Week 10 Homework

2022-11-02

Question 14.1 (a)

1. Use the mean/mode imputation method to impute values for the missing data.

```
summary(bc)
```

```
##        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
```

```
##--------------------------------------------------------------------
# impute by mean method since teh data is a numeric variable
bc_mean <- bc
bc_mean$V7 <- as.integer(bc_mean$V7)
```

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

```
str(bc_mean)
```

```
## ’data.frame’:    699 obs. of  11 variables:
##  $ V1 : int  1000025 1002945 1015425 1016277 1017023 1017122 1018099 1018561 1033078 1033078 ...
##  $ V2 : int  5 5 3 6 4 8 1 2 2 4 ...
##  $ V3 : int  1 4 1 8 1 10 1 1 1 2 ...
##  $ V4 : int  1 4 1 8 1 10 1 2 1 1 ...
##  $ V5 : int  1 5 1 1 3 8 1 1 1 1 ...
##  $ V6 : int  2 7 2 3 2 7 2 2 2 2 ...
##  $ V7 : int  1 10 2 4 1 10 10 1 1 1 ...
##  $ V8 : int  3 3 3 3 3 9 3 3 1 2 ...
##  $ V9 : int  1 2 1 7 1 7 1 1 1 1 ...
##  $ V10: int  1 1 1 1 1 1 1 1 5 1 ...
##  $ V11: int  2 2 2 2 2 4 2 2 2 2 ...
```

Question 14.1 (b)

2. Use regression to impute values for the missing data.

```
##----------------------------------------------------------------------
# Imputation using linear regression using  the simputation package
# making a copy of source table
bc_1 <- bc
# converting column V7 to Integer which converts "?" to NA
bc_1$V7 <- as.integer(bc_1$V7)
```

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

```
# using  the linear regression impuatation model in vanilla form
# creates a linear imputation without perturbation
# use all the variables but V11 against V7 for imputation regression
bc_imp <- impute_lm(bc_1, V7~V1+V2+V3+V4+V5+V6+V8+V9+V10)
# using the pipe method to get a column sum of all NA values after running model
bc_imp |> is.na() |> colSums()
```

```
##  V1  V2  V3  V4  V5  V6  V7  V8  V9 V10 V11
##   0   0   0   0   0   0   0   0   0   0   0
```

```
# rounding the V7 column to make it match other columns as a 1-10 numeric
bc_imp$V7 <- round(bc_imp$V7)
str(bc_imp)
```

```
## ’data.frame’:    699 obs. of  11 variables:
##  $ V1 : int  1000025 1002945 1015425 1016277 1017023 1017122 1018099 1018561 1033078 1033078 ...
##  $ V2 : int  5 5 3 6 4 8 1 2 2 4 ...
##  $ V3 : int  1 4 1 8 1 10 1 1 1 2 ...
##  $ V4 : int  1 4 1 8 1 10 1 2 1 1 ...
##  $ V5 : int  1 5 1 1 3 8 1 1 1 1 ...
##  $ V6 : int  2 7 2 3 2 7 2 2 2 2 ...
##  $ V7 : num  1 10 2 4 1 10 10 1 1 1 ...
##  $ V8 : int  3 3 3 3 3 9 3 3 1 2 ...
##  $ V9 : int  1 2 1 7 1 7 1 1 1 1 ...
##  $ V10: int  1 1 1 1 1 1 1 1 5 1 ...
##  $ V11: int  2 2 2 2 2 4 2 2 2 2 ...
```

Question 14.1(c)

3. Use regression with perturbation to impute values for the missing data. Using the simputation package we are able to add a “add_residual = ‘normal’ ” which uses perturbation via normal distrubtion which uses teh mean and sd to add randomness to the imputation

```
##-----------------------------------------------------------------------
# Imputation using linear regression using  the simputation package
# making a copy of source table
bc_2 <- bc
# converting column V7 to Integer which converts "?" to NA
bc_2$V7 <- as.integer(bc_2$V7)
```

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

```
# using  the linear regression impuatation model in vanilla form
# creates a linear imputation without perturbation
# use all the variables but V11 against V7 for imputation regression
bc_imp_1 <- impute_lm(bc_2, V7~V1+V2+V3+V4+V5+V6+V8+V9+V10, add_residual = "normal")
# using the pipe method to get a column sum of all NA values after running model
bc_imp_1 |> is.na() |> colSums()
```

```
##  V1  V2  V3  V4  V5  V6  V7  V8  V9 V10 V11
##   0   0   0   0   0   0   0   0   0   0   0
```

```
# rounding the V7 column to make it match other columns as a 1-10 numeric
bc_imp_1$V7 <- round(bc_imp_1$V7)
str(bc_imp_1)
```

```
## ’data.frame’:    699 obs. of  11 variables:
##  $ V1 : int  1000025 1002945 1015425 1016277 1017023 1017122 1018099 1018561 1033078 1033078 ...
##  $ V2 : int  5 5 3 6 4 8 1 2 2 4 ...
##  $ V3 : int  1 4 1 8 1 10 1 1 1 2 ...
##  $ V4 : int  1 4 1 8 1 10 1 2 1 1 ...
##  $ V5 : int  1 5 1 1 3 8 1 1 1 1 ...
##  $ V6 : int  2 7 2 3 2 7 2 2 2 2 ...
##  $ V7 : num  1 10 2 4 1 10 10 1 1 1 ...
##  $ V8 : int  3 3 3 3 3 9 3 3 1 2 ...
##  $ V9 : int  1 2 1 7 1 7 1 1 1 1 ...
##  $ V10: int  1 1 1 1 1 1 1 1 5 1 ...
##  $ V11: int  2 2 2 2 2 4 2 2 2 2 ...
```

Question 15.1

Describe a situation or problem from your job, everyday life, current events, etc., for which optimization would be appropriate. What data would you need?

I have a hobby, beekeeping. I work to ensure that I get the most honey production with the least amount of disturbance to the bees. I check them generally every two weeks and ensure they do not have veroa infestion (a mite that can kill the hive), ensure that the queen is still laying. Using optimization I could collect data to ensure that my bi-weekly check on the bees is not affecting the honey production looking at the last few years worth of amount of honey harvested and teh weather conditions of each year. Combine that data to see if i can predict the amount of honey I should get this year and collect information this year on the size of the hive by weight to see teh amount of bess/honey that the grows weekly in a month and change my hive inspections to 3 weeks or 2.5 weeks etc month over month and observe changes in the size of teh hive to get the determine the best schedules to check the bees as well as be able tp predict based on weight the amount of honey I might collect each year.