Code Extension

Alain T Kuiete

12/5/2019

Introduction

In this assignment, we are looking at tidyverse which has collection of R packages that can help us loading the dataset to R, cleaning, transforming and visuzalizing of the data.The goal of this assignment is to create a sample dataset that shows the capabilities of tidyverse with and example dataset. The example dataset I selected is "Wine Data" from Kaggle. Since, we are selecting an example dataset, we might as well select an example business obsective. The business problem in question that I choose to answer at the end of this analysis is "What are the top ranked wines from US?" Can we determine what type of wine we can select based on their origin?".

About the Data Set

The data set we chose for this assignment is wine reviews. The variable descriptions are outlined below

- Country: The country that the wine is from.
- Description: The description of the variable.
- Designation: The vineyard within the winery where the grapes that made the wine are from.
- Points: The number of points Wine Enthusiast 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), but this value can sometimes be blank

Loading Tidyverse

```r
library(tidyverse)
```

```
## -- Attaching packages ----------------------------------------------- tidyverse 1.2.1 --
```

```
## v ggplot2 3.2.1     v purrr   0.3.2
## v tibble  2.1.3     v dplyr   0.8.3
## v tidyr   0.8.3     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.4.0
## -- Conflicts -------------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

When we install and load the tidyverse, we see that we loaded below packages

- ggplot2
- tibble
- tidyr
- readr
- purr
- dplyr
- stringr
- forcats

Data Collection

We can use read_csv function from readr package within tidyverse to read the data from csv.

```
wine <- read_csv("https://raw.githubusercontent.com/anilak1978/tidyverse/master/winemag-data_first150k.
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
## Parsed with column specification:
## cols(
##   X1 = col_double(),
##   country = col_character(),
##   description = col_character(),
##   designation = col_character(),
##   points = col_double(),
##   price = col_double(),
##   province = col_character(),
##   region_1 = col_character(),
##   region_2 = col_character(),
##   variety = col_character(),
##   winery = col_character()
## )
```

```
head(wine)
```

```
## # A tibble: 6 x 11
##      X1 country description designation points price province region_1
##   <dbl> <chr>   <chr>       <chr>        <dbl> <dbl> <chr>    <chr>
## 1     0 US      This treme~ Martha's V~     96   235 Califor~ Napa Va~
## 2     1 Spain   Ripe aroma~ Carodorum ~     96   110 Norther~ Toro
## 3     2 US      Mac Watson~ Special Se~     96    90 Califor~ Knights~
## 4     3 US      This spent~ Reserve         96    65 Oregon   Willame~
## 5     4 France  This is th~ La Brûlade      95    66 Provence Bandol
## 6     5 Spain   Deep, dens~ Numanthia       95    73 Norther~ Toro
## # ... with 3 more variables: region_2 <chr>, variety <chr>, winery <chr>
```

We can use as_tibble function from tibble package within tidyverse. This will change the class of the wine dataframe to tibble. With data frame being tibble we can further leverage dplyr package within tidyverse.

```
wine <- as_tibble(wine)
head(wine)
```

```
## # A tibble: 6 x 11
##      X1 country description designation points price province region_1
##   <dbl> <chr>   <chr>       <chr>        <dbl> <dbl> <chr>    <chr>
## 1     0 US      This treme~ Martha's V~     96   235 Califor~ Napa Va~
## 2     1 Spain   Ripe aroma~ Carodorum ~     96   110 Norther~ Toro
## 3     2 US      Mac Watson~ Special Se~     96    90 Califor~ Knights~
## 4     3 US      This spent~ Reserve         96    65 Oregon   Willame~
## 5     4 France  This is th~ La Brûlade      95    66 Provence Bandol
## 6     5 Spain   Deep, dens~ Numanthia       95    73 Norther~ Toro
## # ... with 3 more variables: region_2 <chr>, variety <chr>, winery <chr>
```

Data Cleaning and Transformation

When we look at the dataset, we see there are many columns that may not be neccesary for our analysis. For example our business objective is to only look at the wines that are from US. In this case we can group based on selected columns using select function in dplyr package. We can further filter the dataset for the wines that are from US by using filter() function. We can also arrange the dataset to display points by decreasing order by using arrange function. Since we are using multiple functions to the data, we might as well use pipe for code efficiency.

```
#filter, select the needed columns and arrange
wine_df <- wine %>%
  filter(country=="US") %>%
  select(country, province, region_1, variety, points, price) %>%
  arrange(desc(points))

head(wine_df)
```

```
## # A tibble: 6 x 6
##   country province   region_1                variety            points price
##   <chr>   <chr>      <chr>                   <chr>               <dbl> <dbl>
## 1 US      Oregon     Walla Walla Valley (OR) Syrah                 100    65
## 2 US      Oregon     Walla Walla Valley (OR) Syrah                 100    65
## 3 US      California Napa Valley             Cabernet Sauvign~     100   200
## 4 US      California Stags Leap District     Cabernet Sauvign~     100   215
## 5 US      California Russian River Valley    Pinot Noir            100   100
## 6 US      California Rutherford              Cabernet Blend        100   245
```

Since we are looking for the top ranked wines based on their origin, we can group them based on their variety. We can use group_by function in dplyr package.

```r
wine_group<- wine_df %>%
  group_by(variety)

wine_group
```

```
## # A tibble: 62,397 x 6
## # Groups:   variety [218]
##    country province   region_1              variety            points price
##    <chr>   <chr>      <chr>                 <chr>               <dbl> <dbl>
##  1 US      Oregon     Walla Walla Valley (O~ Syrah                100    65
##  2 US      Oregon     Walla Walla Valley (O~ Syrah                100    65
##  3 US      California Napa Valley           Cabernet Sauvign~    100   200
##  4 US      California Stags Leap District   Cabernet Sauvign~    100   215
##  5 US      California Russian River Valley  Pinot Noir           100   100
##  6 US      California Rutherford            Cabernet Blend       100   245
##  7 US      Oregon     Walla Walla Valley (O~ Syrah                100    65
##  8 US      California Russian River Valley  Pinot Noir           100   100
##  9 US      California Napa Valley           Cabernet Sauvign~    100   200
## 10 US      California Rutherford            Cabernet Blend       100   245
## # ... with 62,387 more rows
```

We can look at an overview of our latest dataset by using glimpse function.

```r
glimpse(wine_group)
```

```
## Observations: 62,397
## Variables: 6
## Groups: variety [218]
## $ country  <chr> "US", "US", "US", "US", "US", "US", "US", "US", "US",...
## $ province <chr> "Oregon", "Oregon", "California", "California", "Cali...
## $ region_1 <chr> "Walla Walla Valley (OR)", "Walla Walla Valley (OR)",...
## $ variety  <chr> "Syrah", "Syrah", "Cabernet Sauvignon", "Cabernet Sau...
## $ points   <dbl> 100, 100, 100, 100, 100, 100, 100, 100, 100, 100, 99,...
## $ price    <dbl> 65, 65, 200, 215, 100, 245, 65, 100, 200, 245, 65, 14...
```

We have 62,397 observations, grouped by variety of 218 wines.

We should look for missing values and handle them as needed.

```r
sum(is.na(wine_group))
```

```
## [1] 394
```

```r
sum(is.na(wine_group$country))
```

```
## [1] 0
```

```r
sum(is.na(wine_group$province))
```

```
## [1] 0
```

```
sum(is.na(wine_group$region_1))
```

```
## [1] 136
```

```
sum(is.na(wine_group$variety))
```

```
## [1] 0
```

```
sum(is.na(wine_group$points))
```

```
## [1] 0
```

```
sum(is.na(wine_group$price))
```

```
## [1] 258
```

We have total of 394 missing values. 136 missing values in region_1 column and 258 missing values in price column. COnsidering we have 62,397 observations, we can remove the 394 missing values from our dataset. We can use drop_na function from dplyr function to do this.

```
wine_final <- wine_group %>%
  drop_na()

wine_final
```

```
## # A tibble: 62,003 x 6
## # Groups:   variety [217]
##    country province   region_1               variety           points price
##    <chr>   <chr>      <chr>                  <chr>              <dbl> <dbl>
##  1 US      Oregon     Walla Walla Valley (O~ Syrah                100    65
##  2 US      Oregon     Walla Walla Valley (O~ Syrah                100    65
##  3 US      California Napa Valley            Cabernet Sauvign~    100   200
##  4 US      California Stags Leap District    Cabernet Sauvign~    100   215
##  5 US      California Russian River Valley   Pinot Noir           100   100
##  6 US      California Rutherford             Cabernet Blend       100   245
##  7 US      Oregon     Walla Walla Valley (O~ Syrah                100    65
##  8 US      California Russian River Valley   Pinot Noir           100   100
##  9 US      California Napa Valley            Cabernet Sauvign~    100   200
## 10 US      California Rutherford             Cabernet Blend       100   245
## # ... with 61,993 more rows
```

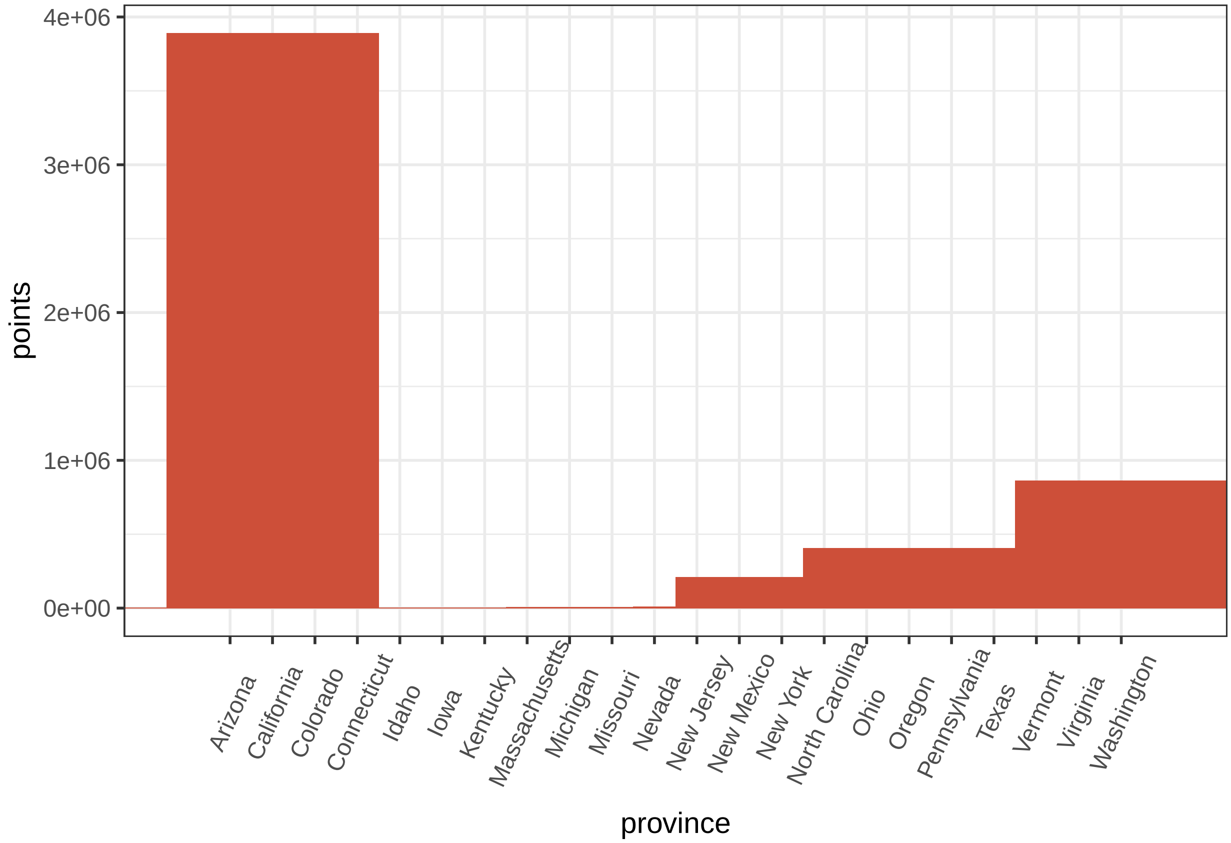
We have loaded and cleaned our data by using tibble, dplyr packages from tidyverse. Our data set is ready for analysis.

Data Exploration and Visualization

We can use ggplot package from tidyverser to visualize the top ranking wine in the US.

```
theme_set(theme_bw())
ggplot(wine_final, aes(province, points))+
  geom_bar(stat="identity", width=5, fill="tomato3")+
  theme(axis.text.x=element_text(angle=65, vjust=0.6))
```

```
## Warning: position_stack requires non-overlapping x intervals
```

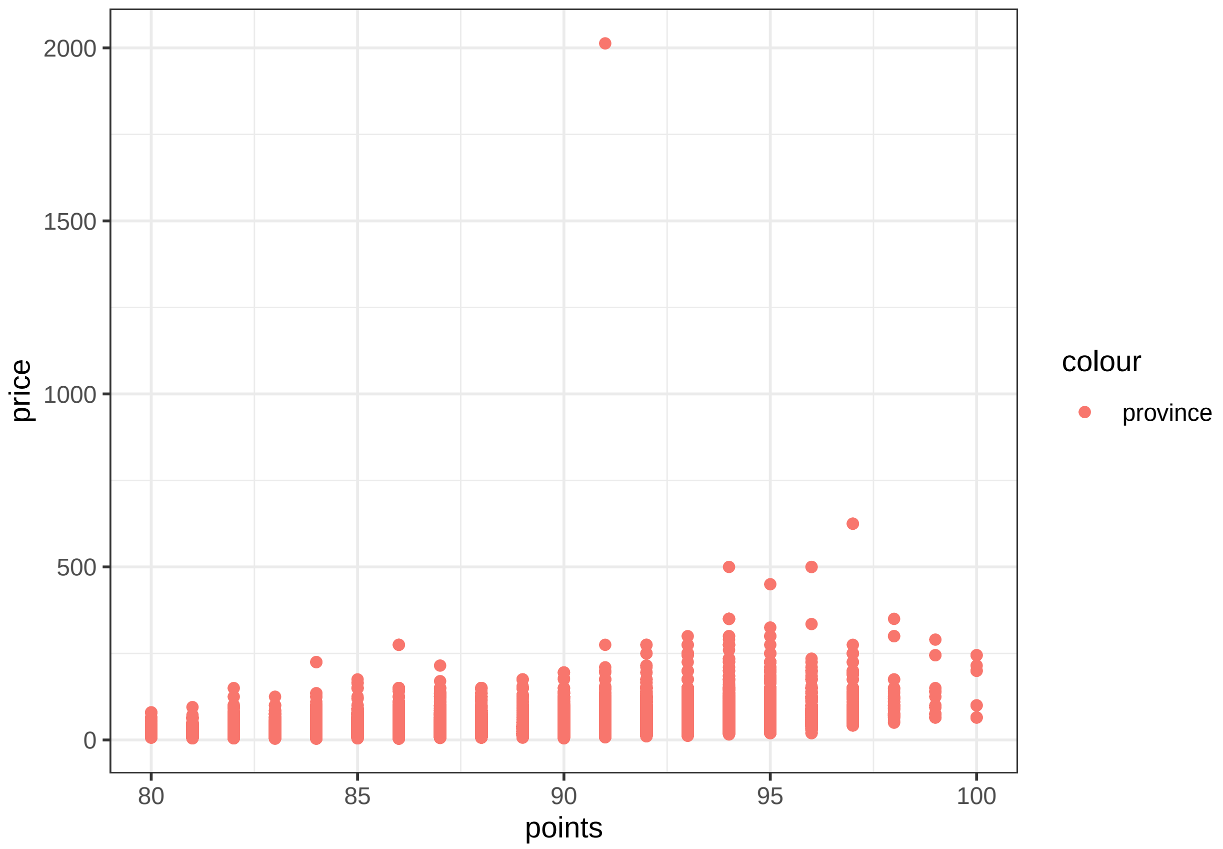


Conclusion

In this assignment, we were able to read the data, clean and transform and visualize by using tidyverse packages, dplyr, tibble and ggplot. Based on our analysis, we were able to find the origin of the top ranking wine within US. We can extend this analysis further and look to see if there are correlations between variables such as points vs price. We can also explore ways to create a simple or multiple linear predictive model.

Add ons

```
ggplot(wine_final, aes(x=points, y=price))+ geom_point(aes(color='province'))
```



There is an outlier that has an extremrly high price 2000.

What is that wine? This value strongly skewed the distribution.

```
filter(wine_final, price>2000)
```

```
## # A tibble: 1 x 6
## # Groups:   variety [1]
##   country province   region_1     variety    points price
##   <chr>   <chr>      <chr>        <chr>       <dbl> <dbl>
## 1 US      California Arroyo Seco Chardonnay     91  2013
```

It look like a mistake. The year 2013 fell in variable price.because it is rate less than 100 but has an extraordinary price.

what is the price of other variety of Chardonnay?

```
 med.price <- wine_final %>% filter(variety %in% 'Chardonnay', province %in% 'California', points==91) %
```

```
wine_final$price[wine_final$price==2013] <- med.price
```

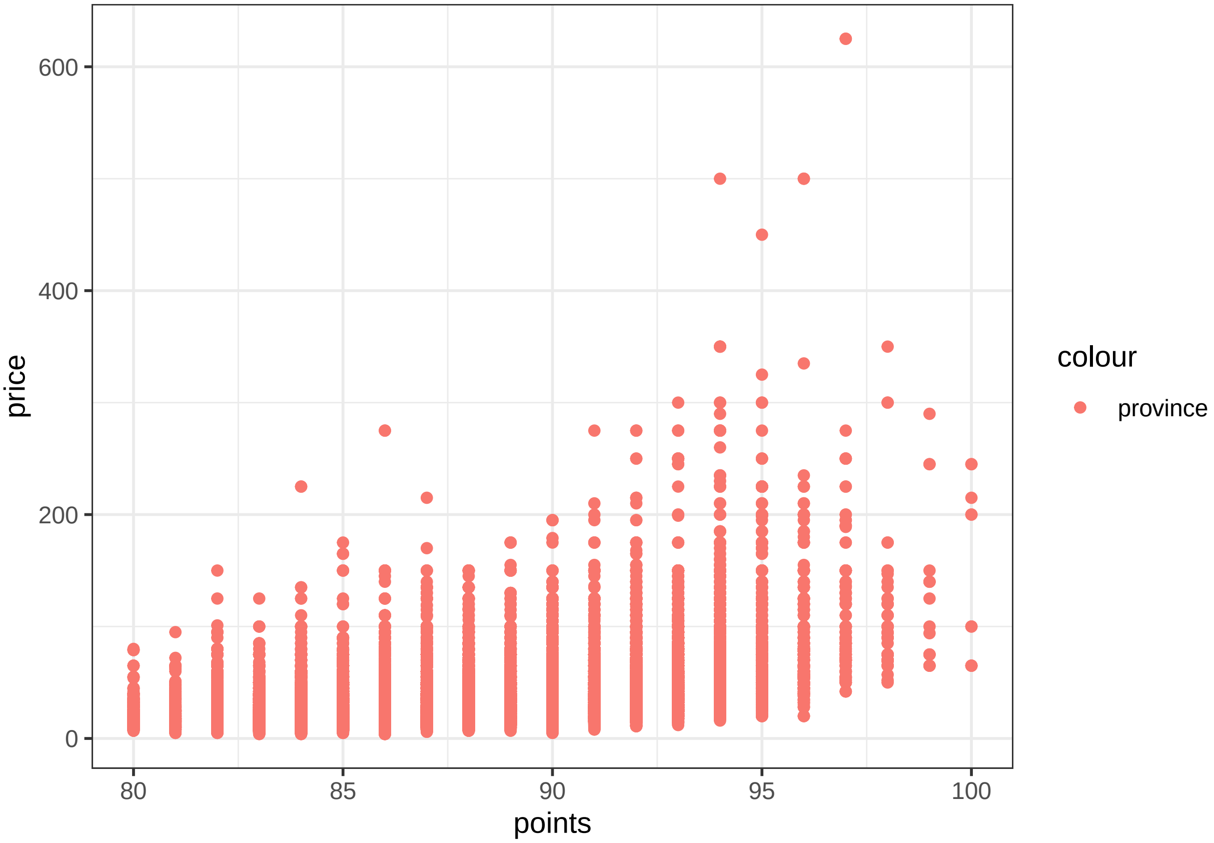
```
## Warning in wine_final$price[wine_final$price == 2013] <- med.price: number
## of items to replace is not a multiple of replacement length
```

```
wine_final$price <- as.numeric(wine_final$price)
```

```
## Warning: NAs introduced by coercion
```

```
ggplot(wine_final, aes(x=points, y=price, color='province'))+ geom_point()
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



The outlier point has been changed giving more realistic data. We used median instead of mean become the number 2013 will still be part of the mean but not the median.