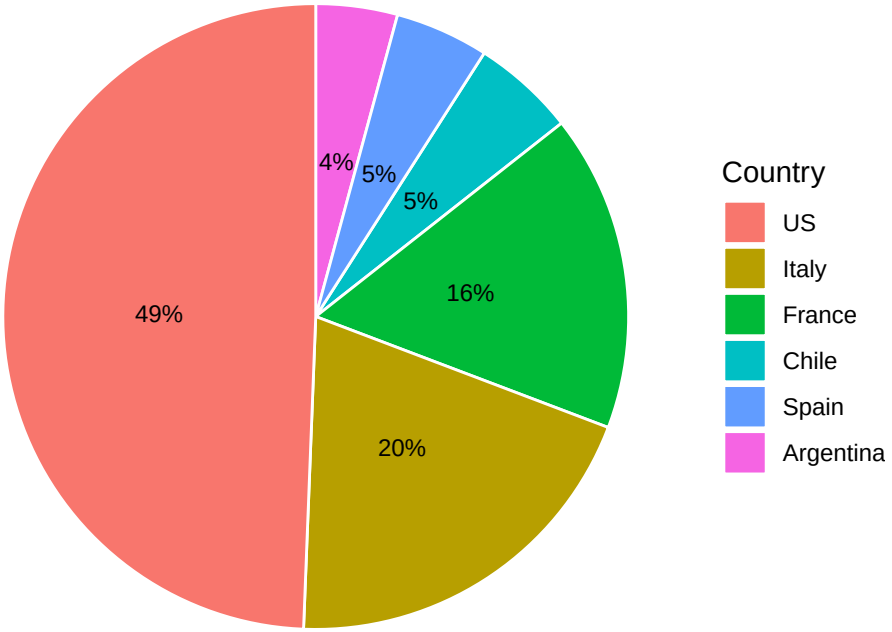
```
# Don't forget to change chr price to numeric one
md$price <- as.numeric(md$price)
md$points <- as.numeric(md$points)
```

Country Factor

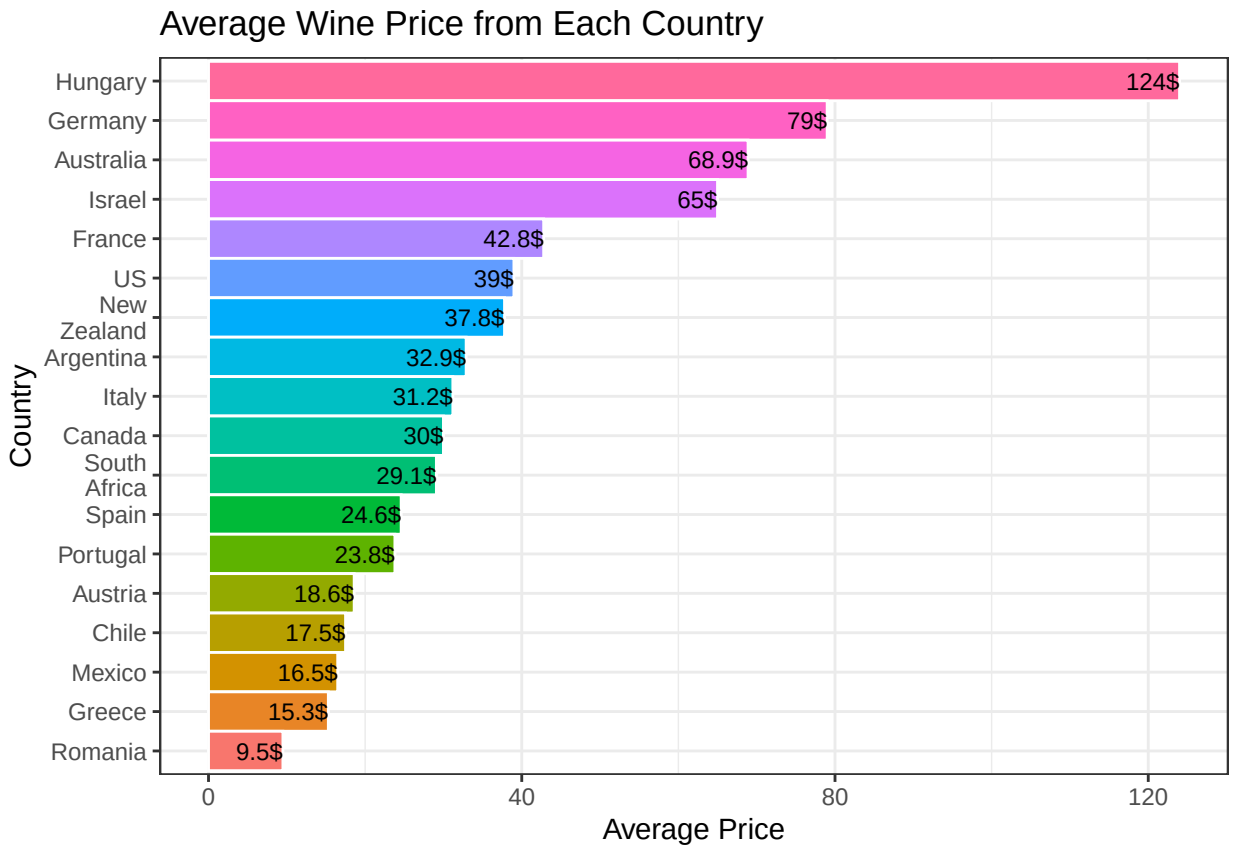
```
# market share by countries
md %>%
  select(country,region_1) %>%
  group_by(country) %>%
  summarize(n=n()) %>%
  filter(n>30)  %>%
  mutate(country = fct_reorder(country, desc(n))) %>%
  ggplot(aes('',n, fill=country))+
  geom_bar(stat="identity",width=1, color = "white")+
  theme(axis.ticks.y = element_blank(),
      axis.text.y = element_blank(),
      axis.text.x = element_blank())+
  coord_polar(theta="y")+
  scale_fill_brewer(palette = "Set3")+
  theme_void()+
  geom_text(aes(label=paste(round(100*n/sum(n),0),"%",sep="")),
            position = position_stack(vjust = 0.5),size=3)+
  coord_polar(theta = "y")+
  labs(title = 'Market Share by Country',
       subtitle = "Only showing countries with more than 30 wine products")+
  scale_fill_discrete(name = "Country")+
  theme(plot.title = element_text(size = 13),
        plot.subtitle = element_text(size = 8))
```

Market Share by Country

Only showing countries with more than 30 wine products




```
# mean price by country
md %>%
  select(country, price) %>%
  group_by(country) %>%
  summarize(price = round(mean(price),1)) %>%
  arrange(desc(as.numeric(price))) -> md_cty

md_cty %>%
  mutate(country = fct_reorder(country, price)) %>%
  ggplot(aes(country, price, fill = country))+
  geom_bar(stat="identity", width=1, color = "white")+
  theme_bw()+
  theme(legend.position = "none")+
  scale_x_discrete(
    labels = function(country) str_wrap(country, width = 7))+
  geom_text(aes(label=paste(price,"$", sep="")), size=3,hjust=1)+
  coord_flip()+
  labs(x="Country",
       y="Average Price",
       title = "Average Wine Price from Each Country ")
```




```
md %>%
  select(country) %>%
  group_by(country) %>%
  summarize(n =n()) %>%
  arrange(desc(n)) -> mdx

mdx <- c(mdx[1:5,1])
md %>%
  select(country, price) %>%
  filter(country %in% unlist(mdx),
         price <100) %>%
  mutate(country = fct_reorder(country, desc(price))) %>%
  ggplot(aes(country, price, fill=country))+
  geom_boxplot()+
  geom_jitter(size = 0.4, alpha = 0.7, color ="#7468de" )+
  theme_bw()+
  scale_fill_brewer(palette = "Set3")+
  labs(title="Boxplot for Wine Price in Each Country",
       y = "Price",
       X = "Country")+
  theme(legend.position="none")
```


```
# Confidence interval for all over the world and US wine mean price
md %>%
  lm(price~1,.) %>%
  confint(level=0.99) %>%
  kbl(caption = "99% Confidence Interval for Average Wine Price") %>%
  kable_classic_2(full_width = F,html_font = "Cambria")
```

\begin{table}

\caption{99% Confidence Interval for Average Wine Price}

	0.5 %	99.5 %
(Intercept)	33.37792	41.34182

\end{table}

```
md %>%
  filter(country == "US") %>%
  lm(price~1,.) %>%
  confint(level = 0.99) %>%
  kbl(caption = "99% Confidence Interval for Average Wine Price in US") %>%
  kable_classic_2(full_width = F,html_font = "Cambria")
```

\begin{table}

\caption{99% Confidence Interval for Average Wine Price in US}

	0.5 %	99.5 %
(Intercept)	35.78934	42.23579

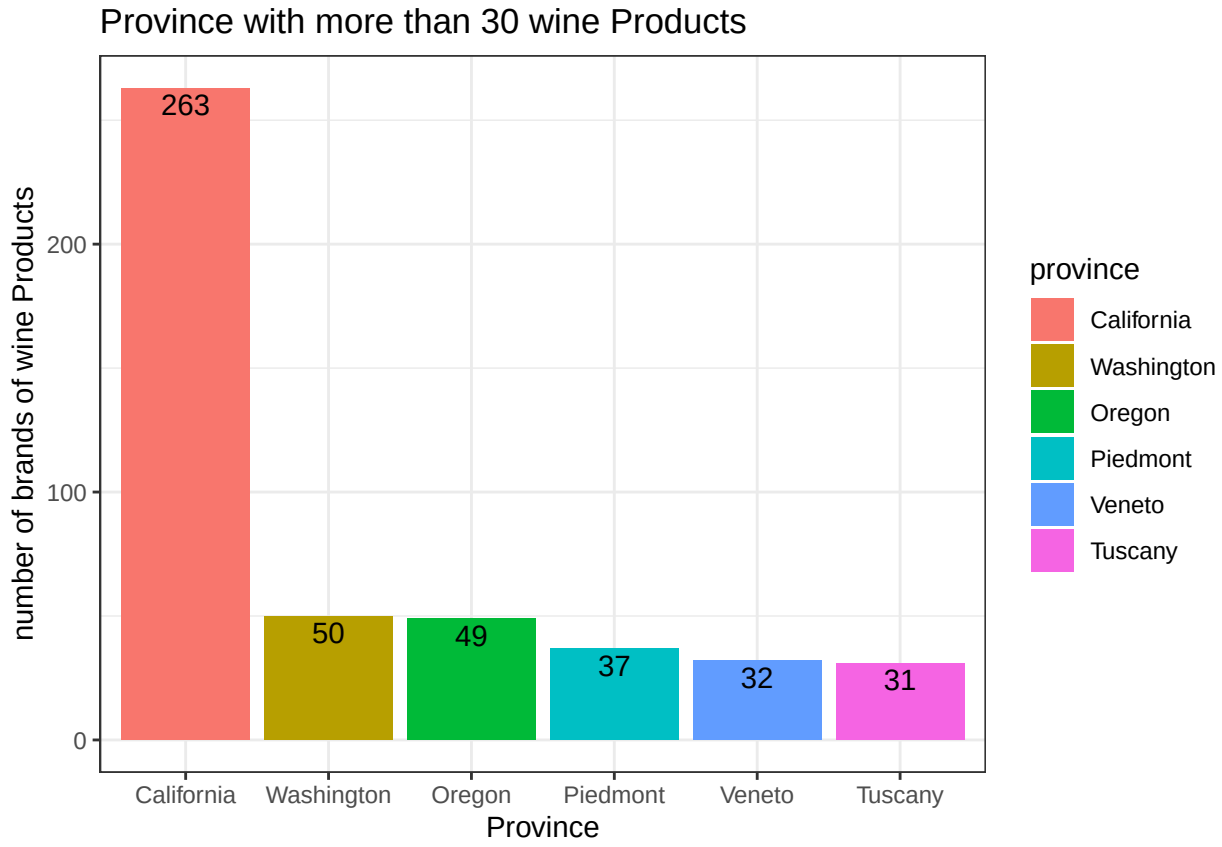
\end{table} we are at a 99% confidence interval to state that the mean price of wine of US would lie between 35.8 and 42.2 dollars, and that for the world is between 33.38 and 41.34.

Province VS Number of Products

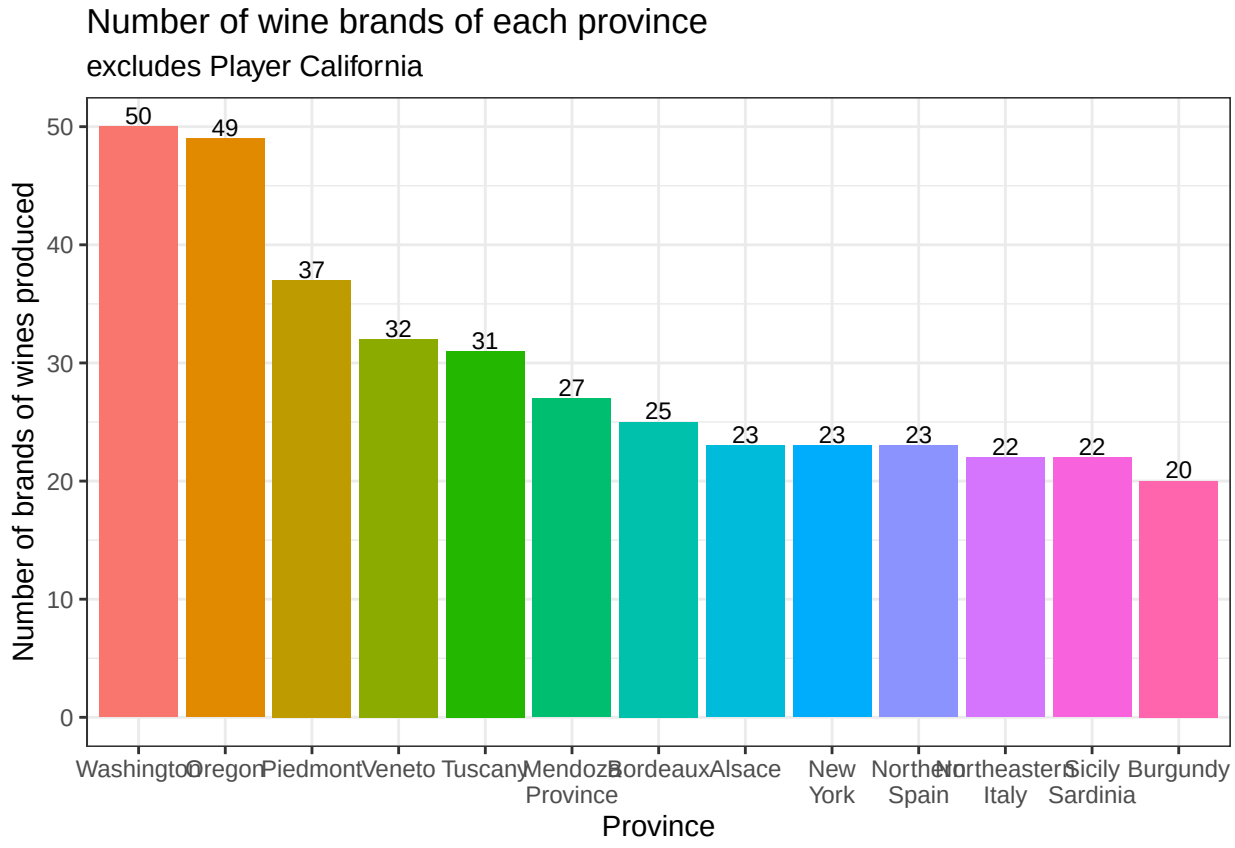
```
colnames(md)
```

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

```
# Province with more than 30 wine brands
md %>%
  select(province) %>%
  group_by(province) %>%
  summarise(n = n()) %>%
  arrange(desc(n)) %>%
  filter(n >30) %>%
  mutate(province = fct_reorder(province,desc(n))) %>%
  ggplot(aes(province, n, fill=province)) +
  geom_bar(stat="identity") +
  theme_bw() +
  labs(x = "Province",
       y = "number of brands of wine Products",
       title = "Province with more than 30 wine Products")+
  geom_text(aes(label = n), vjust = 1.3)
```



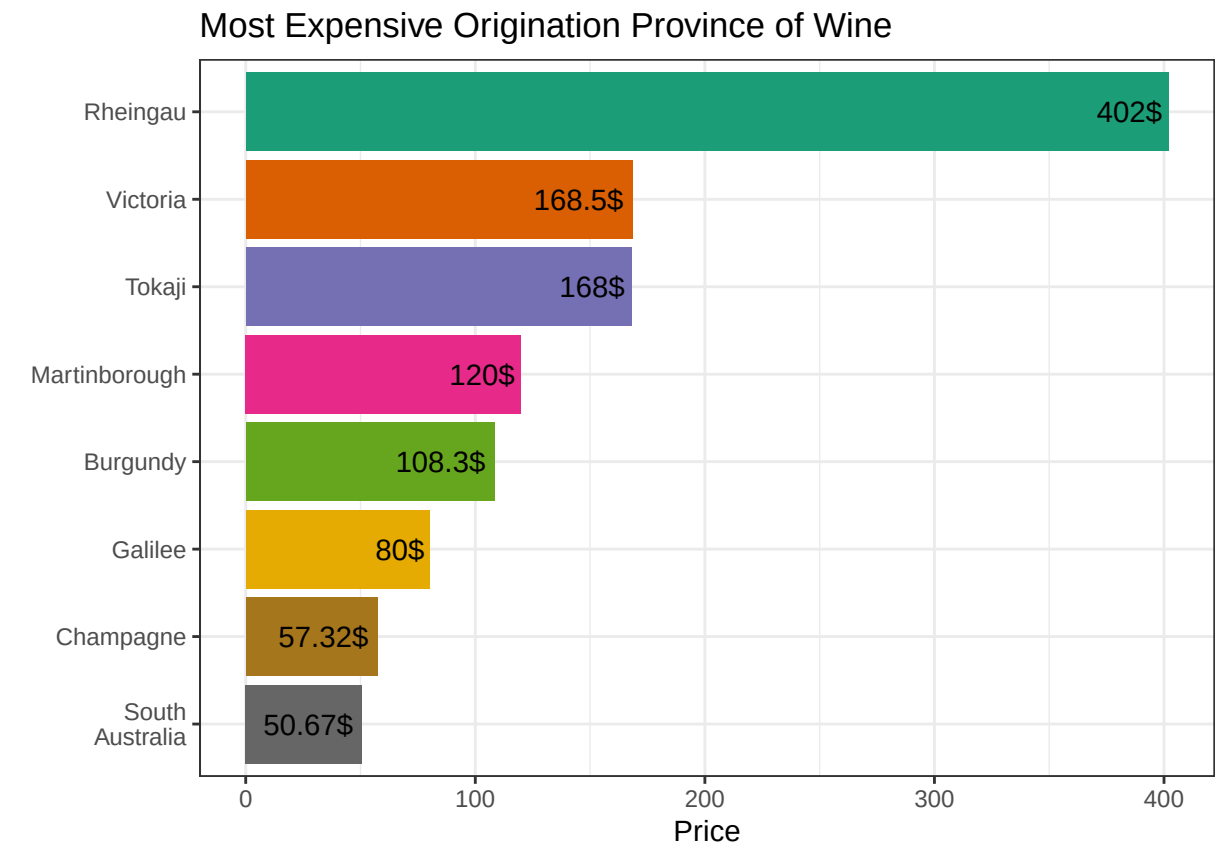
```r
# if California is outstreched the scale a little bit too much
md %>%
  select(province) %>%
  group_by(province) %>%
  summarise(n = n()) %>%
  arrange(desc(n)) %>%
  filter(n > 19, n < 60) %>%
  mutate(province = fct_reorder(province,desc(n))) %>%
  ggplot(aes(province, n, fill=province)) +
  geom_bar(stat="identity") +
  theme_bw() +
  labs(x = "Province",
       y = "Number of brands of wines produced",
       title = "Number of wine brands of each province",
       subtitle = "excludes Player California")+
  geom_text(aes(label = n), vjust = -0.15, size = 3)+
  scale_x_discrete(
    labels = function(province) str_wrap(province, width = 2))+
  theme(legend.position = "none")
```



Province vs AVG price

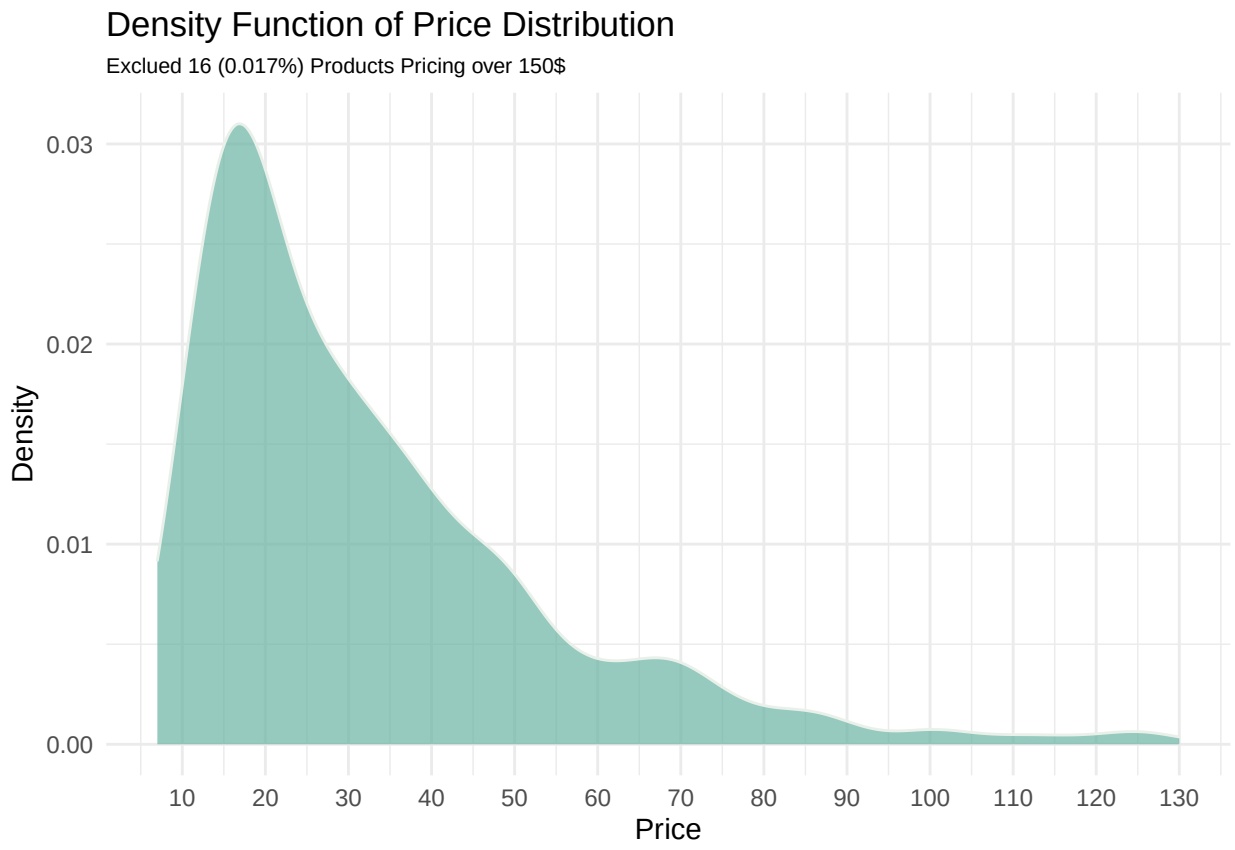
```r
md %>%
  select(province, price) %>%
  na.omit() %>%
  group_by(province) %>%
  summarize(province_avg = sum(price)/n()) %>%
```

```
  arrange(desc(province_avg)) -> mad2

mad2 <- c(mad2[1:8,1])

md %>%
  select(province, price) %>%
  na.omit() %>%
  group_by(province) %>%
  summarize(province_avg = sum(price)/n()) %>%
  filter(province %in% unlist(mad2)) %>%
  arrange(desc(province_avg)) %>%
  mutate(province = fct_reorder(province, province_avg)) %>%
  ggplot(aes(province, province_avg, fill = province))+
  geom_bar(stat="identity")+
  theme_bw()+
  labs(x = "",
       y = "Price",
       title = "Price of wine")+
  scale_fill_brewer(palette = "Dark2", direction = -1)+
  scale_x_discrete(labels = function(country) str_wrap(country, width = 7))+
  coord_flip()+
  theme(legend.position = "none")+
  geom_text(aes(label=paste(round(province_avg,2),"$", sep="")),hjust=1.1)+
  labs(title = "Most Expensive Origination Province of Wine")
```




```
md %>%
  select(price) %>%
```

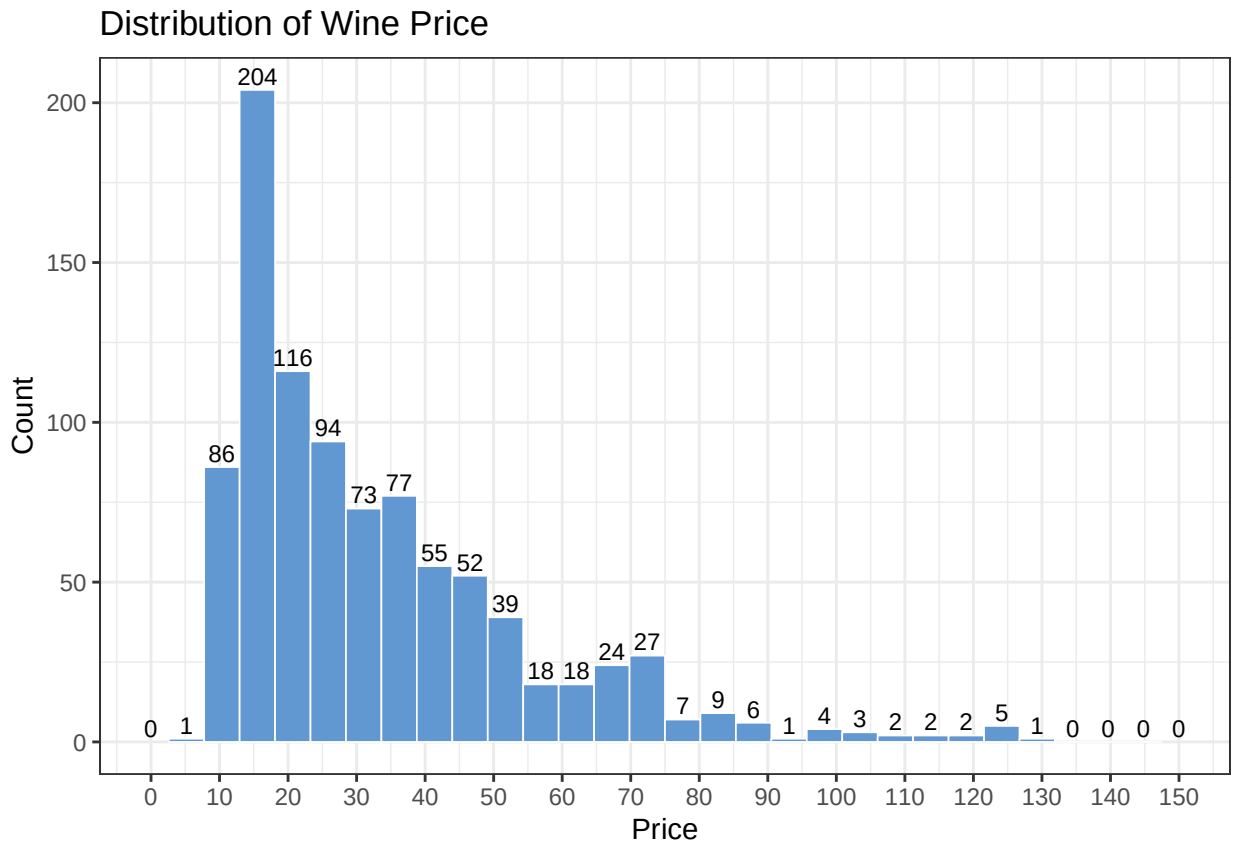
```
  filter(price<150) %>%
  ggplot(aes(price))+
  geom_density(fill="#69b3a2",
               color="#e9f0e9",
               alpha=0.7)+
  theme_minimal()+
  labs(x = "",
       y = "price",
       title = "Price of wine")+
  scale_x_continuous(breaks=round(seq(0,160, by = 10),1))+
  labs(title = "Density Function of Price Distribution",
       subtitle = "Exclued 16 (0.017%) Products Pricing over 150$",
       x = 'Price',
       y = 'Density')+
  theme(plot.title = element_text(size = 13),
        plot.subtitle = element_text(size = 8))
```



Distribution of Price along with correlation between Price and Points

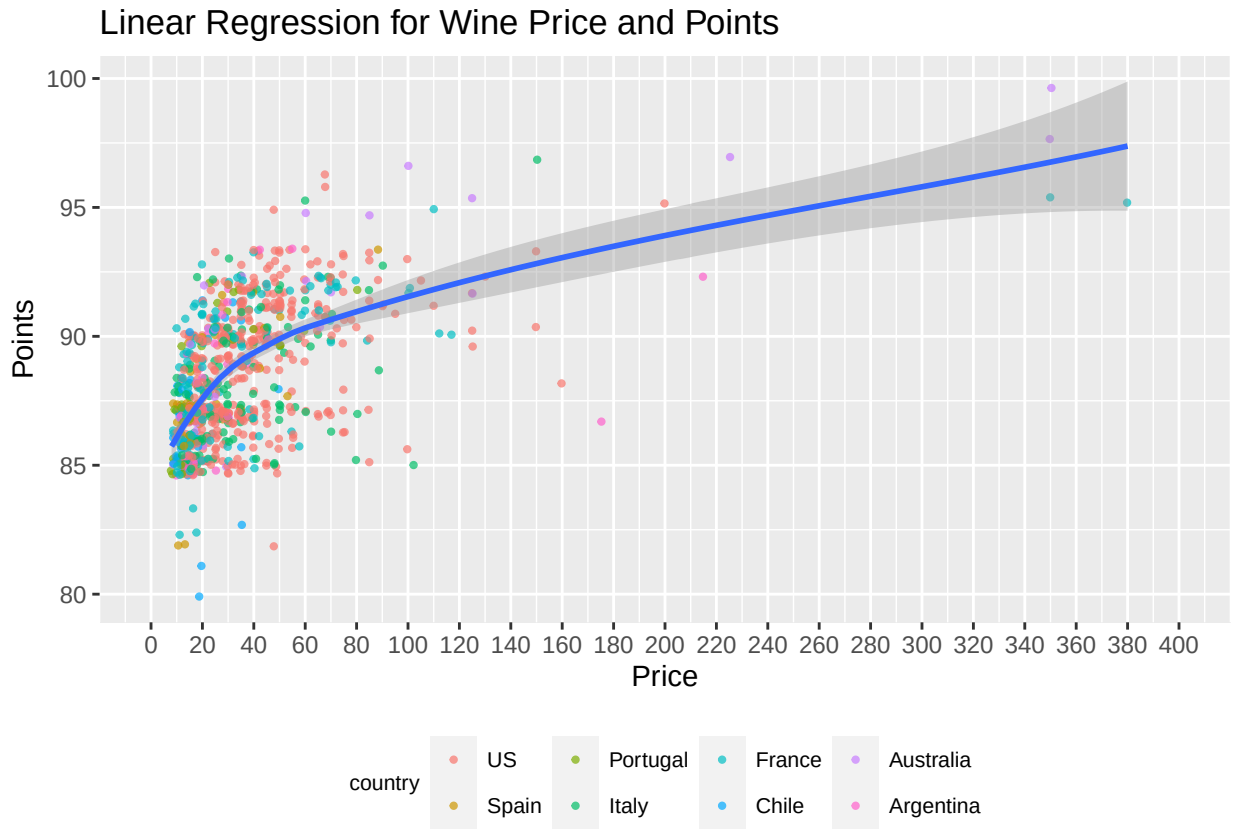
```
md %>%
  select(country) %>%
  group_by(country) %>%
  summarize(n =n()) %>%
  arrange(desc(n)) -> mdx8

mdx8 <- c(mdx8[1:8,1])
```

```
md %>%
  select(price, points) %>%
  filter(price<150) %>%
  na.omit() %>%
  ggplot(aes(price))+
  geom_histogram(bins=30, fill = "#6298d1", colour="white", lwd = 0.3)+
  labs(title="Distribution of Wine Price",
       x = "Price",
       y = "Count")+
  stat_bin(geom='text', color='black',
           aes(label=..count..),
           size =3,vjust=-0.3)+
  theme_bw()+
  scale_x_continuous(breaks= seq(0,150,10),limit=c(0,150,15))
```




```
md %>%
  select(price, points, country) %>%
  na.omit() %>%
  filter(country %in% unlist(mdx8)) %>%
  mutate(country = fct_reorder(country,desc(country))) %>%
  ggplot(aes(price, points))+
  geom_jitter(aes(colour=country), alpha=0.7, size = 0.8)+
  theme(legend.position = "bottom",
        legend.title = element_text(size = 8),
        legend.text = element_text(size = 8))+
  geom_smooth()+
  labs(title="Linear Regression for Wine Price and Points",
```

```
      y = "Points",
      x = "Price")+
  scale_x_continuous(breaks= seq(0,400,20),limit=c(0,400,20))
```

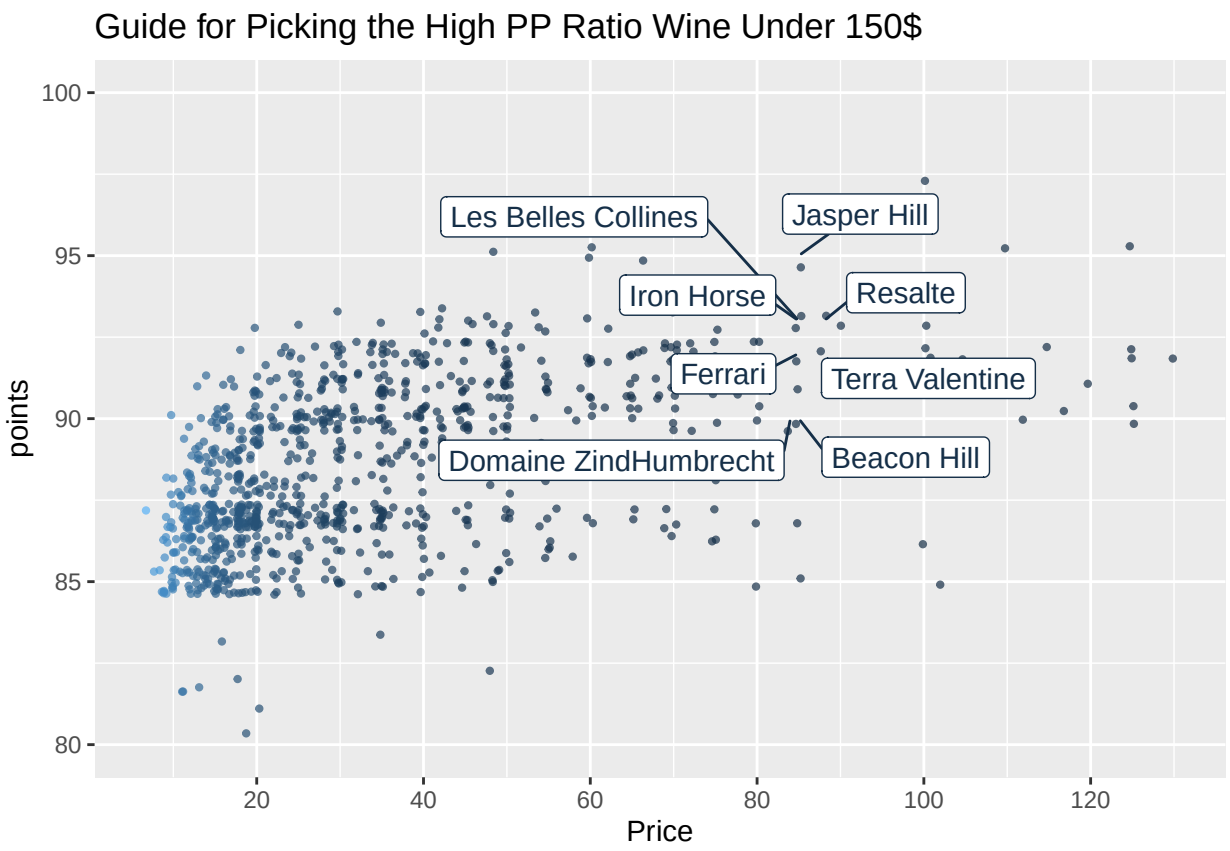



```
lm1 <- lm(points~price, data=md)
summary(lm1)
```

```
##
## Call:
## lm(formula = points ~ price, data = md)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.212  -1.657  -0.072   1.768   6.746
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 87.593078   0.096567  907.08   <2e-16 ***
## price        0.026605   0.001602   16.61   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.326 on 940 degrees of freedom
## Multiple R-squared:  0.2269, Adjusted R-squared:  0.2261
## F-statistic:   276 on 1 and 940 DF,  p-value: < 2.2e-16
```

```
md %>%
  select(price, points, winery, title) %>%
```

```
  filter(price<150) %>%
  na.omit() %>%
  mutate(ratio = points/price) %>%
  ggplot(aes(price, points, colour=ratio))+
  geom_jitter(alpha=0.7, size = 0.8)+
  theme(legend.position = "none")+
  geom_label_repel(aes(label = ifelse(ratio>1.05 & price > 80, winery, NA)),
                   box.padding = 0.5,max.overlaps = 90)+
  labs(title="Guide for Picking the High PP Ratio Wine Under 150$",
       y = "points",
       x = "Price") +
  theme(plot.title = element_text(size = 13),
      plot.subtitle = element_text(size = 8))+
  scale_x_continuous(breaks=round(seq(0,160, by = 20),1))+
  scale_y_continuous(limit=c(80,100,5))
```



Data selection for Word Mining

```
md %>%
  filter(points/price> 0.8) -> md_good
corpus = Corpus(VectorSource(md_good$description))
corpus
```

```
## <<SimpleCorpus>>
## Metadata:  corpus specific: 1, document level (indexed): 0
## Content:  documents: 917
```

```
md_good$description %>%
  VectorSource() %>%
  Corpus -> corpus
```

Word Processing

```
corpus %>%
  tm_map(PlainTextDocument) %>%
  tm_map(tolower) %>%
  tm_map(removePunctuation) %>%
  tm_map(stemDocument, language = "english") %>%
  tm_map(stripWhitespace) -> corpus

corpus <- tm_map(corpus,removeWords, stopwords("english"))
```

```
corpus %>%
  TermDocumentMatrix() %>%
  as.matrix() %>%
  rowSums() %>%
  sort(decreasing = TRUE) -> mat1

word_f <- data.frame(word = names(mat1), freq=mat1)
```

Keyword for Picking the Right Wine

```
word_f %>%
  filter(word != "wine") %>%
  wordcloud2()
```

```
## QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-rstudio-user'
## TypeError: Attempting to change the setter of an unconfigurable property.
## TypeError: Attempting to change the setter of an unconfigurable property.
```

```
# it does not show in a pdf output
```