Homework10

Pablo

3/18/2019

Contents

14.1

1. Use the mean/mode imputation method to impute values for the missing data.
2. Use regression to impute values for the missing data.
3. Use regression with perturbation to impute values for the missing data

Mean/Mode

```
set.seed(42)

breast.cancer.wisconsin.data <- read.csv("~/Documents/R/GeorgiaTech/MissingData/breast-cancer-wisconsin

# search for missing data
#View(breast.cancer.wisconsin.data)
# column v7 seems to have ? string

# find how much data is missing
missing <- nrow(breast.cancer.wisconsin.data[which(breast.cancer.wisconsin.data$V7 == "?"),])/nrow(breas
missing*100
```

```
## [1] 2.288984
```

```
# less than 5% (ok)

# since the variable V7 of the data set is in range 1-10,
# it will be better to use the mode rather than the mean.

missing_data <- which(breast.cancer.wisconsin.data$V7=="?", arr.ind = TRUE)

mode <- function(x) {
  ux <- unique(x)
  return (as.numeric(ux[which.max(tabulate(match(x, ux)))]))
}
# remove missing data
```

```
mode_data <-mode(breast.cancer.wisconsin.data[-missing_data,"V7"])

#Now we will use this function to create a dummy variable
#that will indicate missing value using 0, otherwise willtake the value 1.
addDummuy <- function(t)
{
  x <- dim(length(t))
  x[which(t != "?")] = 1
  x[which(t == "?")] = 0
  return(x)
}

# do imputation, put mode in the data set.
new_data <- breast.cancer.wisconsin.data
new_data$dummy <- addDummuy(new_data$V7)
for(i in 1:nrow(new_data))
{
  if(new_data$dummy[i] == 0)
  {
    new_data$V7[i] <- mode_data

  }
}
# validate after adding new data (should be 0)
missing <- nrow(new_data[which(new_data$V7 == "?"),])/nrow(new_data)
missing
```

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

Regression

```
# response is in first column
data <- breast.cancer.wisconsin.data
data_lm <- (breast.cancer.wisconsin.data[-missing_data,2:10])
data_lm$V7 <- as.integer(data_lm$V7)
model <- lm(V7~., data = data_lm)
summary(model)
```

```
##
## Call:
## lm(formula = V7 ~ ., data = data_lm)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1137 -0.7185 -0.4731 -0.2994  7.3848
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.862817   0.162497  11.464  < 2e-16 ***
## V2           0.068118   0.034746   1.960  0.05035 .
## V3           0.087939   0.063482   1.385  0.16643
## V4           0.110046   0.061190   1.798  0.07255 .
## V5          -0.076950   0.038270  -2.011  0.04475 *
```

```
## V6           0.043216   0.052123   0.829  0.40733
## V8           0.044536   0.049211   0.905  0.36579
## V9           0.119422   0.037076   3.221  0.00134 **
## V10          0.001405   0.049448   0.028  0.97733
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.896 on 674 degrees of freedom
## Multiple R-squared:  0.2326, Adjusted R-squared:  0.2235
## F-statistic: 25.54 on 8 and 674 DF,  p-value: < 2.2e-16
```

```
# discard predictor by p - values > 0.1
model2 <- lm(V7 ~ V2+V4+V5+V9, data = data_lm)
summary(model2)
```

```
##
## Call:
## lm(formula = V7 ~ V2 + V4 + V5 + V9, data = data_lm)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9752 -0.7586 -0.4826 -0.3230  7.6159
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.95020    0.13693  14.242  < 2e-16 ***
## V2           0.07980    0.03432   2.325   0.0204 *
## V4           0.20051    0.04284   4.680 3.46e-06 ***
## V5          -0.04953    0.03575  -1.385   0.1664
## V9           0.14200    0.03525   4.028 6.25e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.897 on 678 degrees of freedom
## Multiple R-squared:  0.2268, Adjusted R-squared:  0.2222
## F-statistic: 49.71 on 4 and 678 DF,  p-value: < 2.2e-16
```

```
# now that the model is built
# use it to add the missing values to the data set.



#Now we will use this function to create a dummy variable
#that will indicate missing value using 0, otherwise willtake the value 1.

data$dummy <- addDummuy(data$V7)

head(data)
```

```
##        V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 dummy
## 1 1000025  5  1  1  1  2  1  3  1   1   2     1
## 2 1002945  5  4  4  5  7 10  3  2   1   2     1
## 3 1015425  3  1  1  1  2  2  3  1   1   2     1
## 4 1016277  6  8  8  1  3  4  3  7   1   2     1
## 5 1017023  4  1  1  3  2  1  3  1   1   2     1
## 6 1017122  8 10 10  8  7 10  9  7   1   4     1
```

```
new_points <- c()
for(i in 1:nrow(data))
{
  if(data$dummy[i] == 0)
  {
    # predict each value and save it as integer (no floats) in the missing i value
    data_point = as.integer(predict(model2, newdata = data[i,2:9]))
    data$V7[i] <- data_point
    new_points <- c(new_points, data_point)

  }
}
missing <- nrow(data[which(data$V7 == "?"),])/nrow(data)

# check that there is no missing data
missing
```

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

Regression with perturbation

```
# will generate random numbers following normal distribution rnorm(n,mean,sd)
random_distribution <- rnorm(nrow(new_data[missing_data,]),mean(new_points),sd(new_points))
# this should have 16 data points
random_distribution
```

```
##  [1] 3.937593 2.084343 2.972669 3.230920 3.012057 2.523394 4.072172
##  [8] 2.534371 4.557494 2.564956 3.874318 4.814296 1.295267 2.358080
## [15] 2.497355 3.233876
```

```
# before adding data points
missing <- nrow(new_data[which(new_data$V7 == "?"),])/nrow(new_data)
missing
```

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

```
# now add data points
j <- 1
for (i in 1:nrow(data)) {
  if(data$dummy[i] == 0) {
    # add values from random distribution
    new_point <- as.integer(random_distribution[j])
    # validate range and force only valid values for this categorical variable
    if (new_point < 1) new_point <- 1
    if (new_point > 10) new_point <- 10
    new_data$V7[i] <- new_point
    j = j+1
  }
}

# after adding data points
missing <- nrow(new_data[which(new_data$V7 == "?"),])/nrow(new_data)
missing
```

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

Output

Since I'm not doing the optional part (will start studing for midterm2) there is no much analysis I can make, but if I had made it there could be a comparison of quality of models between the methods of imputation. It is a long procedure because each model has to be imputated with the different methods, so I will look forward to the peer review.

What I can comment on the imputation methods is that for the normal random distribution there can be values that are outside the allowed range. In this case the range for the V7 column is 1-10 (categorical) so in the last method I had to check the range before adding the new calculated data point.

15.1

For the soccer world cup, an optimization model could be made to find how many stadiums are needed to host all the games. The variables that can be used is the number of games per day, the number of teams playing, the number of available cities, the expected number of attendees per game, etc. The objective function is to minimze the number of needed stadiums and some possible constraints is more than (>0) stadiums are needed, less than (<1) per city and less than 100k attendees per game.