ISYE 6501 : Homework 10
3/31/2021

Table of Contents

14.1.1 Mean/Mode imputation

Using data from the breast cancer data set I was able to first validate that there was missing data, determine the column containing the missing information and replace those missing values with mode and mean data.

CODE:

```
rm(list = ls())
set.seed(42)
setwd("~/Documents/ISYE6501 Intro to Analytics Modeling/FA_SP_hw10")

#Library
install.packages("tidyverse")
library(tidyverse)
install.packages("Hmisc")
library(Hmisc)

#Ingest data into a dataframe

newdata <- function() {
  datacancer <- read.table("breast-cancer-wisconsin.data.txt", header=FALSE,
                           sep=",", stringsAsFactors = FALSE)

  return(datacancer)
}

raw_cancerdf <- newdata()
summary(raw_cancerdf)

#Viewing the raw data shows V7 has the ?s
#Determine the number of rows with ?s

colSums(raw_cancerdf == '?')
```

```
#calculate the mode for all rows of column V7
Mode = function(x){
  ta = table(x)
  tam = max(ta)
  if (all(ta == tam))
    mod = NA
  else
    if(is.numeric(x))
      mod = as.numeric(names(ta)[ta == tam])
  else
    mod = names(ta)[ta == tam]
  return(mod)
}

modeinfo = Mode(raw_cancerdf$V7)
modeinfo

row_missingdata <- which(raw_cancerdf$V7 == '?', arr.ind=T)
row_missingdata

#Create a dataframe from the raw data to update ?s with the mode
mode_cancerdf <- raw_cancerdf

#Using a for statement replace all ?s with the mode of column V7

for (i in 1:nrow(mode_cancerdf)){
  if(mode_cancerdf$V7[i] == '?') {
    print('? data found')
    mode_cancerdf$V7[i] = modeinfo
  }
}

mode_cancerdf

#Use impute method to replace the ?s with the mean for column V7

mean_cancerdf <- raw_cancerdf

meanV7<- mean(as.integer(mean_cancerdf[-row_missingdata, 'V7']))
meanV7


mean_cancerdf[row_missingdata, 'V7']<- as.integer(meanV7)
mean_cancerdf[row_missingdata, 'V7']
```

```
mean_cancerdf
```

```
OUTPUT:
summary(raw_cancerdf)
       V1                V2               V3               V4        
 Min.   :   61634   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000  
 1st Qu.:  870688   1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 1.000  
 Median : 1171710   Median : 4.000   Median : 1.000   Median : 1.000  
 Mean   : 1071704   Mean   : 4.418   Mean   : 3.134   Mean   : 3.207  
 3rd Qu.: 1238298   3rd Qu.: 6.000   3rd Qu.: 5.000   3rd Qu.: 5.000  
 Max.   :13454352   Max.   :10.000   Max.   :10.000   Max.   :10.000  

       V5               V6              V7                  V8        
 Min.   : 1.000   Min.   : 1.000   Length:699         Min.   : 1.000  
 1st Qu.: 1.000   1st Qu.: 2.000   Class :character   1st Qu.: 2.000  
 Median : 1.000   Median : 2.000   Mode  :character   Median : 3.000  
 Mean   : 2.807   Mean   : 3.216                      Mean   : 3.438  
 3rd Qu.: 4.000   3rd Qu.: 4.000                      3rd Qu.: 5.000  
 Max.   :10.000   Max.   :10.000                      Max.   :10.000  

       V9              V10              V11      
 Min.   : 1.000   Min.   : 1.000   Min.   :2.00  
 1st Qu.: 1.000   1st Qu.: 1.000   1st Qu.:2.00  
 Median : 1.000   Median : 1.000   Median :2.00  
 Mean   : 2.867   Mean   : 1.589   Mean   :2.69  
 3rd Qu.: 4.000   3rd Qu.: 1.000   3rd Qu.:4.00  
 Max.   :10.000   Max.   :10.000   Max.   :4.00  
```

```
colSums(raw_cancerdf == '?')
V1  V2  V3  V4  V5  V6  V7  V8  V9  V10 V11 
 0   0   0   0   0   0  16   0   0   0   0 

modeinfo
[1] "1"

row_missingdata
 [1]  24  41 140 146 159 165 236 250 276 293 295 298 316 322 412 618
```

After completion of the replacement of ?s with the mode the columns no longer show missing values.

```
colSums(mode_cancerdf == '?')
 V1  V2  V3  V4  V5  V6  V7  V8  V9 V10 V11 
  0   0   0   0   0   0   0   0   0   0   0 
```

14.1.2 Use Regression imputation

Using the same logic, V7 is the column that is defined to have data. Linear regression is used to train the model with V7 as the response. Stepwise regression is then used to identify optimal factors for retraining the model.

```
CODE:

reg_cancerdf <- raw_cancerdf

#determine the variables that have missing data and the numbers

row_missingdata <- which(raw_cancerdf$V7 == '?', arr.ind=T)
row_missingdata

reg_cancerdf_lm <-(reg_cancerdf[-row_missingdata,2:10])
reg_cancerdf_lm$V7 <- as.integer(reg_cancerdf_lm$V7)

linear_model <- lm(V7~., data = reg_cancerdf_lm)
summary(linear_model)

#Using stepwise regression determine the optimal factors
step(linear_model)

#Using data from step process train model

linear_model2 <- lm(V7~ + V2 + V4 +V5 + V8, data = reg_cancerdf_lm)
summary(linear_model2)
```

```
OUTPUT:

row_missingdata
 [1]  24  41 140 146 159 165 236 250 276 293 295 298 316 322 412 618

Call:
lm(formula = V7 ~ ., data = reg_cancerdf_lm)

Residuals:
   Min        1Q        Median    3Q       Max
-9.7316    -0.9426    -0.3002   0.6725   8.6998

Coefficients:
              Estimate    Std. Error   t value   Pr(>|t|)
(Intercept)   -0.616652   0.194975     -3.163    0.00163 **
```

```
V2              0.230156     0.041691     5.521          4.83e-08 ***
V3              -0.067980    0.076170     -0.892         0.37246
V4              0.340442     0.073420     4.637          4.25e-06 ***
V5              0.339705     0.045919     7.398          4.13e-13 ***
V6              0.090392     0.062541     1.445          0.14883
V8              0.320577     0.059047     5.429          7.91e-08 ***
V9              0.007293     0.044486     0.164          0.86983
V10             -0.075230    0.059331     -1.268         0.20524
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.274 on 674 degrees of freedom
Multiple R-squared:  0.615,           Adjusted R-squared:  0.6104
F-statistic: 134.6 on 8 and 674 DF,  p-value: < 2.2e-16

step(linear_model)
Start:  AIC=1131.43
V7 ~ V2 + V3 + V4 + V5 + V6 + V8 + V9 + V10

       Df Sum of Sq    RSS    AIC
- V9    1     0.139 3486.8 1129.5
- V3    1     4.120 3490.8 1130.2
- V10   1     8.317 3495.0 1131.0
<none>              3486.6 1131.4
- V6    1    10.806 3497.5 1131.5
- V4    1   111.227 3597.9 1150.9
- V8    1   152.482 3639.1 1158.7
- V2    1   157.657 3644.3 1159.6
- V5    1   283.119 3769.8 1182.8

Step:  AIC=1129.45
V7 ~ V2 + V3 + V4 + V5 + V6 + V8 + V10

       Df Sum of Sq    RSS    AIC
- V3    1     4.028 3490.8 1128.2
- V10   1     8.179 3495.0 1129.0
<none>              3486.8 1129.5
- V6    1    11.211 3498.0 1129.7
- V4    1   114.768 3601.6 1149.6
- V2    1   158.696 3645.5 1157.8
- V8    1   160.776 3647.6 1158.2
- V5    1   285.902 3772.7 1181.3

Step:  AIC=1128.24
V7 ~ V2 + V4 + V5 + V6 + V8 + V10
```

```
       Df Sum of Sq    RSS    AIC
- V6    1     8.606 3499.4 1127.9
- V10   1     8.889 3499.7 1128.0
<none>              3490.8 1128.2
- V4    1   153.078 3643.9 1155.6
- V2    1   155.308 3646.1 1156.0
- V8    1   157.123 3647.9 1156.3
- V5    1   282.133 3772.9 1179.3

Step:  AIC=1127.92
V7 ~ V2 + V4 + V5 + V8 + V10

       Df Sum of Sq    RSS    AIC
- V10   1     5.562 3505.0 1127.0
<none>              3499.4 1127.9
- V2    1   159.594 3659.0 1156.4
- V8    1   169.954 3669.4 1158.3
- V4    1   206.785 3706.2 1165.1
- V5    1   295.807 3795.2 1181.3

Step:  AIC=1127.01
V7 ~ V2 + V4 + V5 + V8

       Df Sum of Sq    RSS    AIC
<none>              3505.0 1127.0
- V2    1    155.70 3660.7 1154.7
- V8    1    172.42 3677.4 1157.8
- V4    1    201.22 3706.2 1163.1
- V5    1    290.68 3795.7 1179.4

Call:
lm(formula = V7 ~ V2 + V4 + V5 + V8, data = reg_cancerdf_lm)

Coefficients:
(Intercept)           V2           V4           V5           V8
    -0.5360       0.2262       0.3173       0.3323       0.3238

>
> #Using data from step process train model
>
> linear_model2 <- lm(V7~ + V2 + V4 +V5 + V8, data = reg_cancerdf_lm)
> summary(linear_model2)

Call:
lm(formula = V7 ~ +V2 + V4 + V5 + V8, data = reg_cancerdf_lm)
```

```
Residuals:
    Min      1Q  Median      3Q     Max 
-9.8115 -0.9531 -0.3111  0.6678  8.6889 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.53601    0.17514  -3.060   0.0023 ** 
V2           0.22617    0.04121   5.488 5.75e-08 ***
V4           0.31729    0.05086   6.239 7.76e-10 ***
V5           0.33227    0.04431   7.499 2.03e-13 ***
V8           0.32378    0.05606   5.775 1.17e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.274 on 678 degrees of freedom
Multiple R-squared:  0.6129,	Adjusted R-squared:  0.6107 
F-statistic: 268.4 on 4 and 678 DF,  p-value: < 2.2e-16
```

Based on the new model the missing values are now imputed using regression and displayed.

```
CODE:

impute_model <- predict(linear_model2, reg_cancerdf[row_missingdata,])
impute_model
```

```
OUTPUT:

impute_model
       24        41       140       146       159       165       236       250 
5.4585352 7.9816106 0.9872832 1.6218560 0.9807851 2.2157441 2.7152652 
1.7634059 
      276       293       295       298       316       322       412       618 
2.0741942 6.0866099 0.9872832 2.5265324 5.2438347 1.7634059 0.9872832 
0.6634986 
```

<u>14.1.3 Regression with perturbation</u>

```
CODE:

cancerdf.perturbation <- raw_cancerdf
cancerdf.perturbation [row_missingdata, 'V7'] <- 
rnorm(length(impute_model),impute_model, sd(impute_model))

cancerdf.perturbation$V7 <- as.integer(cancerdf.perturbation$V7)
```

```
cancerdf.perturbation$V7
OUTPUT:

cancerdf.perturbation$V7
  [1]  1 10  2  4  1 10 10  1  1  1  1  1  3  3  9  1  1  1 10  1 10  7  1  8  1
 [26]  7  1  1  1  1  1  1  5  1  1  1  1  1 10  7  6  3 10  1  1  1  9  1  1  8
 [51]  3  4  5  8  8  5  6  1 10  2  3  2  8  2  1  2  1 10  9  1  1  2  1 10  4
 [76]  2  1  1  3  1  1  1  1  2  9  4  8 10  1  1  1  1  1  1  1  1  1  1  6 10
[101]  5  5  1  3  1  3 10 10  1  9  2  9 10  8  3  5  2 10  3  2  1  2 10 10  7
[126]  1 10  1 10  1  1  1 10  1  1  2  1  1  1  1  1  1  5  5  1  3  8  2  1 10
[151]  1 10  5  3  1 10  1  1  1 10 10  1  1  3  1  2 10  1  1  1  1  1  1 10 10
[176] 10  1  1  1 10  1  1  1 10 10  1  8 10  8  1  8 10  1  1  1  1  7  1  1  1
[201] 10 10  1  1  1 10  5  1  1  1 10  8  1 10 10  5  1  1  4  1  1 10  5  8 10
[226]  1 10  5  1 10  7  8  1 10  1  6 10  2  9 10  2  1  1  5  1  2 10  9  1  1
[251]  1 10 10 10  8 10  1  1  1  8 10 10 10 10  3  1 10 10  4  1 10  1 10  4  1
[276]  6  1  1  1  7  1  1 10 10 10 10 10  1  5 10  1  1  5 10  3 10  5  7  1 10
[301]  4  1 10  1 10 10  1  1  3  5  1  1  1  1  1  2 10  8  1  5 10  1  1 10  1
[326]  1 10  1  4 10  8  1  1 10 10  1 10  1  1 10 10  1  1  1 10  1  1  1  1  8
[351]  1  1  3 10  1  1  3 10  4  7 10 10  3  3  1  1 10 10  1  1  1  1  1  1  1
[376]  1  1  1  1  1  1 10  1  1  1  1 10  1  1  2  1 10  1  1  1  1  1  1  1  1
[401]  9  1  1  4  1  1  1  1  2  1  1  0  4  1 10  3 10  1  2  1  3 10  1  1  1
[426] 10  1  2  1  1  1  1  1  1  8 10  1  1  1  1 10  4  3  2  1  1  1  1  1 10
[451]  1  1  1 10  1  6 10  3  1  1  1  5  1  1  1  4 10 10  1  1  1  1  1  1  1
[476]  1  1  1  1 10  1  1  5 10  1  3  1 10  3  4  1 10  1 10  5  1  1  1  1  1
[501]  1  1  1  1  1  1  5  4  1  1  1  1  1  1 10 10  1  1  1 10  1  1  5 10  1
[526]  1  1  1  1  1 10  1  1  1  1  1  1  1  1  1  2  1  1  1  1  1 10  1  1  5
[551]  1  1  1  5  1  1  1  1  1  1  1  1  1  1  1 10  1  3 10  5 10 10  1  1  2
[576]  1  1  1  1  1  1 10 10  1  1  1 10  1  3  1  1 10 10  1 10  1  1  1  1  1
[601]  1  1  1  1 10  8  1  1 10  1 10  2 10  1  1  1  1  2  1  1  1  2  1  1  1
[626]  4  6  5  1  1  1  1  1  3  1  1  1  2  1  1  1  1  1  1  1  1  1  1  2  1
[651]  4  1  1  1  1  1  1  1 10  1  1  1  1  1  1  1  1  1  1  5  8  1  1  1  1
[676]  1  1  1  1  1 10 10  1  1  1  1  1  1  1  1  1  5  1  1  2  1  3  4  5
>
```

Each of the methods has value in replacing the missing data. Mode/mean imputation is the easiest in terms of basic replacement but, may risk skewing values in smaller datasets.

15.1 Optimization

A good model for optimization would be determining the maximum number of classrooms needed to split students in a high school into groups of no more than 15 for standardized testing. Note that the normal class size is 28 and all classrooms are normally utilized. Variables that should be used are the number of students, the length of time allowed for testing, the number of classrooms currently available and the days that testing could be administered. The objective

function is to determine if all students can be tested on the same day or if testing needs to be split among multiple days to accommodate all students.