R Project

Junyan Huang

April 25, 2018

# About the Data Set

## Overview

The data set is about different types of wine. It includes the country, province, region, winery the wine is made from, the price of the wine, the point and description that wine enthusiast gives. To fit my research, I will choose some variables and some portion of the observations.

## Source

The original data set can be found in the URL link below: <https://www.kaggle.com/zynicide/wine-reviews/data>

## Clean Data Set

* Retrieve data set from local computer and delete any row with missing value(s).

setwd("D:/STUDY/R/Project")  
getwd()

## [1] "D:/STUDY/R/Project"

wine\_raw <- read.csv("winemag.csv", header = T, na.strings=c("","NA"))

wine\_raw2 <- wine\_raw[complete.cases(wine\_raw), ]  
ncol(wine\_raw2)

## [1] 8

nrow(wine\_raw2)

## [1] 114393

After removing empty cases, the data set consists of 8 columns and 114393 rows.

* Choose the columns for my reaserch. The columns chosen are points, price, region and variey.

wine\_raw3 <- wine\_raw2[, c(3, 4, 6, 7)]  
head(wine\_raw3)

## points price region variety  
## 1 96 235 Napa Valley Cabernet Sauvignon  
## 2 96 110 Toro Tinta de Toro  
## 3 96 90 Knights Valley Sauvignon Blanc  
## 4 96 65 Willamette Valley Pinot Noir  
## 5 95 66 Bandol Provence red blend  
## 6 95 73 Toro Tinta de Toro

* Create a smaller data set that fits my research. I choose regions of Columbia Valley (WA), Mendoza, Russian River Valley, California, and varieties of Chardonnay, Pinot Noir, Cabernet Sauvignon, Malbec, Red Blend and Merlot.

library("dplyr")

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

wine\_raw4<- filter(wine\_raw3, region %in% c("Columbia Valley (WA)", "Mendoza", "Russian River Valley", "California"))  
wine\_raw4<- filter(wine\_raw4, variety %in% c("Chardonnay", "Pinot Noir", "Cabernet Sauvignon", "Malbec", "Red Blend", "Merlot"))  
head(wine\_raw4)

## points price region variety  
## 1 90 37 Russian River Valley Chardonnay  
## 2 90 42 Columbia Valley (WA) Chardonnay  
## 3 90 18 Russian River Valley Chardonnay  
## 4 91 30 Mendoza Malbec  
## 5 91 25 Mendoza Malbec  
## 6 91 65 Russian River Valley Pinot Noir

ncol(wine\_raw4)

## [1] 4

nrow(wine\_raw4)

## [1] 9644

write.csv(wine\_raw4, "wine\_raw4.csv")

wine\_raw5 <- read.csv("wine\_raw5.csv", header = T)  
head(wine\_raw5)

## ï..points price region variety  
## 1 90 37 Russian River Valley Chardonnay  
## 2 90 42 Columbia Valley (WA) Chardonnay  
## 3 90 18 Russian River Valley Chardonnay  
## 4 91 30 Mendoza Malbec  
## 5 91 25 Mendoza Malbec  
## 6 91 65 Russian River Valley Pinot Noir

summary(wine\_raw5)

## ï..points price region   
## Min. : 80.0 Min. : 4.00 California :1857   
## 1st Qu.: 85.0 1st Qu.: 12.00 Columbia Valley (WA):2515   
## Median : 87.0 Median : 20.00 Mendoza :2639   
## Mean : 87.3 Mean : 26.46 Russian River Valley:2633   
## 3rd Qu.: 90.0 3rd Qu.: 35.00   
## Max. :100.0 Max. :190.00   
## variety   
## Cabernet Sauvignon:1759   
## Chardonnay :2204   
## Malbec :1474   
## Merlot :1054   
## Pinot Noir :1888   
## Red Blend :1265

head(wine\_raw5)

## ï..points price region variety  
## 1 90 37 Russian River Valley Chardonnay  
## 2 90 42 Columbia Valley (WA) Chardonnay  
## 3 90 18 Russian River Valley Chardonnay  
## 4 91 30 Mendoza Malbec  
## 5 91 25 Mendoza Malbec  
## 6 91 65 Russian River Valley Pinot Noir

For some reason, after I filtered the variables and rows I need, the removed orginal categories in region still stay as NA. So I saved the cvs file to computer and dealed with it in excel, and then reload it to R.

* Turn the column “region” from categorical to numeric (“Columbia Valley (WA)” = 1, “Mendoza” = 2, “Russian River Valley” = 3, “California”= 4)

region\_no.<- wine\_raw5$region  
region\_no.<- factor(region\_no.)  
region\_no. <- sapply(as.character(region\_no.), switch, "Columbia Valley (WA)" = 1, "Mendoza" = 2, "Russian River Valley" = 3, "California"= 4, USE.NAMES = F)  
#summary(region\_no.)  
#length(region\_no.)  
#region\_no. <- match(region\_no., wine\_raw5$  
wine <- data.frame(wine\_raw5$ï..points, wine\_raw5$price, region\_no., wine\_raw5$variety)  
library(data.table)

##   
## Attaching package: 'data.table'

## The following objects are masked from 'package:dplyr':  
##   
## between, first, last

setnames(wine, old = c("wine\_raw5.ï..points","wine\_raw5.price", "region\_no.", "wine\_raw5.variety" ), new = c("points", "price", "region\_no.", "variety"))  
summary(wine)

## points price region\_no.   
## Min. : 80.0 Min. : 4.00 Min. :1.000   
## 1st Qu.: 85.0 1st Qu.: 12.00 1st Qu.:1.000   
## Median : 87.0 Median : 20.00 Median :2.000   
## Mean : 87.3 Mean : 26.46 Mean :2.397   
## 3rd Qu.: 90.0 3rd Qu.: 35.00 3rd Qu.:3.000   
## Max. :100.0 Max. :190.00 Max. :4.000   
## variety   
## Cabernet Sauvignon:1759   
## Chardonnay :2204   
## Malbec :1474   
## Merlot :1054   
## Pinot Noir :1888   
## Red Blend :1265

head(wine)

## points price region\_no. variety  
## 1 90 37 3 Chardonnay  
## 2 90 42 1 Chardonnay  
## 3 90 18 3 Chardonnay  
## 4 91 30 2 Malbec  
## 5 91 25 2 Malbec  
## 6 91 65 3 Pinot Noir

Therefore, the data set that I will use consists 4 columns and 9644 rows.

## Metadata

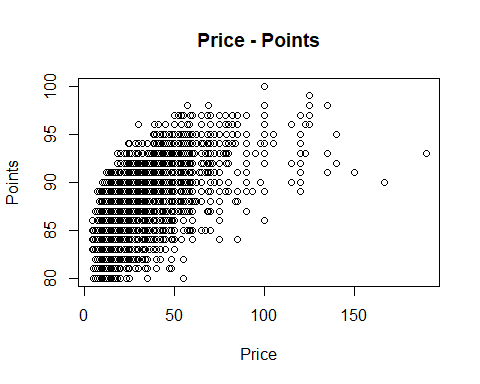
* points: The number of points Wine Enthusiast rated the wine on a scale of 1-100 (numeric)
* price: The cost for a bottle of the wine (numeric)
* region: The wine growing area in a province or state (ie Napa) (string) (“Columbia Valley (WA)” = 1, “Mendoza” = 2, “Russian River Valley” = 3, “California”= 4)
* variety: The type of grapes used to make the wine (ie Pinot Noir) (string)

# Apply Techniques

## 1. Cluster Analysis

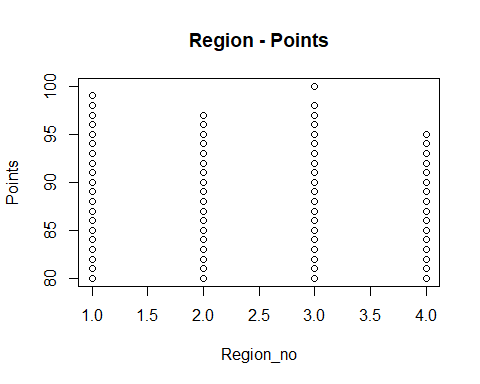
* Research Question: Explore how well the variables points, price and region explain the variable variety. For example, if a certain range of point, a certain range of price and a certain region will correctly explain a certain variety.
* Create scatter plots and see the relationship between variables.

plot(wine$price, wine$points, xlab = 'Price', ylab = 'Points', main = "Price - Points")



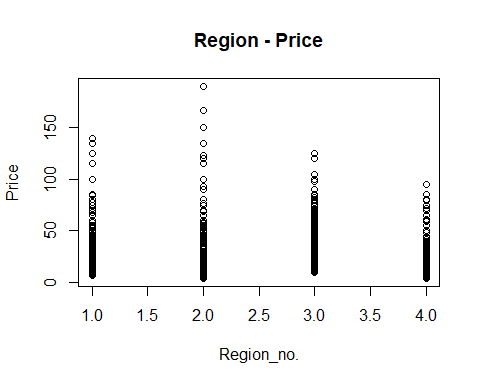
From the plot, we can the relation ship between price and point is very week. There is a weak trend that the higher the price, the higher the points will be.

plot(wine$region\_no., wine$points, xlab = "Region\_no", ylab = "Points", main = "Region - Points" )



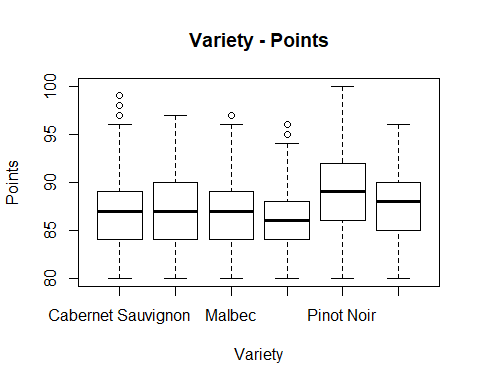
From the plot, we can see the relationship between points and region is not strong. The points vary from 80 to 100 among the 4 regions. And some higher points (above 95) are in region 1 and region 3.

plot(wine$region\_no., wine$price, xlab = "Region\_no.", ylab = "Price", main = "Region - Price")



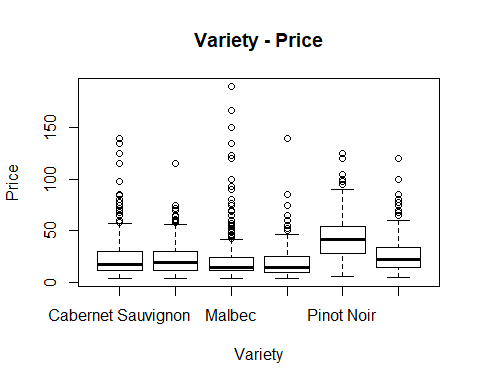
From the plot, we can see the relationship between price and region is not strong either. Price is mainly within $50 for all 4 regions. And region 2 and region 1 have some higher prices.

plot(wine$variety, wine$points, xlab = "Variety", ylab = "Points", main = "Variety - Points")



From the plot, we can see there’s no strong relationship between variety and points. The points of Pinot Noir is slightly higher.

plot(wine$variety, wine$price, xlab = "Variety", ylab = "Price", main = "Variety - Price")



From the plot, we can see the relationship between variety and price is weak. Price is mainly winthin $50. The price of Pinot Noir is slight hiher. And we may also find some higher price in Malbec.

* Create a kmeans() object.

wine\_numeric <- wine[, c(1,2,3)]  
wine\_numeric\_k3 <- kmeans(wine\_numeric, center = 3, nstart = 20)  
#wine\_numeric\_k3  
wine\_numeric\_k3$size

## [1] 2940 5872 832

wine\_numeric\_k3$centers

## points price region\_no.  
## 1 89.34728 37.48912 2.215306  
## 2 85.61069 14.84911 2.461512  
## 3 91.93029 69.38942 2.587740

The three clusters can be labeled by the price.

Cluster 2: most expensive

Cluster 3: medium expensive

Cluster 1: least expensive

* Appy the clasters to the variety variable.

wine\_play <- wine  
wine\_play$clstuer\_id <- wine\_numeric\_k3$cluster  
table(wine\_play$variety, wine\_play$clstuer\_id)

##   
## 1 2 3  
## Cabernet Sauvignon 464 1206 89  
## Chardonnay 670 1461 73  
## Malbec 241 1156 77  
## Merlot 189 836 29  
## Pinot Noir 960 442 486  
## Red Blend 416 771 78

From the table above, we can see the varieties are spread among the three clusters. Cabernet Sauvignon, Chardonnay, Malbec, Merlot and Red Blend are mainly in cluster 2, the most expensive. Pinot Noir is mainlt in cluster 1, the lease expensive. Cluster 2 has the least number for all five varieties.

Therefore, to answer the research question, it seems that variables points, price and region together cannot well explain the variety of wine.

(In order to get more meaningful results, I experimented with different number of clusters, but the results are the similar - the six varieties are spread out among the clusters. It may indicate that these three variables - points, price and region may not very good predictors for the variety.)

## 2. Neutral Networks

* Research Question: Build a predictive model to predict the variety of wine from its points, price, and production region.
* Create a training and a testing set.

library("caret")

## Loading required package: lattice

## Loading required package: ggplot2

library("nnet")  
set.seed(838239)  
wine\_samples <- createDataPartition(wine$variety, p = .10, list = FALSE)  
wine\_train <- wine[wine\_samples, ]  
wine\_test <- wine[-wine\_samples, ]

* Create a neural nework.

wine\_nn <- nnet(variety ~., data = wine\_train, size =40, maxit = 10000)

## # weights: 406  
## initial value 2330.420533   
## iter 10 value 1604.153007  
## iter 20 value 1545.114129  
## iter 30 value 1473.645240  
## iter 40 value 1407.796115  
## iter 50 value 1363.951066  
## iter 60 value 1331.756427  
## iter 70 value 1260.780106  
## iter 80 value 1235.804466  
## iter 90 value 1228.342901  
## iter 100 value 1221.577549  
## iter 110 value 1221.086400  
## iter 120 value 1219.794553  
## iter 130 value 1205.195274  
## iter 140 value 1192.167520  
## iter 150 value 1185.304702  
## iter 160 value 1178.226020  
## iter 170 value 1173.661371  
## iter 180 value 1167.235400  
## iter 190 value 1155.018832  
## iter 200 value 1149.267309  
## iter 210 value 1145.717539  
## iter 220 value 1143.582299  
## iter 230 value 1143.375973  
## iter 240 value 1142.743057  
## iter 250 value 1140.500461  
## iter 260 value 1138.261073  
## iter 270 value 1136.557825  
## iter 280 value 1133.454408  
## iter 290 value 1131.887898  
## iter 300 value 1131.437151  
## iter 310 value 1131.104490  
## iter 320 value 1130.975316  
## iter 330 value 1130.892575  
## iter 340 value 1130.747731  
## iter 350 value 1130.576814  
## iter 360 value 1130.362901  
## iter 370 value 1130.118396  
## iter 380 value 1129.968045  
## iter 390 value 1129.881312  
## iter 400 value 1129.760121  
## iter 410 value 1129.628186  
## iter 420 value 1129.496050  
## iter 430 value 1129.383815  
## iter 440 value 1129.184512  
## iter 450 value 1129.072280  
## iter 460 value 1128.963206  
## iter 470 value 1128.884045  
## iter 480 value 1128.852331  
## iter 490 value 1128.816037  
## iter 500 value 1128.763085  
## iter 510 value 1128.732158  
## iter 520 value 1128.705508  
## iter 530 value 1128.675994  
## iter 540 value 1128.649133  
## iter 550 value 1128.609364  
## iter 560 value 1128.593487  
## iter 570 value 1128.537674  
## iter 580 value 1128.531514  
## iter 590 value 1128.401144  
## iter 600 value 1128.291183  
## iter 610 value 1128.276572  
## iter 620 value 1128.248255  
## iter 630 value 1128.225339  
## iter 640 value 1128.214954  
## iter 650 value 1128.177807  
## iter 660 value 1128.164159  
## iter 670 value 1128.137183  
## iter 680 value 1128.124887  
## iter 690 value 1128.062984  
## iter 700 value 1128.046977  
## iter 710 value 1128.040527  
## iter 720 value 1128.032982  
## iter 730 value 1127.925437  
## iter 740 value 1127.890310  
## iter 750 value 1127.884267  
## iter 760 value 1127.786327  
## iter 770 value 1127.764880  
## iter 780 value 1127.758689  
## iter 790 value 1127.754445  
## iter 800 value 1127.634069  
## iter 810 value 1127.565578  
## iter 820 value 1127.563038  
## iter 830 value 1127.560714  
## iter 840 value 1127.549533  
## iter 850 value 1127.545146  
## iter 860 value 1127.542245  
## iter 870 value 1127.541421  
## iter 880 value 1127.508560  
## iter 890 value 1127.505695  
## iter 900 value 1127.504584  
## final value 1127.504465   
## converged

After experimenting with a lot of numbers of size, I found size 40 has the best result.

summary(wine\_nn)

## a 3-40-6 network with 406 weights  
## options were - softmax modelling   
## b->h1 i1->h1 i2->h1 i3->h1   
## 0.64 0.42 0.63 0.43   
## b->h2 i1->h2 i2->h2 i3->h2   
## -0.10 -0.53 -0.09 0.46   
## b->h3 i1->h3 i2->h3 i3->h3   
## 0.70 -0.34 -0.01 -0.03   
## b->h4 i1->h4 i2->h4 i3->h4   
## -0.06 -0.25 -0.10 0.10   
## b->h5 i1->h5 i2->h5 i3->h5   
## -0.48 0.41 0.11 -0.06   
## b->h6 i1->h6 i2->h6 i3->h6   
## 14.99 -80.21 243.48 161.96   
## b->h7 i1->h7 i2->h7 i3->h7   
## -0.05 -6.45 -7.22 -0.75   
## b->h8 i1->h8 i2->h8 i3->h8   
## 0.02 -0.17 -0.28 -0.64   
## b->h9 i1->h9 i2->h9 i3->h9   
## -514.91 343.32 -948.11 -7.19   
## b->h10 i1->h10 i2->h10 i3->h10   
## 0.67 0.50 -0.13 0.19   
## b->h11 i1->h11 i2->h11 i3->h11   
## 0.40 0.26 0.24 0.50   
## b->h12 i1->h12 i2->h12 i3->h12   
## -0.01 -0.84 -0.39 -0.40   
## b->h13 i1->h13 i2->h13 i3->h13   
## 0.27 0.42 0.63 0.30   
## b->h14 i1->h14 i2->h14 i3->h14   
## 0.21 0.67 0.18 -0.37   
## b->h15 i1->h15 i2->h15 i3->h15   
## 12.65 0.00 0.00 -16.11   
## b->h16 i1->h16 i2->h16 i3->h16   
## 246.85 53.15 -73.29 418.35   
## b->h17 i1->h17 i2->h17 i3->h17   
## 0.48 0.25 0.43 0.20   
## b->h18 i1->h18 i2->h18 i3->h18   
## -0.68 -0.32 -0.19 0.68   
## b->h19 i1->h19 i2->h19 i3->h19   
## 0.13 0.22 0.16 -0.50   
## b->h20 i1->h20 i2->h20 i3->h20   
## -4.08 -0.05 -0.02 4.19   
## b->h21 i1->h21 i2->h21 i3->h21   
## -34.99 2.08 -1.97 0.75   
## b->h22 i1->h22 i2->h22 i3->h22   
## -0.38 -0.65 -0.06 0.07   
## b->h23 i1->h23 i2->h23 i3->h23   
## -0.02 -0.54 0.22 0.65   
## b->h24 i1->h24 i2->h24 i3->h24   
## 0.12 0.28 0.60 -0.33   
## b->h25 i1->h25 i2->h25 i3->h25   
## -54.18 -0.09 -0.03 30.71   
## b->h26 i1->h26 i2->h26 i3->h26   
## 0.48 -0.55 -0.09 -0.18   
## b->h27 i1->h27 i2->h27 i3->h27   
## -0.37 -0.26 -0.17 0.61   
## b->h28 i1->h28 i2->h28 i3->h28   
## 0.12 -0.42 0.09 -0.54   
## b->h29 i1->h29 i2->h29 i3->h29   
## -0.32 0.68 -0.38 0.44   
## b->h30 i1->h30 i2->h30 i3->h30   
## 0.67 -0.61 -0.45 0.04   
## b->h31 i1->h31 i2->h31 i3->h31   
## 0.08 -0.73 -0.66 -0.51   
## b->h32 i1->h32 i2->h32 i3->h32   
## -0.51 0.65 0.57 0.12   
## b->h33 i1->h33 i2->h33 i3->h33   
## 0.64 0.31 0.38 0.16   
## b->h34 i1->h34 i2->h34 i3->h34   
## -74.18 0.76 1.15 1.90   
## b->h35 i1->h35 i2->h35 i3->h35   
## 0.01 0.29 0.06 -0.36   
## b->h36 i1->h36 i2->h36 i3->h36   
## -0.60 -0.70 -0.37 -0.25   
## b->h37 i1->h37 i2->h37 i3->h37   
## 0.65 0.48 -0.13 0.56   
## b->h38 i1->h38 i2->h38 i3->h38   
## -0.68 -0.45 0.08 -0.03   
## b->h39 i1->h39 i2->h39 i3->h39   
## 0.03 -0.53 -0.17 -0.26   
## b->h40 i1->h40 i2->h40 i3->h40   
## 0.16 0.25 0.28 -0.69   
## b->o1 h1->o1 h2->o1 h3->o1 h4->o1 h5->o1 h6->o1 h7->o1 h8->o1   
## 38.20 37.65 -0.55 0.30 0.44 37.13 0.52 -0.74 0.13   
## h9->o1 h10->o1 h11->o1 h12->o1 h13->o1 h14->o1 h15->o1 h16->o1 h17->o1   
## -0.10 37.33 38.16 0.61 37.91 38.35 19.07 -467.90 38.28   
## h18->o1 h19->o1 h20->o1 h21->o1 h22->o1 h23->o1 h24->o1 h25->o1 h26->o1   
## 0.28 38.46 17.32 314.47 -0.35 -0.35 38.13 -16.61 0.70   
## h27->o1 h28->o1 h29->o1 h30->o1 h31->o1 h32->o1 h33->o1 h34->o1 h35->o1   
## 0.36 -0.03 37.58 0.34 -0.44 38.27 38.45 -452.85 38.13   
## h36->o1 h37->o1 h38->o1 h39->o1 h40->o1   
## -0.45 37.34 -0.20 0.44 37.37   
## b->o2 h1->o2 h2->o2 h3->o2 h4->o2 h5->o2 h6->o2 h7->o2 h8->o2   
## 2.15 2.17 0.38 0.66 0.07 2.72 -0.20 0.23 0.09   
## h9->o2 h10->o2 h11->o2 h12->o2 h13->o2 h14->o2 h15->o2 h16->o2 h17->o2   
## 0.81 2.41 2.65 -0.02 2.36 1.73 -176.56 96.55 2.45   
## h18->o2 h19->o2 h20->o2 h21->o2 h22->o2 h23->o2 h24->o2 h25->o2 h26->o2   
## -0.45 2.57 -63.48 317.21 0.10 0.03 2.85 59.89 -0.31   
## h27->o2 h28->o2 h29->o2 h30->o2 h31->o2 h32->o2 h33->o2 h34->o2 h35->o2   
## 0.15 -0.42 2.75 0.34 0.10 2.08 2.35 -447.22 2.00   
## h36->o2 h37->o2 h38->o2 h39->o2 h40->o2   
## -0.21 2.47 0.20 -0.03 2.24   
## b->o3 h1->o3 h2->o3 h3->o3 h4->o3 h5->o3 h6->o3 h7->o3 h8->o3   
## 38.50 37.35 -0.26 0.60 -0.68 38.56 1.04 1.42 -0.01   
## h9->o3 h10->o3 h11->o3 h12->o3 h13->o3 h14->o3 h15->o3 h16->o3 h17->o3   
## 1.11 38.18 37.57 0.60 38.33 38.70 -330.69 11.14 37.56   
## h18->o3 h19->o3 h20->o3 h21->o3 h22->o3 h23->o3 h24->o3 h25->o3 h26->o3   
## -0.25 37.56 -13.27 -163.88 -0.63 0.28 37.87 4.61 -0.44   
## h27->o3 h28->o3 h29->o3 h30->o3 h31->o3 h32->o3 h33->o3 h34->o3 h35->o3   
## -0.39 0.03 37.46 -0.08 0.50 38.66 38.63 -451.73 37.51   
## h36->o3 h37->o3 h38->o3 h39->o3 h40->o3   
## -0.02 38.72 -0.65 -0.43 38.10   
## b->o4 h1->o4 h2->o4 h3->o4 h4->o4 h5->o4 h6->o4 h7->o4 h8->o4   
## 34.50 35.57 -0.28 0.20 -0.67 34.68 -0.67 -0.54 0.05   
## h9->o4 h10->o4 h11->o4 h12->o4 h13->o4 h14->o4 h15->o4 h16->o4 h17->o4   
## -0.04 34.86 34.82 0.50 34.83 34.61 89.10 -475.26 34.69   
## h18->o4 h19->o4 h20->o4 h21->o4 h22->o4 h23->o4 h24->o4 h25->o4 h26->o4   
## 0.10 35.50 0.55 366.62 -0.57 0.70 35.48 1.74 -0.65   
## h27->o4 h28->o4 h29->o4 h30->o4 h31->o4 h32->o4 h33->o4 h34->o4 h35->o4   
## -0.69 -0.39 35.48 -0.05 0.30 35.66 35.46 -454.06 35.17   
## h36->o4 h37->o4 h38->o4 h39->o4 h40->o4   
## -0.15 35.22 -0.53 -0.63 35.18   
## b->o5 h1->o5 h2->o5 h3->o5 h4->o5 h5->o5 h6->o5 h7->o5 h8->o5   
## -82.93 -82.64 0.00 -0.05 0.61 -82.82 -0.15 -1.03 0.01   
## h9->o5 h10->o5 h11->o5 h12->o5 h13->o5 h14->o5 h15->o5 h16->o5 h17->o5   
## -0.35 -82.72 -82.06 -0.69 -82.96 -82.43 -229.70 266.30 -82.64   
## h18->o5 h19->o5 h20->o5 h21->o5 h22->o5 h23->o5 h24->o5 h25->o5 h26->o5   
## 0.26 -82.40 -72.63 -406.90 -0.39 0.43 -82.56 70.01 -0.31   
## h27->o5 h28->o5 h29->o5 h30->o5 h31->o5 h32->o5 h33->o5 h34->o5 h35->o5   
## -0.38 0.34 -82.80 0.56 -0.08 -82.08 -82.05 1464.88 -81.83   
## h36->o5 h37->o5 h38->o5 h39->o5 h40->o5   
## -0.29 -82.55 0.36 0.65 -82.84   
## b->o6 h1->o6 h2->o6 h3->o6 h4->o6 h5->o6 h6->o6 h7->o6 h8->o6   
## -31.11 -31.24 -0.15 0.36 0.26 -30.91 0.13 -0.04 -0.21   
## h9->o6 h10->o6 h11->o6 h12->o6 h13->o6 h14->o6 h15->o6 h16->o6 h17->o6   
## -0.86 -31.53 -31.37 -0.30 -30.60 -30.30 628.62 568.86 -30.55   
## h18->o6 h19->o6 h20->o6 h21->o6 h22->o6 h23->o6 h24->o6 h25->o6 h26->o6   
## -0.13 -31.02 132.95 -426.17 -0.37 -0.58 -31.18 -117.39 -0.65   
## h27->o6 h28->o6 h29->o6 h30->o6 h31->o6 h32->o6 h33->o6 h34->o6 h35->o6   
## 0.06 0.58 -31.24 0.11 -0.07 -31.52 -30.49 340.64 -30.92   
## h36->o6 h37->o6 h38->o6 h39->o6 h40->o6   
## 0.27 -30.96 0.50 -0.11 -31.51

In the summary result, the model has three nodes in the first level, 40 nodes in the second level to predict the six types of varieties.

* Create an extra column in both the train and the test data sets and populate the new column using the predict function. Then review the predicted value.

wine\_train$predict <- predict(wine\_nn, wine\_train, type = "class")  
wine\_test$predict <- predict(wine\_nn, wine\_test, type = "class")  
head(wine\_train[,4:5], 10)

## variety predict  
## 6 Pinot Noir Pinot Noir  
## 38 Cabernet Sauvignon Chardonnay  
## 45 Chardonnay Pinot Noir  
## 56 Chardonnay Pinot Noir  
## 57 Cabernet Sauvignon Merlot  
## 62 Cabernet Sauvignon Red Blend  
## 88 Cabernet Sauvignon Cabernet Sauvignon  
## 90 Chardonnay Pinot Noir  
## 116 Cabernet Sauvignon Malbec  
## 117 Red Blend Cabernet Sauvignon

In the result, the first colum is the actual variety and the second column is the predicted variety. The predicted results seem to be not very good with pretty low accuracy.

* Evaluate the trained neural network in table.

cm <- table(wine$variety[-wine\_samples], wine\_test$predict)  
cm

##   
## Cabernet Sauvignon Chardonnay Malbec Merlot  
## Cabernet Sauvignon 220 501 448 82  
## Chardonnay 31 964 267 111  
## Malbec 20 78 1164 6  
## Merlot 89 461 90 104  
## Pinot Noir 14 450 41 6  
## Red Blend 121 345 182 68  
##   
## Pinot Noir Red Blend  
## Cabernet Sauvignon 7 325  
## Chardonnay 416 194  
## Malbec 2 56  
## Merlot 3 201  
## Pinot Noir 1160 28  
## Red Blend 7 415

From the table, we can see the prediction result is not good. The varieties are mis-predicted as other varieties. Among them, the highest accuracy rate may be Malbec and Pinot Noir.

* Calculate the accuracy rate

accuracy <- sum(cm[1],cm[8], cm[15], cm[22], cm[29], cm[36])/sum(cm[1:36])  
accuracy

## [1] 0.4641005

So the best model I can make is size 40 with a accuracy rate of 46.41%.

* Create a random row of data and see how well the model works.

newdata <- data.frame(90, 50, 2, NA)  
names(newdata) <- c("points", "price", "region\_no.", "variety")  
pred\_variety <- predict(wine\_nn, newdata, type = "class")  
print(pred\_variety)

## [1] "Malbec"

Since the predictive model only has 46.41% accuracy rate. So I can only be 46.41% confident that with points of 90 and price of $50, and produced in Mendoza (region no.2), the wine variety is Malbec.

## 3. Association Analysis

* Research Question: Since the three variables points, price andd region together cannot predict the wine variety very well as was experimented above, is it possible that one or more of these variables can do a better job? So in this part, I will use association analysis to find rules in the data set.
* Turn the numeric columns into characters.

wine\_3 <- wine\_raw5  
points\_char <- wine\_3$ï..points  
points\_char <- as.character(points\_char)  
points\_char[1:20]

## [1] "90" "90" "90" "91" "91" "91" "86" "86" "86" "91" "87" "87" "87" "87"  
## [15] "94" "94" "91" "91" "90" "90"

price\_char <- wine\_3$price  
price\_char <- as.character(price\_char)  
price\_char[1:20]

## [1] "37" "42" "18" "30" "25" "65" "10" "18" "10" "59" "10" "18" "11" "36"  
## [15] "60" "50" "55" "40" "15" "30"

wine\_3 <- data.frame(points\_char, price\_char, wine\_3$region, wine\_3$variety)  
setnames(wine\_3, old = c("points\_char", "price\_char", "wine\_3.region", "wine\_3.variety" ), new = c("points\_char", "price\_char", "region", "variety"))  
head(wine\_3)

## points\_char price\_char region variety  
## 1 90 37 Russian River Valley Chardonnay  
## 2 90 42 Columbia Valley (WA) Chardonnay  
## 3 90 18 Russian River Valley Chardonnay  
## 4 91 30 Mendoza Malbec  
## 5 91 25 Mendoza Malbec  
## 6 91 65 Russian River Valley Pinot Noir

summary(wine\_3)

## points\_char price\_char region   
## 87 :1157 10 : 753 California :1857   
## 88 : 938 12 : 552 Columbia Valley (WA):2515   
## 86 : 926 15 : 494 Mendoza :2639   
## 85 : 908 20 : 473 Russian River Valley:2633   
## 84 : 907 11 : 416   
## 90 : 808 13 : 393   
## (Other):4000 (Other):6563   
## variety   
## Cabernet Sauvignon:1759   
## Chardonnay :2204   
## Malbec :1474   
## Merlot :1054   
## Pinot Noir :1888   
## Red Blend :1265   
##

* Generate the rules for outcome of “variety=Malbec”.

library(arules)

## Loading required package: Matrix

##   
## Attaching package: 'arules'

## The following object is masked from 'package:dplyr':  
##   
## recode

## The following objects are masked from 'package:base':  
##   
## abbreviate, write

wine\_3\_rules1 <- apriori(wine\_3, parameter = list(supp = .001, conf = .70), appearance = list(default="lhs", rhs="variety=Malbec"), control=list(verbose=F))  
   
wine\_3\_rules1 <- sort(wine\_3\_rules1, by = "confidence", decreasing = TRUE)  
inspect(wine\_3\_rules1[1:10])

## lhs rhs support confidence lift count  
## [1] {points\_char=90,   
## price\_char=24,   
## region=Mendoza} => {variety=Malbec} 0.001140606 1.0000000 6.542741 11  
## [2] {price\_char=55,   
## region=Mendoza} => {variety=Malbec} 0.001140606 0.8461538 5.536165 11  
## [3] {points\_char=93,   
## region=Mendoza} => {variety=Malbec} 0.002488594 0.8000000 5.234193 24  
## [4] {points\_char=89,   
## price\_char=18,   
## region=Mendoza} => {variety=Malbec} 0.001140606 0.7857143 5.140725 11  
## [5] {price\_char=120,   
## region=Mendoza} => {variety=Malbec} 0.001866445 0.7826087 5.120406 18  
## [6] {points\_char=87,   
## price\_char=14,   
## region=Mendoza} => {variety=Malbec} 0.001866445 0.7826087 5.120406 18  
## [7] {points\_char=90,   
## price\_char=20,   
## region=Mendoza} => {variety=Malbec} 0.001451680 0.7777778 5.088798 14  
## [8] {points\_char=91,   
## region=Mendoza} => {variety=Malbec} 0.007776856 0.7352941 4.810839 75  
## [9] {points\_char=92,   
## price\_char=50,   
## region=Mendoza} => {variety=Malbec} 0.001347988 0.7222222 4.725313 13  
## [10] {points\_char=91,   
## price\_char=25,   
## region=Mendoza} => {variety=Malbec} 0.001347988 0.7222222 4.725313 13

From the result, we can get rules like these:

1. For wine with points of 90, price of $24 and produced in Mendoza, we can be 100% confident that the wine variety is Malbec.
2. For wine with price of $55 and produced in Mendoza, we can be 84.6% confident that the wine variety is Malbec.
3. For wine with points of $93 and produced in Mendoza, we can be 80% confident that the wine variety is Malbec.
4. For wine with price of $55 and produced in Mendoza, we can be 84.6% confident that the wine variety is Malbec.

…

* Generate the rules for outcome of “variety=Red Blend”.

wine\_3\_rules2 <- apriori(wine\_3, parameter = list(supp = .001, conf = .5), appearance = list(default="lhs", rhs="variety=Red Blend"), control=list(verbose=F))  
wine\_3\_rules2 <- sort(wine\_3\_rules2, by = "confidence", decreasing = TRUE)  
inspect(wine\_3\_rules2[1:10])

## lhs rhs support confidence lift count  
## [1] {points\_char=89,   
## price\_char=25,   
## region=Mendoza} => {variety=Red Blend} 0.001036914 1.0000000 7.623715 10  
## [2] {price\_char=80,   
## region=Mendoza} => {variety=Red Blend} 0.001244297 0.9230769 7.037276 12  
## [3] {points\_char=93,   
## price\_char=35,   
## region=Columbia Valley (WA)} => {variety=Red Blend} 0.001451680 0.7000000 5.336601 14  
## [4] {price\_char=50,   
## region=Columbia Valley (WA)} => {variety=Red Blend} 0.002384903 0.6571429 5.009870 23  
## [5] {price\_char=39,   
## region=Columbia Valley (WA)} => {variety=Red Blend} 0.001140606 0.6470588 4.932992 11  
## [6] {points\_char=88,   
## price\_char=15,   
## region=Columbia Valley (WA)} => {variety=Red Blend} 0.001244297 0.6000000 4.574229 12  
## [7] {points\_char=88,   
## price\_char=20,   
## region=Columbia Valley (WA)} => {variety=Red Blend} 0.002592285 0.5952381 4.537926 25  
## [8] {points\_char=88,   
## price\_char=24,   
## region=Columbia Valley (WA)} => {variety=Red Blend} 0.001036914 0.5882353 4.484538 10  
## [9] {price\_char=33,   
## region=Columbia Valley (WA)} => {variety=Red Blend} 0.001140606 0.5789474 4.413730 11  
## [10] {points\_char=87,   
## price\_char=20,   
## region=Columbia Valley (WA)} => {variety=Red Blend} 0.001659063 0.5714286 4.356409 16

From the result, we can get rules like these:

1. For wine with points of 89, price of $25 and produced in Mendoza, we can be 100% confident that the wine variety is Red Blend.
2. For wine with price of $80 and produced in Mendoza, we can be 92.3% confident that the wine variety is Red Blend.
3. For wine with points of 93, price of $32 and produced in Columbia Valley (WA), we can be 70% confident that the wine variety is Red Blend.
4. For wine with points of 50 and produced in Columbia Valley (WA), we can be 65.7% confident that the wine variety is Red Blend.

…

* Generate the rules for outcome of “variety=Pinot Noir”.

wine\_3\_rules3 <- apriori(wine\_3, parameter = list(supp = .001, conf = .8), appearance = list(default="lhs", rhs="variety=Pinot Noir"), control=list(verbose=F))  
wine\_3\_rules3 <- sort(wine\_3\_rules3, by = "confidence", decreasing = TRUE)  
inspect(wine\_3\_rules3[1:10])

## lhs rhs support confidence lift count  
## [1] {price\_char=90,   
## region=Russian River Valley} => {variety=Pinot Noir} 0.001036914 1 5.108051 10  
## [2] {price\_char=100,   
## region=Russian River Valley} => {variety=Pinot Noir} 0.001451680 1 5.108051 14  
## [3] {price\_char=85,   
## region=Russian River Valley} => {variety=Pinot Noir} 0.001244297 1 5.108051 12  
## [4] {price\_char=57,   
## region=Russian River Valley} => {variety=Pinot Noir} 0.002488594 1 5.108051 24  
## [5] {points\_char=92,   
## price\_char=56} => {variety=Pinot Noir} 0.001140606 1 5.108051 11  
## [6] {points\_char=93,   
## price\_char=54} => {variety=Pinot Noir} 0.001451680 1 5.108051 14  
## [7] {points\_char=92,   
## price\_char=54} => {variety=Pinot Noir} 0.001762754 1 5.108051 17  
## [8] {price\_char=54,   
## region=Russian River Valley} => {variety=Pinot Noir} 0.007154708 1 5.108051 69  
## [9] {points\_char=93,   
## price\_char=48} => {variety=Pinot Noir} 0.001140606 1 5.108051 11  
## [10] {points\_char=92,   
## price\_char=56,   
## region=Russian River Valley} => {variety=Pinot Noir} 0.001140606 1 5.108051 11

From the result, we can get rules like these:

1. For wine with points of 92, price of $56, we can be 100% confident that the wine variety is Pinot Noir.
2. For wine with points of 93, price of $54, we can be 100% confident that the wine variety is Pinot Noir.
3. For wine with price of $54 and produced in Russian River Valley, we can be 100% confident that the wine variety is Pinot Noir.
4. For wine with points of 93, price of $48, we can be 100% confident that the wine variety is Pinot Noir.

* Generate the rules for outcome of “variety=Cabernet Sauvignon”.

wine\_3\_rules4 <- apriori(wine\_3, parameter = list(supp = .001, conf = .5), appearance = list(default="lhs", rhs="variety=Cabernet Sauvignon"), control=list(verbose=F))  
wine\_3\_rules4 <- sort(wine\_3\_rules4, by = "confidence", decreasing = TRUE)  
inspect(wine\_3\_rules4[1:10])

## lhs rhs support confidence lift count  
## [1] {points\_char=90,   
## price\_char=40,   
## region=Columbia Valley (WA)} => {variety=Cabernet Sauvignon} 0.001036914 0.9090909 4.984237 10  
## [2] {points\_char=92,   
## price\_char=42,   
## region=Columbia Valley (WA)} => {variety=Cabernet Sauvignon} 0.001347988 0.7647059 4.192623 13  
## [3] {points\_char=96,   
## region=Columbia Valley (WA)} => {variety=Cabernet Sauvignon} 0.001451680 0.7368421 4.039855 14  
## [4] {price\_char=48,   
## region=Columbia Valley (WA)} => {variety=Cabernet Sauvignon} 0.001140606 0.6875000 3.769329 11  
## [5] {price\_char=42,   
## region=Columbia Valley (WA)} => {variety=Cabernet Sauvignon} 0.002903360 0.6829268 3.744256 28  
## [6] {price\_char=75,   
## region=Columbia Valley (WA)} => {variety=Cabernet Sauvignon} 0.001659063 0.6666667 3.655107 16  
## [7] {price\_char=60,   
## region=Columbia Valley (WA)} => {variety=Cabernet Sauvignon} 0.001555371 0.6521739 3.575648 15  
## [8] {points\_char=95,   
## region=Columbia Valley (WA)} => {variety=Cabernet Sauvignon} 0.001347988 0.6500000 3.563729 13  
## [9] {points\_char=89,   
## price\_char=30,   
## region=Columbia Valley (WA)} => {variety=Cabernet Sauvignon} 0.002073828 0.6250000 3.426663 20  
## [10] {points\_char=86,   
## price\_char=15,   
## region=California} => {variety=Cabernet Sauvignon} 0.001140606 0.5789474 3.174172 11

From the result, we can get rules like these:

1. For wine with points of 90, price of $40 and produced in Columbia Valley (WA), we can be 90.9% confident that the wine variety is Cabernet Sauvignon.
2. For wine with points of 92, price of $42 and produced in Columbia Valley (WA), we can be 76.5% confident that the wine variety is Cabernet Sauvignon.
3. For wine with points of 96 and produced in Columbia Valley (WA), we can be 73.7% confident that the wine variety is Cabernet Sauvignon.
4. For wine with price of $48 and produced in Columbia Valley (WA), we can be 68.8% confident that the wine variety is Cabernet Sauvignon.

…

* Generate the rules for outcome of “variety=Chardonnay”.

wine\_3\_rules5 <- apriori(wine\_3, parameter = list(supp = .001, conf = .5), appearance = list(default="lhs", rhs="variety=Chardonnay"), control=list(verbose=F))  
wine\_3\_rules5 <- sort(wine\_3\_rules5, by = "confidence", decreasing = TRUE)  
inspect(wine\_3\_rules5[1:10])

## lhs rhs support confidence lift count  
## [1] {points\_char=95,   
## price\_char=50,   
## region=Russian River Valley} => {variety=Chardonnay} 0.001140606 1.0000000 4.375681 11  
## [2] {points\_char=84,   
## price\_char=20,   
## region=Russian River Valley} => {variety=Chardonnay} 0.001036914 1.0000000 4.375681 10  
## [3] {points\_char=85,   
## price\_char=20,   
## region=Russian River Valley} => {variety=Chardonnay} 0.001244297 1.0000000 4.375681 12  
## [4] {price\_char=16,   
## region=Russian River Valley} => {variety=Chardonnay} 0.003007051 0.9354839 4.093379 29  
## [5] {points\_char=90,   
## price\_char=20,   
## region=Russian River Valley} => {variety=Chardonnay} 0.001140606 0.9166667 4.011041 11  
## [6] {points\_char=90,   
## price\_char=30,   
## region=Russian River Valley} => {variety=Chardonnay} 0.001659063 0.8888889 3.889494 16  
## [7] {price\_char=22,   
## region=Russian River Valley} => {variety=Chardonnay} 0.001866445 0.8571429 3.750583 18  
## [8] {points\_char=87,   
## price\_char=20,   
## region=Russian River Valley} => {variety=Chardonnay} 0.002281211 0.8148148 3.565369 22  
## [9] {points\_char=88,   
## price\_char=30,   
## region=Russian River Valley} => {variety=Chardonnay} 0.001451680 0.7777778 3.403307 14  
## [10] {points\_char=95,   
## price\_char=50} => {variety=Chardonnay} 0.001140606 0.7333333 3.208832 11

From the result, we can get rules like these:

1. For wine with points of 95, price of $50 and produced in Russian River Valley, we can be 100% confident that the wine variety is Chardonnay.
2. For wine with price of $16 and produced in Russian River Valley, we can be 93.5% confident that the wine variety is Chardonnay.
3. For wine with points of 90, price of $20 and produced in Russian River Valley, we can be 91.7% confident that the wine variety is Chardonnay.
4. For wine with price of $22 and produced in Russian River Valley, we can be 81.5% confident that the wine variety is Chardonnay. …

* Generate the rules for outcome of “variety=Merlot”.

wine\_3\_rules6 <- apriori(wine\_3, parameter = list(supp = .001, conf = .5), appearance = list(default="lhs", rhs="variety=Merlot"), control=list(verbose=F))  
wine\_3\_rules6 <- sort(wine\_3\_rules6, by = "confidence", decreasing = TRUE)  
inspect(wine\_3\_rules6)

## lhs rhs support confidence lift count  
## [1] {points\_char=82,   
## price\_char=9,   
## region=California} => {variety=Merlot} 0.001347988 0.8125000 7.434298 13  
## [2] {points\_char=83,   
## price\_char=8,   
## region=California} => {variety=Merlot} 0.001036914 0.5882353 5.382297 10  
## [3] {points\_char=91,   
## price\_char=28,   
## region=Columbia Valley (WA)} => {variety=Merlot} 0.001140606 0.5500000 5.032448 11  
## [4] {points\_char=83,   
## price\_char=7,   
## region=California} => {variety=Merlot} 0.001451680 0.5384615 4.926872 14  
## [5] {points\_char=82,   
## price\_char=9} => {variety=Merlot} 0.001659063 0.5333333 4.879949 16  
## [6] {price\_char=9,   
## region=Columbia Valley (WA)} => {variety=Merlot} 0.002073828 0.5263158 4.815740 20

From the result, we can get rules like these:

1. For wine with points of 85, price of $9 and produced in California, we can be 81.3% confident that the wine variety is Merlot.
2. For wine with points of 83, price of $8 and produced in California, we can be 58.8% confident that the wine variety is Merlot.
3. For wine with points of 91, price of $28 and produced in Columbia Valley (WA), we can be 55% confident that the wine variety is Merlot.
4. For wine with points of 83, price of $7 and produced in California, we can be 53.8% confident that the wine variety is Merlot.

Therefore, it seems that the combination of the three variables - points, price and region doesn’t necessarily do a better job. In the rules above, sometimes the combination of two of them can better predict the wine variety.