Modeling Problem I

Karol Orozco & Charles Hanks

Predicting Province

```r
knitr::opts_chunk$set(echo = TRUE, message = FALSE, warning = FALSE)

library(tidyverse)
library(formatR)
library(moderndive)
library(skimr)
library(caret)

wine_pinot <- readRDS(gzcon(url(
  "https://github.com/karolo89/machine_learning_assignment/raw/main/pinot.rds")))
```

```r
#adding log price column
pinot <- wine_pinot %>%
  mutate(lprice = log(price))

pinot <- pinot %>%
  mutate(id = as.factor(id))%>%
  mutate(year = as.factor(year))

summary(pinot)
```

```
      id          province              price             points
 1      :   1   Length:8380        Min.   :   7.00   Min.   :80.00
 2      :   1   Class :character   1st Qu.:  31.00   1st Qu.:88.00
 3      :   1   Mode  :character   Median :  45.00   Median :90.00
 4      :   1                      Mean   :  52.52   Mean   :89.98
 5      :   1                      3rd Qu.:  60.00   3rd Qu.:92.00
 6      :   1                      Max.   :2500.00   Max.   :98.00
```

```
 (Other):8374
      year       description             lprice
 2014    :2046   Length:8380        Min.   :1.946
 2013    :1819   Class :character   1st Qu.:3.434
 2012    :1505   Mode  :character   Median :3.807
 2015    : 815                      Mean   :3.779
 2011    : 582                      3rd Qu.:4.094
 2010    : 502                      Max.   :7.824
 (Other):1111
```

Preliminary EDA, Feature Engineering Brainstorm, Initial Thoughts

```
pinot %>%
  group_by(province) %>%
  summarize(prov_freq = n(),
            percent_of_ds = round(prov_freq/8380,2))
```

```
# A tibble: 6 x 3
  province          prov_freq percent_of_ds
  <chr>                 <int>         <dbl>
1 Burgundy               1193          0.14
2 California             3959          0.47
3 Casablanca_Valley       131          0.02
4 Marlborough             229          0.03
5 New_York                131          0.02
6 Oregon                 2737          0.33
```

```
#nearly half of wines are californian, good to know...

pinot %>%
  filter(str_detect(description, "[Oo]ak")) %>%
  nrow()
```

```
[1] 1301
```

```
#1301/8380 have the word oak in description

pinot %>% filter(str_detect(description, "[Oo]ak")) %>%
  group_by(province) %>% summarize(prov_freq = n(),
```

```r
                                     oak_perc = round(prov_freq/1301,2))
```

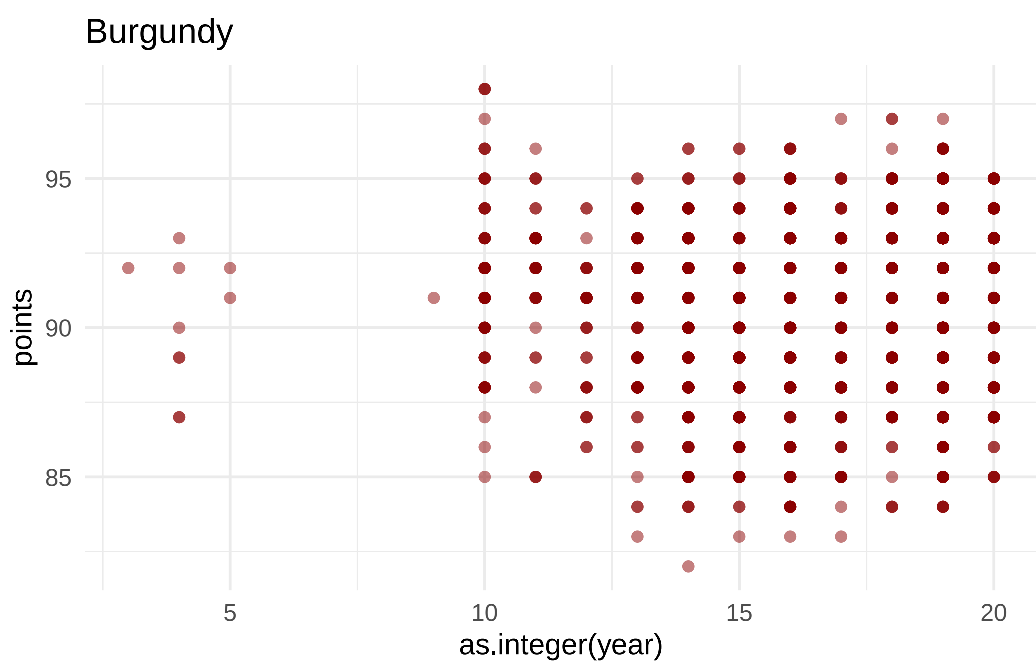
```
# A tibble: 6 x 3
  province          prov_freq oak_perc
  <chr>                 <int>    <dbl>
1 Burgundy                  8     0.01
2 California              739     0.57
3 Casablanca_Valley        64     0.05
4 Marlborough              32     0.02
5 New_York                  9     0.01
6 Oregon                  449     0.35
```

```r
#it is likely California or Oregon if there is oak in the description

#some french language patterns to think about developing a regex from:
# "_de_" / "d'"
# "name-name"
# accented letters: "é","ô",
# "St."

pinot %>%
  group_by(province) %>%
  summarize(avgPrice = mean(price),
            avgPoints = mean(points))
```

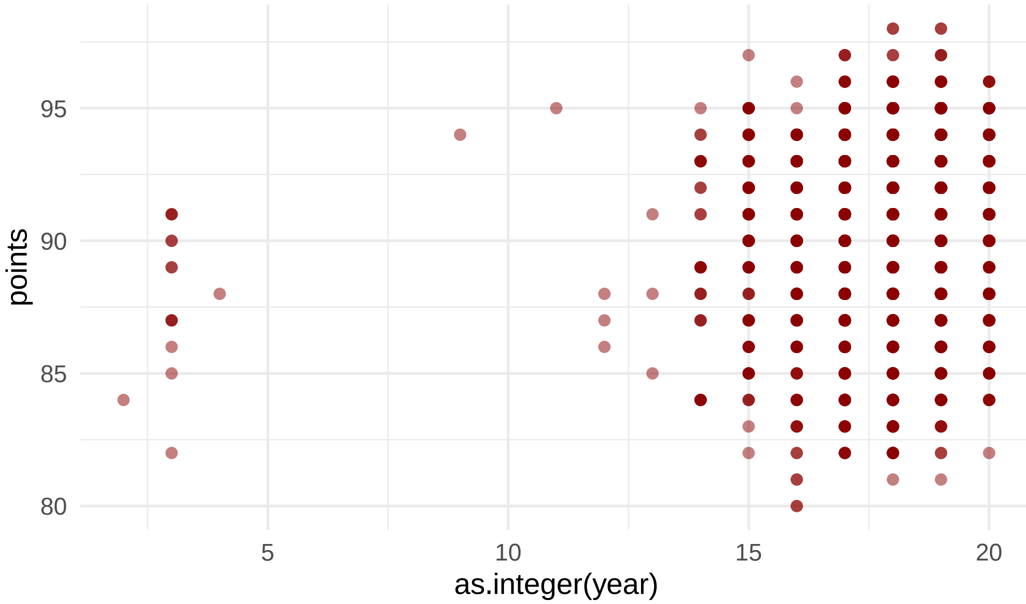
```
# A tibble: 6 x 3
  province          avgPrice avgPoints
  <chr>                <dbl>     <dbl>
1 Burgundy              98.0      90.4
2 California            47.5      90.5
3 Casablanca_Valley     21.1      86.3
4 Marlborough           27.7      87.6
5 New_York              25.7      87.7
6 Oregon                44.9      89.5
```

```r
# Burgundy wines are on average significantly more expensive...
# and casablanca valley wines on average have the lowest price and score.

#which wines do people recommend waiting before drinking? i.e "drink from XXXX"
```

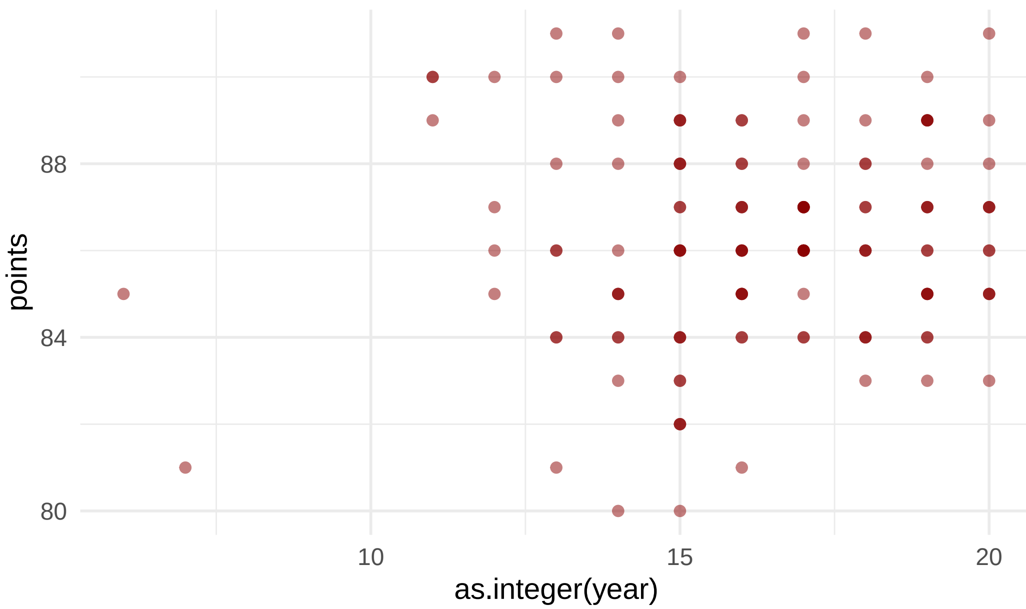
```r
#some words to check out: "edge","tannins","dense","firm", oregon pinot is fruity.

province_vec = c("Burgundy", "California", "Casablanca_Valley","Marlborough",
                 "New_York", "Oregon")

for(i in province_vec){
  plot = ggplot(pinot %>%
                  filter(province == i), aes(x = as.integer(year), y = points)) +
    geom_point(alpha =.5, color = "red4") +
    ggtitle(i)+
    theme_minimal()

  print(plot)
}
```

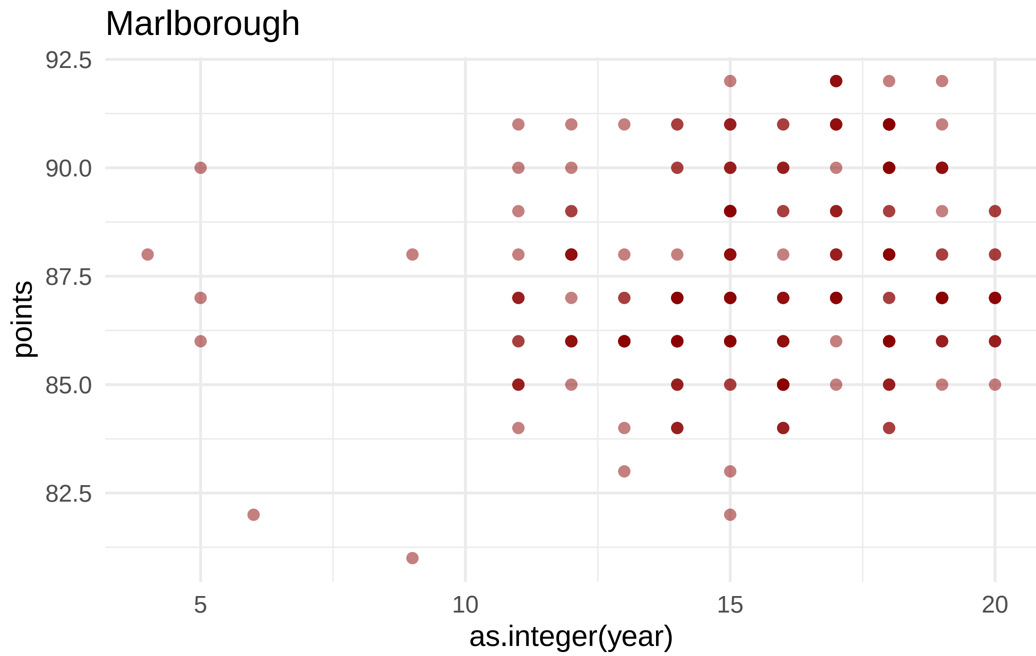


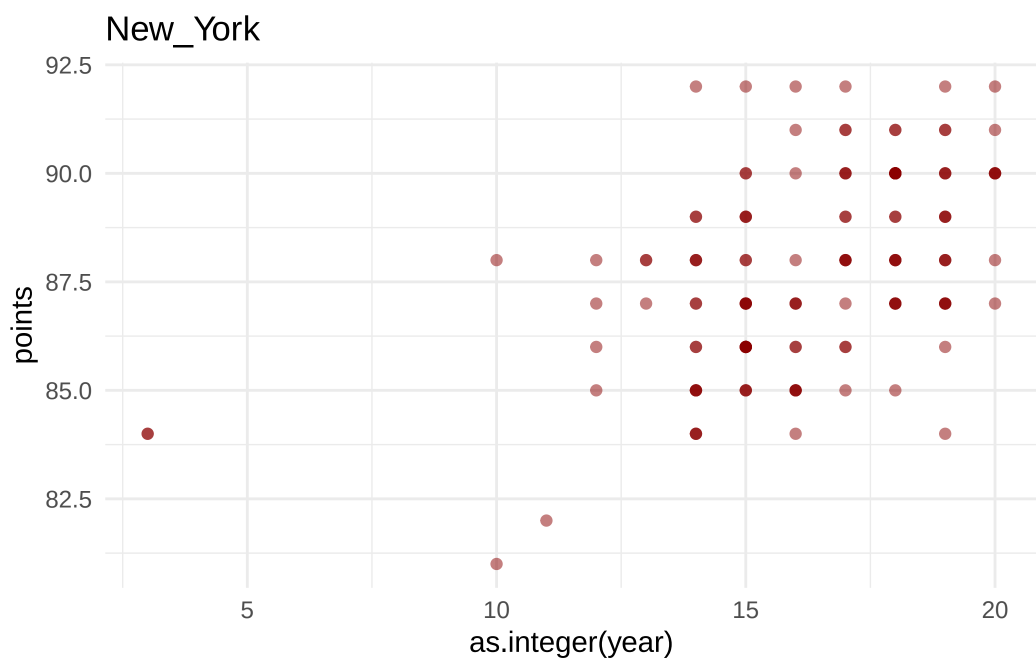
California

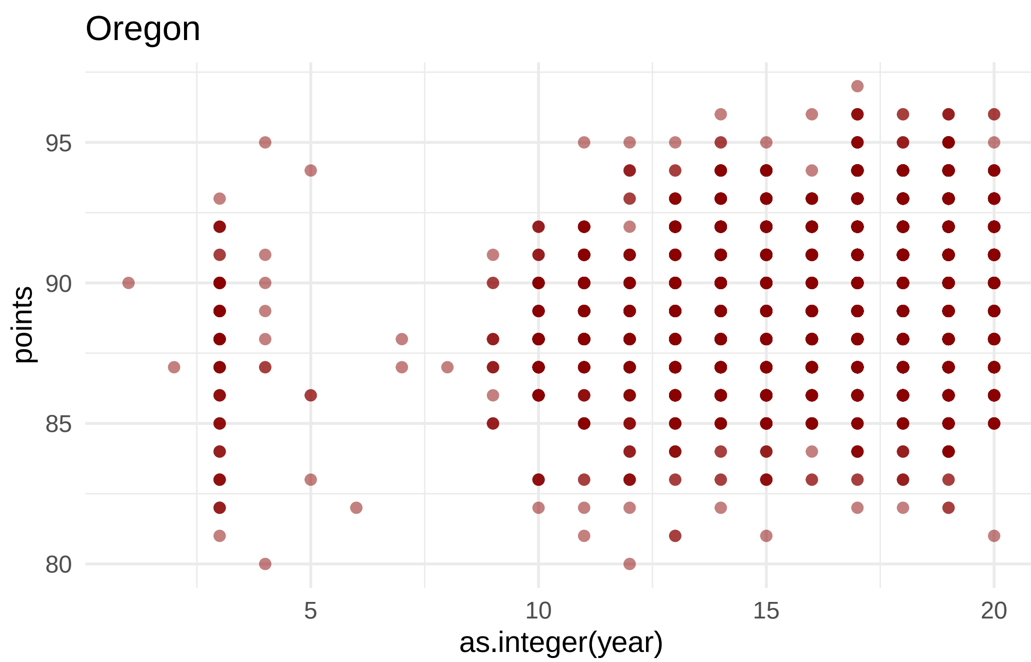


Casablanca_Valley

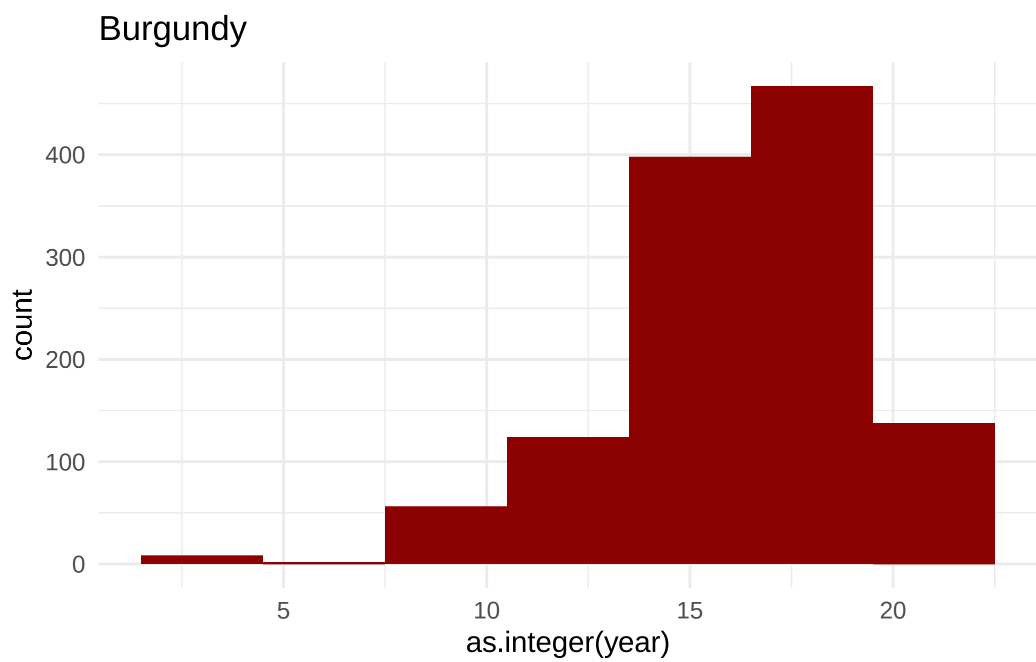


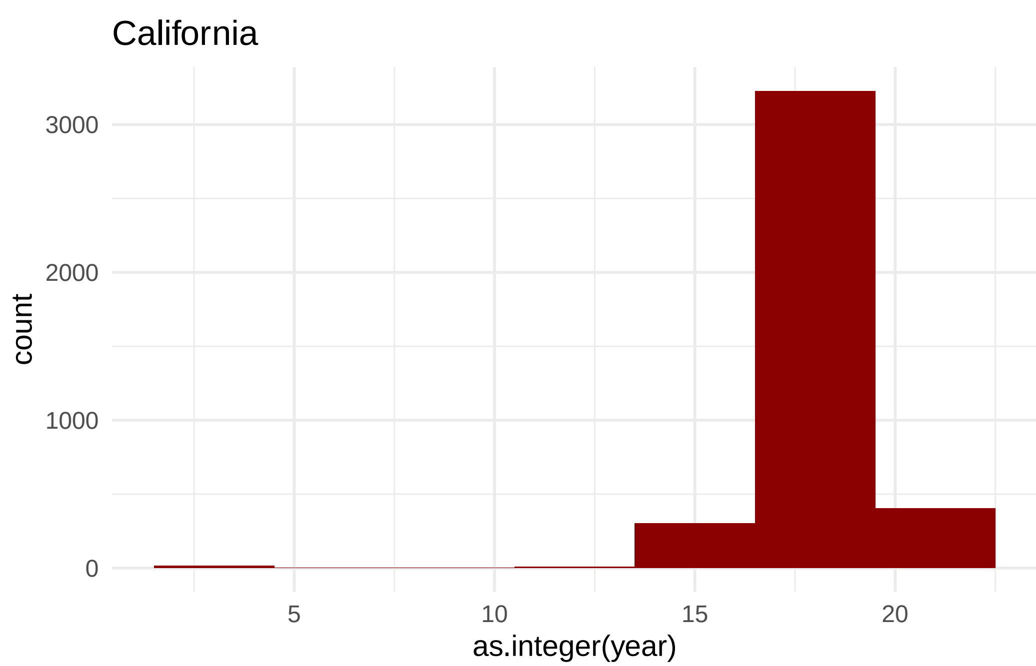

Marlborough
92.5
90.0
87.5
85.0
82.5
points
5
10
15
20
as.integer(year)

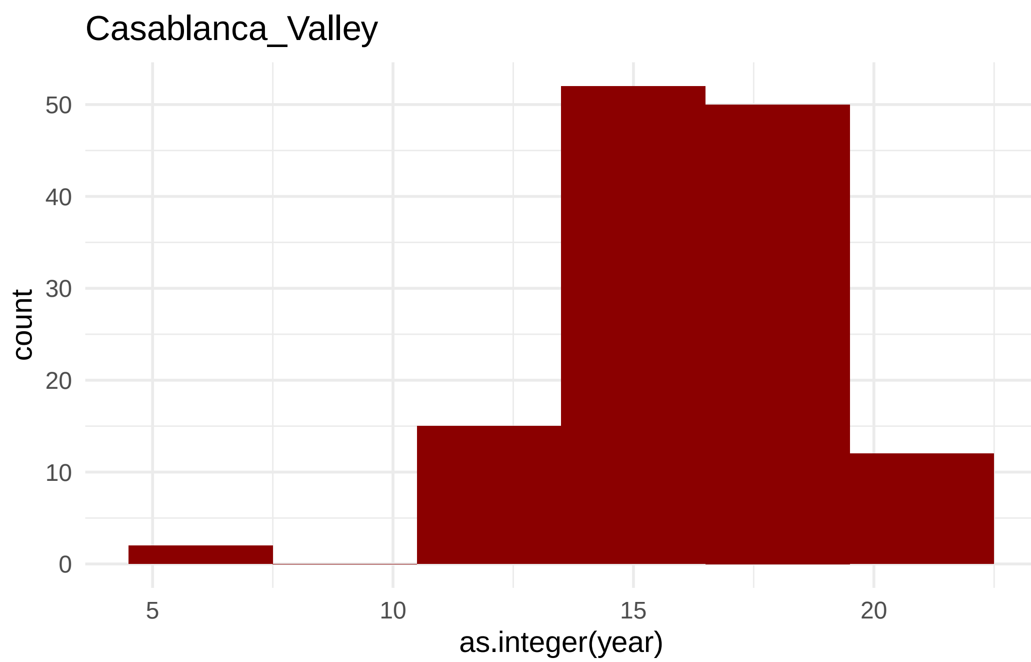

New_York
92.5
90.0
87.5
85.0
82.5
points
5
10
15
20
as.integer(year)

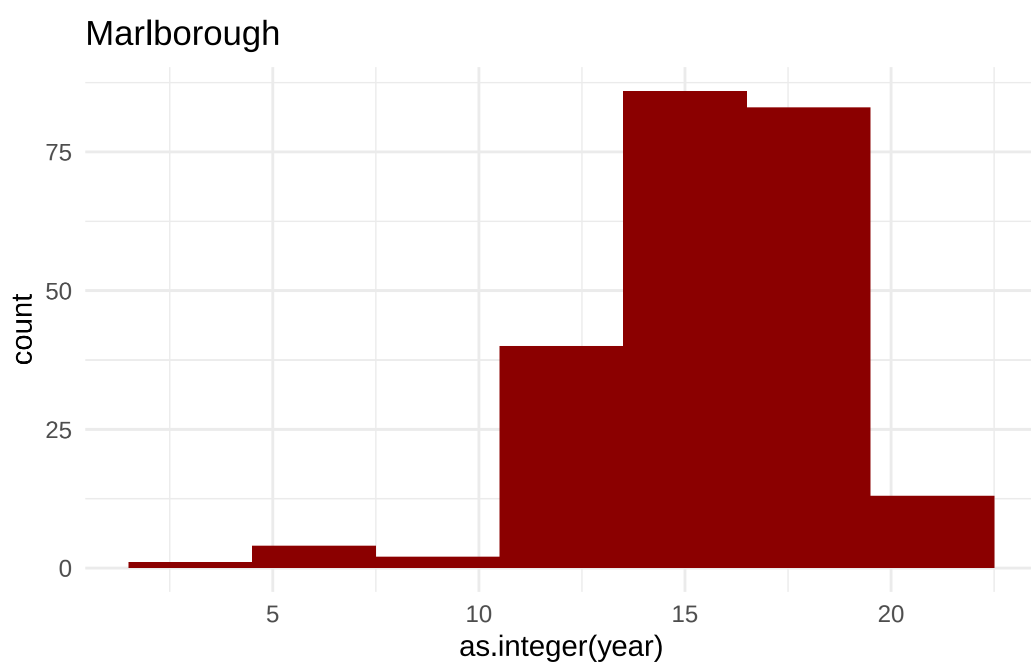


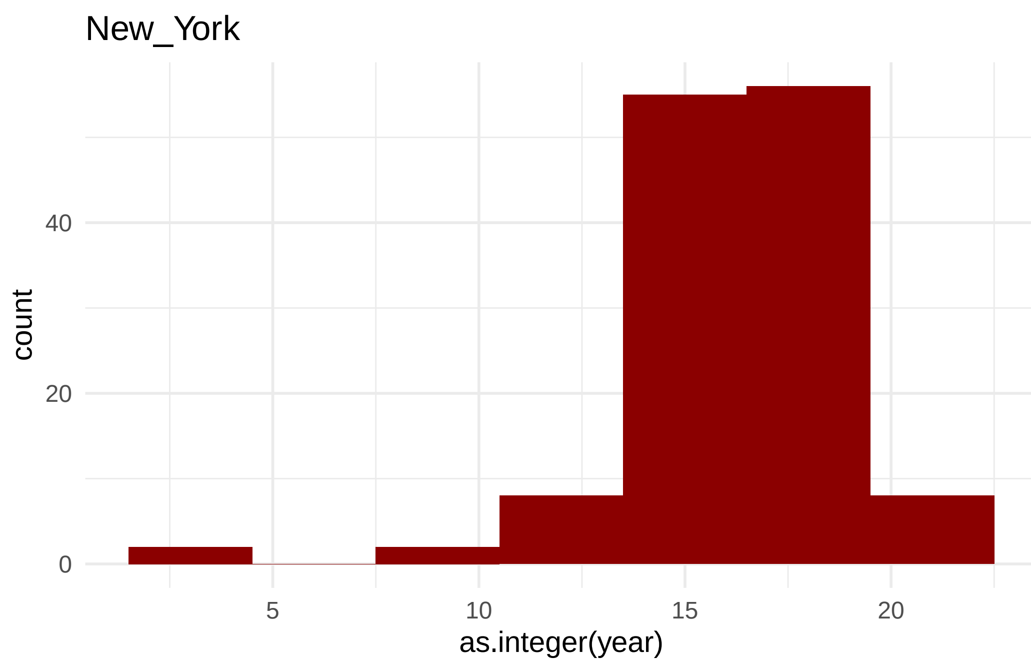

```
for(i in province_vec){
  plot2 = ggplot(pinot %>%
                   filter(province == i), aes(x = as.integer(year))) +
    geom_histogram(binwidth =3, fill = "red4") +
    ggtitle(i)+
    theme_minimal()
  print(plot2)
}
```

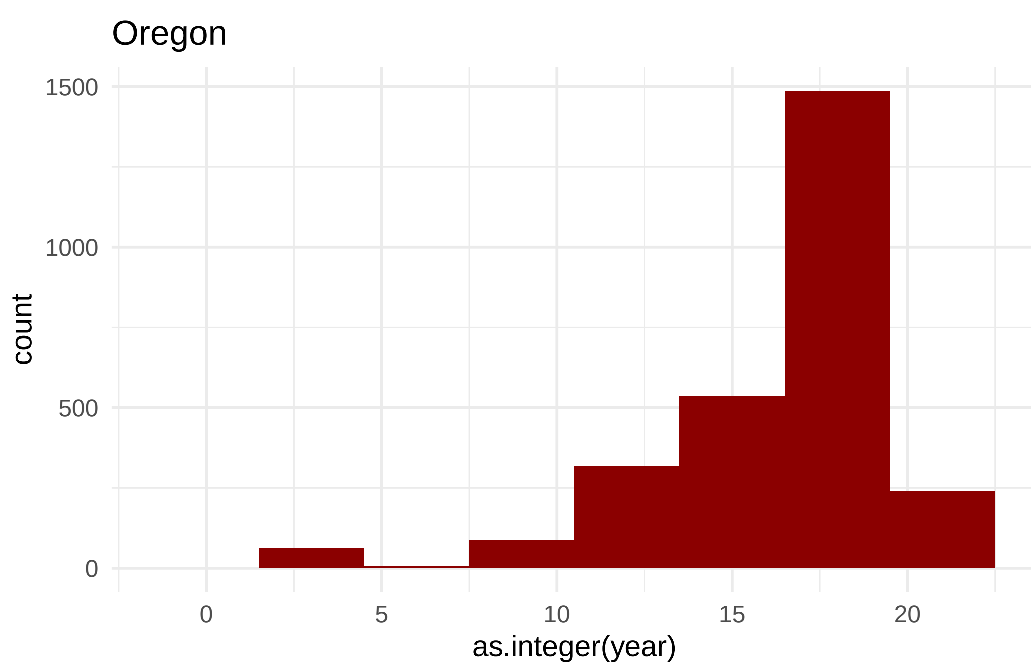

Burgundy
count
400
300
200
100
0
5
10
15
20
as.integer(year)


California
count
3000
2000
1000
0
5
10
15
20
as.integer(year)


Casablanca_Valley
count
0
10
20
30
40
50
5
10
15
20
as.integer(year)


Marlborough
count
0
25
50
75
5
10
15
20
as.integer(year)


New_York
40
20
0
count
5
10
15
20
as.integer(year)


Oregon
1500
1000
500
0
count
0
5
10
15
20
as.integer(year)

```
#Some findings from viz:
#california pinot noir production did not begin until ~2008, then exploded!
#before year 2000, likely to be oregon
#burgundy pinots score high around 2005,
#after almost no burgundy pinots between 2000 and 2005
#California pinot game WAY STRONG between 2010 and 2015
#New York pinot score high between 2008 and 2015
#What happened around 2014?? Counts drop across provinces....
```

Preprocessing (3pts)

1. Preprocess the dataframe that you created in the previous question using centering and scaling of the numeric features
2. Create dummy variables for the year factor column

Running KNN (5pts)

1. Split your data into an 80/20 training and test set
2. Use Caret to run a KNN model that uses your engineered features to predict province

- use 5-fold cross validated subsampling
- allow Caret to try 15 different values for K

3. Display the confusion matrix on the test data

Kappa (2pts)

Is this a good value of Kappa? Why or why not?

Answer: (write your answer here)

Improvement (2pts)

Looking at the confusion matrix, where do you see room for improvement in your predictions?

Answer: (write your answer here)

KNN Model, 5 fold CV resampling:

```
w = wine_pinot %>% mutate(lprice = log(price),
                    fyear = as.factor(year),
                    oak = as.integer(str_detect(description, "[Oo]ak")),
                    earth = as.integer(str_detect(description, "[Ee]arth")),
                    cherry = as.integer(str_detect(description, "[Cc]herry")),
                    choc = as.integer(str_detect(description, "[Cc]hocolate")),
                    acidity = as.integer(str_detect(description, "[Aa]cidity")),
                    nose = as.integer(str_detect(description, "[Nn]ose")),
                    palate = as.integer(str_detect(description, "[Pp]alate")),
                    chocolate = as.integer(str_detect(description,"[Cc]hocolate")),
                    tart = as.integer(str_detect(description,"[Tt]art")),
                    brisk = as.integer(str_detect(description,"[Bb]risk")),
                    bramble = as.integer(str_detect(description,"[Bb]ramble")),
                    aging = as.integer(str_detect(description, "[Aa]ging")),
                    savory =as.integer(str_detect(description, "[Ss]avory")),
                    clover = as.integer(str_detect(description, "[Cc]love")),
                    aromas = as.integer(str_detect(description, "[Aa]romas")),
                    fruits = as.integer(str_detect(description, "[Ff]ruits")),
                    nose = as.integer(str_detect(description, "[Nn]ose")),
                    points_greater_95 = points >=95,
                    points_less_90 = points <= 90,
                    price_greater_4 = lprice >= 4,
                    price_between_4_3 = lprice < 4 & lprice >= 3,
                    price_less_3 = lprice < 3,
                    before_2010 = year < 2010,
                    beween_2010_2015 = (year >=2010 & year <= 2015),
                    between_2015_2020 = (year > 2015 & year <= 2020)) %>%
                    select(-id,-price,-description)
```

```
set.seed(504)

wine_index <- createDataPartition(w$province, p = 0.8, list = FALSE)
train <- w[ wine_index, ]
test <- w[-wine_index, ]

control <- trainControl(method = "cv", number = 5)

fit <- train(province ~ .,
    data = train,
```

```r
       method = "knn",
       tuneLength = 15,
       trControl = control)

fit
```

```
k-Nearest Neighbors

6707 samples
  28 predictor
   6 classes: 'Burgundy', 'California', 'Casablanca_Valley', 'Marlborough', 'New_York', 'Oreg

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 5365, 5365, 5367, 5366, 5365
Resampling results across tuning parameters:

  k   Accuracy   Kappa
   5  0.6739160  0.4911401
   7  0.6788369  0.4975536
   9  0.6804783  0.4986291
  11  0.6803334  0.4979349
  13  0.6836109  0.5017083
  15  0.6779470  0.4913296
  17  0.6777958  0.4899909
  19  0.6831631  0.4972199
  21  0.6797354  0.4908316
  23  0.6803300  0.4909230
  25  0.6828647  0.4940488
  27  0.6810756  0.4901445
  29  0.6815246  0.4901390
  31  0.6788401  0.4851455
  33  0.6779474  0.4833832

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 13.
```

```r
confusionMatrix(predict(fit,test), factor(test$province))
```

```
Confusion Matrix and Statistics
```

```
                   Reference
Prediction          Burgundy California Casablanca_Valley Marlborough New_York
  Burgundy               166         19                 0           4        2
  California              15        616                13           9       11
  Casablanca_Valley        1          1                 1           0        1
  Marlborough              0          0                 2           1        0
  New_York                 0          0                 0           0        0
  Oregon                  56        155                10          31       12
                   Reference
Prediction          Oregon
  Burgundy              36
  California           129
  Casablanca_Valley      0
  Marlborough            3
  New_York               0
  Oregon               379

Overall Statistics

               Accuracy : 0.6952
                 95% CI : (0.6725, 0.7172)
    No Information Rate : 0.4728
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5167

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Burgundy Class: California Class: Casablanca_Valley
Sensitivity                  0.69748            0.7788                0.0384615
Specificity                  0.95749            0.7993                0.9981785
Pos Pred Value               0.73128            0.7768                0.2500000
Neg Pred Value               0.95021            0.8011                0.9850210
Prevalence                   0.14226            0.4728                0.0155409
Detection Rate               0.09922            0.3682                0.0005977
Detection Prevalence         0.13568            0.4740                0.0023909
Balanced Accuracy            0.82749            0.7890                0.5183200
                     Class: Marlborough Class: New_York Class: Oregon
Sensitivity                   0.0222222         0.00000        0.6929
Specificity                   0.9969287         1.00000        0.7655
```

```
Pos Pred Value              0.1666667          NaN        0.5894
Neg Pred Value              0.9736053      0.98446        0.8369
Prevalence                  0.0268978      0.01554        0.3270
Detection Rate              0.0005977      0.00000        0.2265
Detection Prevalence        0.0035864      0.00000        0.3843
Balanced Accuracy           0.5095755      0.50000        0.7292
```

Group Activity: Naive Bayes Model

Use the top words by province to…

1. Engineer more features that capture the essence of Casablanca, Marlborough and New York

2. Look for difference between California and Oregon

3. Use what you find to run naive Bayes models that achieve a Kappa that approaches 0.5

```r
library(tidytext)
library(caret)
wine = wine_pinot
names(wine)[names(wine) == 'id'] = 'ID'
```

Document term matrix:

```r
df <- wine %>%
  unnest_tokens(word, description) %>%
  anti_join(stop_words) %>% # get rid of stop words
  filter(word != "wine") %>%
  filter(word != "pinot") %>%
  count(ID, word) %>%
  group_by(ID) %>%
  mutate(freq = n/sum(n)) %>%
  mutate(exists = (n>0)) %>%
  ungroup %>%
  group_by(word) %>%
  mutate(total = sum(n))
```

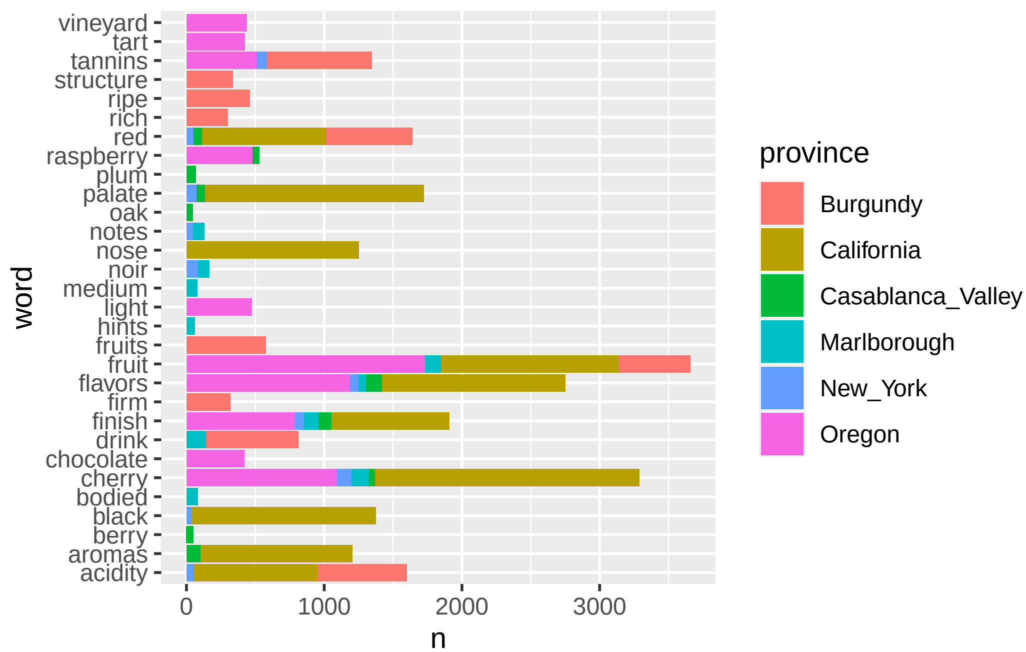
Pivot wide and rejoin with wine:

```r
wino <- df %>%
  filter(total > 900) %>%
  pivot_wider(id_cols = ID, names_from = word, values_from = exists, values_fill = list(ex
```

```
  merge(select(wine,ID, province), all.y=TRUE) #%>%
  #drop_na()

#wino <- merge(select(wine,ID, province), wino, by="ID", all.x=TRUE) %>%
#  arrange(ID)
#View(wino)
wino <- replace(wino, is.na(wino), FALSE)
```

Visualizing distribution to select distinct features for provinces:

```
df %>%
  left_join(select(wine, ID, province), by = "ID") %>%
  count(province, word) %>%
  group_by(province) %>%
  top_n(10,n) %>%
  arrange(province, desc(n)) %>%
  ggplot(aes(x = word, y = n, fill = province)) + geom_col() + coord_flip()
```




```
wino = wino %>% select(ID, province, tart, plum, oak, bodied,black,nose,palate,ripe,cherry
```

train & test model:

```r
wine_index <- createDataPartition(wino$province, p = 0.80, list = FALSE)
train <- wino[ wine_index, ]
test <- wino[-wine_index, ]

fit <- train(province ~ .,
             data = train,
             method = "naive_bayes",
             tuneGrid = expand.grid(usekernel = c(T,F), laplace = T, adjust = T),
             metric = "Kappa",
             trControl = trainControl(method = "cv"))


confusionMatrix(predict(fit, test),factor(test$province))
```

```
Confusion Matrix and Statistics

                   Reference
Prediction          Burgundy California Casablanca_Valley Marlborough New_York
  Burgundy               223        142                 3          15        5
  California               6        481                12           6       15
  Casablanca_Valley        0         16                 6           2        0
  Marlborough              5         17                 0          14        1
  New_York                 1         14                 1           0        3
  Oregon                   3        121                 4           8        2
                   Reference
Prediction          Oregon
  Burgundy             271
  California            65
  Casablanca_Valley      9
  Marlborough           11
  New_York               3
  Oregon               188

Overall Statistics

               Accuracy : 0.5469
                 95% CI : (0.5227, 0.571)
    No Information Rate : 0.4728
    P-Value [Acc > NIR] : 7.579e-10

                  Kappa : 0.3651
```

```
 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: Burgundy Class: California Class: Casablanca_Valley
Sensitivity                   0.9370            0.6081                 0.230769
Specificity                   0.6962            0.8821                 0.983607
Pos Pred Value                0.3384            0.8222                 0.181818
Neg Pred Value                0.9852            0.7151                 0.987805
Prevalence                    0.1423            0.4728                 0.015541
Detection Rate                0.1333            0.2875                 0.003586
Detection Prevalence          0.3939            0.3497                 0.019725
Balanced Accuracy             0.8166            0.7451                 0.607188
                     Class: Marlborough Class: New_York Class: Oregon
Sensitivity                    0.311111        0.115385        0.3437
Specificity                    0.979115        0.988464        0.8774
Pos Pred Value                 0.291667        0.136364        0.5767
Neg Pred Value                 0.980923        0.986069        0.7335
Prevalence                     0.026898        0.015541        0.3270
Detection Rate                 0.008368        0.001793        0.1124
Detection Prevalence           0.028691        0.013150        0.1949
Balanced Accuracy              0.645113        0.551924        0.6106
```

Creating more features

```
features = wine %>%
  mutate(aging = str_detect(description,"aging"),
         chocolate =  str_detect(description, "chocolate"),
         vineyard = str_detect(description, "vineyard")) %>%
            select(ID,aging,chocolate,vineyard)

wino2 = wino %>%
  left_join(features, by = "ID")
```

Test 2

```
wine_index <- createDataPartition(wino2$province, p = 0.80, list = FALSE)
train <- wino2[ wine_index, ]
test <- wino2[-wine_index, ]

fit <- train(province ~ .,
             data = train,
```

```r
               method = "naive_bayes",
               tuneGrid = expand.grid(usekernel = c(T,F), laplace = T, adjust = T),
               metric = "Kappa",
               trControl = trainControl(method = "cv"))
fit
```

```
Naive Bayes

6707 samples
  15 predictor
   6 classes: 'Burgundy', 'California', 'Casablanca_Valley', 'Marlborough', 'New_York', 'Oreg

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 6035, 6037, 6036, 6037, 6037, 6038, ...
Resampling results across tuning parameters:

  usekernel  Accuracy   Kappa
  FALSE      0.3049002  0.2057034
   TRUE      0.5937064  0.3956332

Tuning parameter 'laplace' was held constant at a value of TRUE

Tuning parameter 'adjust' was held constant at a value of TRUE
Kappa was used to select the optimal model using the largest value.
The final values used for the model were laplace = TRUE, usekernel = TRUE
 and adjust = TRUE.
```

```r
confusionMatrix(predict(fit, test),factor(test$province))
```

```
Confusion Matrix and Statistics

                   Reference
Prediction          Burgundy California Casablanca_Valley Marlborough New_York
  Burgundy               233        151                 6          27        7
  California               3        601                18          16       19
  Casablanca_Valley        0          0                 0           0        0
  Marlborough              0          0                 0           0        0
  New_York                 0          0                 0           0        0
  Oregon                   2         39                 2           2        0
```

```
                   Reference
Prediction          Oregon
  Burgundy             231
  California           182
  Casablanca_Valley      0
  Marlborough            0
  New_York               0
  Oregon               134

Overall Statistics

               Accuracy : 0.5786
                 95% CI : (0.5545, 0.6024)
    No Information Rate : 0.4728
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.3731

 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: Burgundy Class: California Class: Casablanca_Valley
Sensitivity                   0.9790            0.7598                  0.00000
Specificity                   0.7059            0.7302                  1.00000
Pos Pred Value                0.3557            0.7163                      NaN
Neg Pred Value                0.9951            0.7722                  0.98446
Prevalence                    0.1423            0.4728                  0.01554
Detection Rate                0.1393            0.3592                  0.00000
Detection Prevalence          0.3915            0.5015                  0.00000
Balanced Accuracy             0.8425            0.7450                  0.50000
                     Class: Marlborough Class: New_York Class: Oregon
Sensitivity                      0.0000         0.00000        0.2450
Specificity                      1.0000         1.00000        0.9600
Pos Pred Value                      NaN             NaN        0.7486
Neg Pred Value                   0.9731         0.98446        0.7236
Prevalence                       0.0269         0.01554        0.3270
Detection Rate                   0.0000         0.00000        0.0801
Detection Prevalence             0.0000         0.00000        0.1070
Balanced Accuracy                0.5000         0.50000        0.6025
```

```
#Higher kappa value, but now model is not predicting any of the sparse provinces
```