Homework 6

Question 14.1 The breast cancer data set breast-cancer-wisconsin.data.txt from http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/ (description at http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29) has missing values. 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. 4. (Optional) Compare the results and quality of classification models (e.g., SVM, KNN) build using (1) the data sets from questions 1,2,3; (2) the data that remains after data points with missing values are removed; and (3) the data set when a binary variable is introduced to indicate missing values.

```r
#Clear all objects from the current workspace
rm(list = ls())

#Import Dataset
filename= "~/Desktop/MicroMaster GTX/Week 6/breast-cancer-wisconsin.data 2.txt"
cancer_data <- read.table(filename, stringsAsFactors = FALSE, header=FALSE, sep = ",")

#Execute head and tail function to ensure data is read accurately
head(cancer_data)
```

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

```r
tail(cancer_data)
```

```
##         V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11
## 694 763235  3  1  1  1  2  1  2  1   2   2
## 695 776715  3  1  1  1  3  2  1  1   1   2
## 696 841769  2  1  1  1  2  1  1  1   1   2
## 697 888820  5 10 10  3  7  3  8 10   2   4
## 698 897471  4  8  6  4  3  4 10  6   1   4
## 699 897471  4  8  8  5  4  5 10  4   1   4
```

```r
#Check missing values in each column. Look for "?"
colSums(cancer_data == '?')
```

```
##  V1  V2  V3  V4  V5  V6  V7  V8  V9 V10 V11
##   0   0   0   0   0   0  16   0   0   0   0
```

```r
#V7 has missing values
cancer_data[cancer_data$V7 == '?',]
```

```
##          V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11
## 24  1057013  8  4  5  1  2  ?  7  3   1   4
## 41  1096800  6  6  6  9  6  ?  7  8   1   2
## 140 1183246  1  1  1  1  1  ?  2  1   1   2
## 146 1184840  1  1  3  1  2  ?  2  1   1   2
## 159 1193683  1  1  2  1  3  ?  1  1   1   2
## 165 1197510  5  1  1  1  2  ?  3  1   1   2
## 236 1241232  3  1  4  1  2  ?  3  1   1   2
## 250  169356  3  1  1  1  2  ?  3  1   1   2
## 276  432809  3  1  3  1  2  ?  2  1   1   2
## 293  563649  8  8  8  1  2  ?  6 10   1   4
## 295  606140  1  1  1  1  2  ?  2  1   1   2
## 298   61634  5  4  3  1  2  ?  2  3   1   2
## 316  704168  4  6  5  6  7  ?  4  9   1   2
## 322  733639  3  1  1  1  2  ?  3  1   1   2
## 412 1238464  1  1  1  1  1  ?  2  1   1   2
## 618 1057067  1  1  1  1  1  ?  1  1   1   2
```

```
num_missing = 100*nrow(cancer_data[cancer_data$V7 == '?',])/nrow(cancer_data)
num_missing
```

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

```
#We have 16 missing values which is less than 5%. Therefore, there is no bias

mod_function = function(x) {
  ux = unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}

#Get the indices for missing data
missing_indices = which(cancer_data$V7 == '?', arr.ind = T)

#Find the mode  for V7 using data thats not missing
mode_V7 = as.numeric(mod_function(cancer_data[-missing_indices, 'V7']))
mode_V7
```

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

```
#Imputation using mode
cancerdata_impute_mode = cancer_data
cancerdata_impute_mode[missing_indices, 'V7'] = mode_V7

#Check for missing values to ensure no missing values
colSums(cancerdata_impute_mode == '?')
```

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

```
#Find the mean  for V7 using data thats not missing
mean_V7 = mean(as.integer(cancer_data[-missing_indices, 'V7']))
mean_V7
```

```
## [1] 3.544656
```

```
#Imputation using mean
cancerdata_impute_mean = cancer_data
cancerdata_impute_mean[missing_indices, 'V7'] = as.integer(mean_V7)

#Check for missing values to ensure no missing values
colSums(cancerdata_impute_mean == '?')
```

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

```
#Imputation using regression
cancerdata_regress = cancer_data[-missing_indices, 2:10]
head(cancerdata_regress)
```

```
##   V2 V3 V4 V5 V6 V7 V8 V9 V10
## 1  5  1  1  1  2  1  3  1   1
## 2  5  4  4  5  7 10  3  2   1
## 3  3  1  1  1  2  2  3  1   1
## 4  6  8  8  1  3  4  3  7   1
## 5  4  1  1  3  2  1  3  1   1
## 6  8 10 10  8  7 10  9  7   1
```

```
model = lm (V7 ~., data = cancerdata_regress)
summary(model)
```

```
##
## Call:
## lm(formula = V7 ~ ., data = cancerdata_regress)
##
## 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
```

```
#Not all variables are significant. Therefore, i need to select significant features

step(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 = cancerdata_regress)
## 
## Coefficients:
## (Intercept)           V2           V4           V5           V8  
##     -0.5360       0.2262       0.3173       0.3323       0.3238
```

```r
model1 = lm(V7 ~ V2+V4+V5+V8, data = cancerdata_regress)
summary(model1)
```

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

```r
#Perform cross validation to check the performance of model1
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
cancerdata_regress$V7 = as.integer(cancerdata_regress$V7)
train.control = trainControl(method = 'repeatedcv' , repeats = 5, number = 5)
model_cv = train(V7 ~ V2+V4+V5+V8, data = cancerdata_regress, method = 'lm', trControl =train.control)
print(model_cv)
```

```
## Linear Regression
##
## 683 samples
##   4 predictor
##
## No pre-processing
## Resampling: Cross-Validated (5 fold, repeated 5 times)
## Summary of sample sizes: 546, 547, 547, 546, 546, 546, ...
## Resampling results:
##
##   RMSE      Rsquared  MAE
##   2.279113  0.611138  1.530568
##
## Tuning parameter 'intercept' was held constant at a value of TRUE
```

```
V7_regress_impute = predict(model1, cancer_data[missing_indices,])

#Check for missing values to ensure no missing values (should be 1-10)
V7_regress_impute
```

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

```
#Imputation using mean
cancerdata_impute_regress = cancer_data
cancerdata_impute_regress[missing_indices, 'V7'] = V7_regress_impute
cancerdata_impute_regress$V7 = as.integer(cancerdata_impute_regress$V7)

#Convert values less than 1 and more than 10 to 1 and 10 respectively
cancerdata_impute_regress$V7[cancerdata_impute_regress$V7>10] = 10
cancerdata_impute_regress$V7[cancerdata_impute_regress$V7<10] = 1

#Imputation using regression pertubation
set.seed(35)
cancerdata_impute_pert = cancer_data
cancerdata_impute_pert[missing_indices, 'V7'] = rnorm(length(V7_regress_impute),
                                                V7_regress_impute, sd(V7_regress_impute))
cancerdata_impute_pert$V7 = as.integer(cancerdata_impute_pert$V7)

#Convert values less than 1 and more than 10 to 1 and 10 respectively
cancerdata_impute_pert$V7[cancerdata_impute_pert$V7>10] = 10
cancerdata_impute_pert$V7[cancerdata_impute_pert$V7<10] = 1
```

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?

Answer: I recently used to work for as a supervisor for a fast food chain and optimization could be approriate to solve scheduling for each week

Data needed: n = number of people who start working on day of each week constrainst= workers work 5 days in a row followed by 2 days off min= worker-days used