Homework_10

Claire Kraft

2024-10-27

Question 14.1

The breast cancer data set breast-cancer-wisconsin.data.txt from UCI Repository (http://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/) (description at UCI Dataset Description (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) built using**
   (1) the data sets from questions 1, 2, 3;
   (2) the data that remains after data points with missing values are removed;
   (3) the data set when a binary variable is introduced to indicate missing values.

Went to the description at UCI Dataset Description (http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29) URL to understand the schema of the table. Column 11 is “Class” and the datatype is binary. If the value is 2, it indicates benign, and if the value is 4, it indicates malignant. According to the aggregation results, there are 241 malignant and 458 benign cases. The dataset contains a total of 691 missing data points. Column 7, is later renamed as “BareNuclei”, has 2.29% (with N=16) of its data missing. There is a strong correlation between cell size and shape.

After performing exploratory data analysis, I imputed the missing values using mean, mode, regression, and regression with perturbation methods.

- For the *mean* imputation, I calculated the mean of the `BareNuclei` column, ignoring the null values, and then replaced the null values with the computations. I repeated the same steps for *mode* imputation.
- For the *regression* imputation, I used the observed values of `BareNuclei` as the dependent variable. The remaining columns became the independent variables. This is a more advanced method, compared to just mean and mode imputations, as it considers the relationship of the dependent and independent variables to impute the missing values [6].
- A drawback of regression imputation is that it does not capture all the variability. *Perturbation* adds variability [2]. The beginning steps of this method are similar to the regression imputation: build the model and predict the missing values. Then I use the `rnorm` function to add “noise” to mimic natural variation. Finally, I replace the missing values with the predictions and noise.

```
# Library imports
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.4     ✓ readr     2.1.5
## ✓ forcats   1.0.0     ✓ stringr   1.5.1
## ✓ ggplot2   3.5.1     ✓ tibble    3.2.1
## ✓ lubridate 1.9.3     ✓ tidyr     1.3.1
## ✓ purrr     1.0.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(data.table)
```

```
##
## Attaching package: 'data.table'
##
## The following objects are masked from 'package:lubridate':
##
##     hour, isoweek, mday, minute, month, quarter, second, wday, week,
##     yday, year
##
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
##
## The following object is masked from 'package:purrr':
##
##     transpose
```

```
#install.packages("visdat")
library(visdat)
library(ggplot2)
#install.packages("naniar")
library(naniar)
library(GGally) #ggplot2 extension for pairs matrix
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
#install.packages("psych")
library(psych)
```

```
##
## Attaching package: 'psych'
##
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
library(rpart)
#install.packages("rpart.plot")
library(rpart.plot)

# Read in data
df <- fread('https://archive.ics.uci.edu/ml/machine-learning-databases/breast-cancer-wisconsin/breast-cancer-wisconsin.dat
a', header=FALSE, stringsAsFactors = FALSE)
# Preview data
head(df)
```

```
##         V1    V2    V3    V4    V5    V6     V7    V8    V9   V10   V11
##      <int> <int> <int> <int> <int> <int> <char> <int> <int> <int> <int>
## 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
```

```
summary(df)
```

```
##        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
```

```
# Count observations per class
df %>%
  count(V11)
```

```
##      V11     n
##    <int> <int>
## 1:     2   458
## 2:     4   241
```

```
# Copy original df
df_cleaning <- df

# Rename columns
df_cleaning <- rename (df_cleaning,
                       SampleCode=V1,
                       ClumpThickness=V2,
                       CellSize=V3,
                       CellShape=V4,
                       Adhesion=V5,
                       SingleEpithelialSize=V6,
                       BareNuclei=V7,
                       BlandChromatin=V8,
                       NormalNucleoli=V9,
                       Mitoses=V10,
                       Diagnosis=V11)

# Convert BareNuclei or V7 column to integer
df_cleaning$BareNuclei <- as.integer(df_cleaning$BareNuclei)
```

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

```
# Convert Diagnosis or V11 to factor
df_cleaning$Diagnosis <- as.factor(df_cleaning$Diagnosis)

# Set the levels of the Diagnosis factor to 0 and 1. This effectively assigns new labels to the existing factor levels.
levels(df_cleaning$Diagnosis) <- c(0, 1)

summary(df_cleaning)
```

```
##    SampleCode       ClumpThickness      CellSize        CellShape
##  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
##
##     Adhesion      SingleEpithelialSize   BareNuclei     BlandChromatin
##  Min.   : 1.000   Min.   : 1.000       Min.   : 1.000   Min.   : 1.000
##  1st Qu.: 1.000   1st Qu.: 2.000       1st Qu.: 1.000   1st Qu.: 2.000
##  Median : 1.000   Median : 2.000       Median : 1.000   Median : 3.000
##  Mean   : 2.807   Mean   : 3.216       Mean   : 3.545   Mean   : 3.438
##  3rd Qu.: 4.000   3rd Qu.: 4.000       3rd Qu.: 6.000   3rd Qu.: 5.000
##  Max.   :10.000   Max.   :10.000       Max.   :10.000   Max.   :10.000
##                                        NA's   :16
##  NormalNucleoli      Mitoses       Diagnosis
##  Min.   : 1.000   Min.   : 1.000   0:458
##  1st Qu.: 1.000   1st Qu.: 1.000   1:241
##  Median : 1.000   Median : 1.000
##  Mean   : 2.867   Mean   : 1.589
##  3rd Qu.: 4.000   3rd Qu.: 1.000
##  Max.   :10.000   Max.   :10.000
##
```

```
head(df_cleaning)
```

```
##    SampleCode ClumpThickness CellSize CellShape Adhesion SingleEpithelialSize
##         <int>          <int>    <int>     <int>    <int>                <int>
## 1:    1000025              5        1         1        1                    2
## 2:    1002945              5        4         4        5                    7
## 3:    1015425              3        1         1        1                    2
## 4:    1016277              6        8         8        1                    3
## 5:    1017023              4        1         1        3                    2
## 6:    1017122              8       10        10        8                    7
##    BareNuclei BlandChromatin NormalNucleoli Mitoses Diagnosis
##         <int>          <int>          <int>   <int>    <fctr>
## 1:          1              3              1       1         0
## 2:         10              3              2       1         0
## 3:          2              3              1       1         0
## 4:          4              3              7       1         0
## 5:          1              3              1       1         0
## 6:         10              9              7       1         1
```

```
# Dupes check
n_distinct(df_cleaning)
```

```
## [1] 691
```

```
# Zoom in on the dupes
duplicates <- df_cleaning %>%
  filter(duplicated(.))

print(duplicates)
```

```
##    SampleCode ClumpThickness CellSize CellShape Adhesion SingleEpithelialSize
##         <int>          <int>    <int>     <int>    <int>                <int>
## 1:    1218860              1        1         1        1                    1
## 2:    1100524              6       10        10        2                    8
## 3:    1116116              9       10        10        1                   10
## 4:    1198641              3        1         1        1                    2
## 5:     320675              3        3         5        2                    3
## 6:     704097              1        1         1        1                    1
## 7:    1321942              5        1         1        1                    2
## 8:     466906              1        1         1        1                    2
##    BareNuclei BlandChromatin NormalNucleoli Mitoses Diagnosis
##         <int>          <int>          <int>   <int>    <fctr>
## 1:          1              3              1       1         0
## 2:         10              7              3       3         1
## 3:          8              3              3       1         1
## 4:          1              3              1       1         0
## 5:         10              7              1       1         1
## 6:          1              2              1       1         0
## 7:          1              3              1       1         0
## 8:          1              1              1       1         0
```

```
missing_col_summary <- colSums(is.na(df_cleaning))
print(missing_col_summary)
```

```
##           SampleCode       ClumpThickness             CellSize
##                    0                    0                    0
##            CellShape             Adhesion SingleEpithelialSize
##                    0                    0                    0
##           BareNuclei       BlandChromatin       NormalNucleoli
##                   16                    0                    0
##              Mitoses            Diagnosis
##                    0                    0
```

```
print(sprintf("Percent of missing observation = %0.3f", 16/nrow(df_cleaning)*100))
```

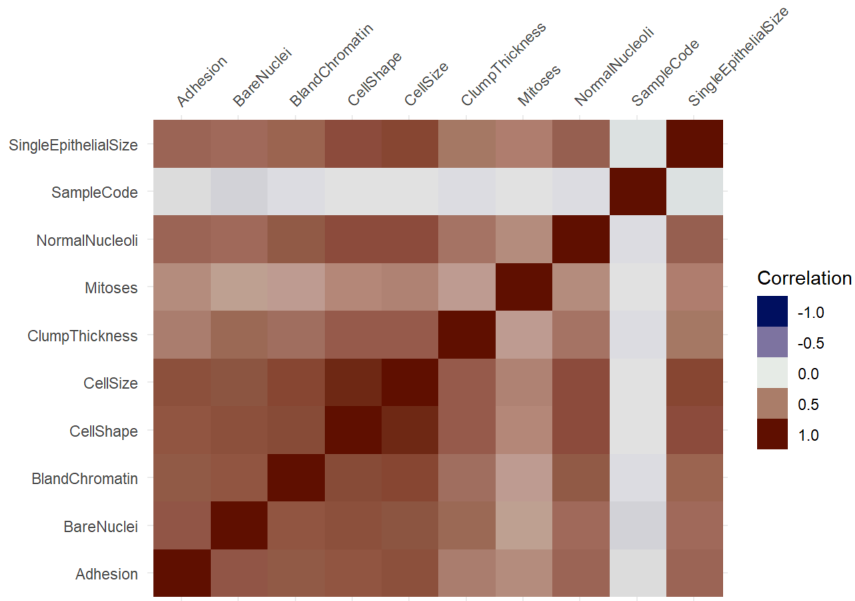
```
## [1] "Percent of missing observation = 2.289"
```

```
# Visualize missing values
vis_miss(df_cleaning)
```


```
# Look at reference 4
# Correlation viz
df_cleaning_x <- select_if(df_cleaning, is.numeric)
cor(df_cleaning_x)
```

```
##                      SampleCode ClumpThickness    CellSize   CellShape
## SampleCode            1.00000000    -0.05530844 -0.04160334 -0.04157607
## ClumpThickness       -0.05530844     1.00000000  0.64491250  0.65458908
## CellSize             -0.04160334     0.64491250  1.00000000  0.90688191
## CellShape            -0.04157607     0.65458908  0.90688191  1.00000000
## Adhesion             -0.06487808     0.48635624  0.70558181  0.68307920
## SingleEpithelialSize -0.04552828     0.52181622  0.75179913  0.71966844
## BareNuclei                    NA             NA          NA          NA
## BlandChromatin       -0.06005053     0.55842816  0.75572098  0.73594845
## NormalNucleoli       -0.05207195     0.53583455  0.72286482  0.71944632
## Mitoses              -0.03490066     0.35003386  0.45869315  0.43891093
##                         Adhesion SingleEpithelialSize BareNuclei BlandChromatin
## SampleCode           -0.06487808          -0.04552828         NA    -0.06005053
## ClumpThickness        0.48635624           0.52181622         NA     0.55842816
## CellSize              0.70558181           0.75179913         NA     0.75572098
## CellShape             0.68307920           0.71966844         NA     0.73594845
## Adhesion              1.00000000           0.59959907         NA     0.66671533
## SingleEpithelialSize  0.59959907           1.00000000         NA     0.61610184
## BareNuclei                    NA                   NA          1             NA
## BlandChromatin        0.66671533           0.61610184         NA     1.00000000
## NormalNucleoli        0.60335241           0.62888069         NA     0.66587781
## Mitoses               0.41763278           0.47910148         NA     0.34416950
##                      NormalNucleoli     Mitoses
## SampleCode              -0.05207195 -0.03490066
## ClumpThickness           0.53583455  0.35003386
## CellSize                 0.72286482  0.45869315
## CellShape                0.71944632  0.43891093
## Adhesion                 0.60335241  0.41763278
## SingleEpithelialSize     0.62888069  0.47910148
## BareNuclei                       NA          NA
## BlandChromatin           0.66587781  0.34416950
## NormalNucleoli           1.00000000  0.42833575
## Mitoses                  0.42833575  1.00000000
```

```
vis_cor(df_cleaning_x)
```




```
# Change Diagnosis from integer to factor
df_cleaning$Diagnosis <- as.factor(df_cleaning$Diagnosis)

pairwise <- ggpairs(df_cleaning, columns = 1:10, ggplot2::aes(colour = Diagnosis), lower=list(combo=wrap("facethist", binwid
th=0.5)))
pairwise
```

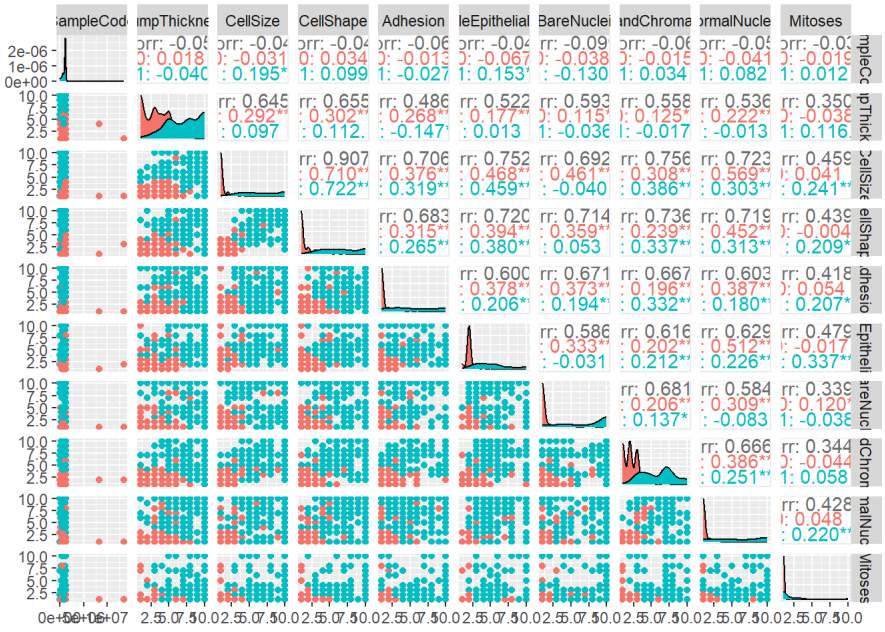
```
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 16 rows containing missing values
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 16 rows containing missing values
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 16 rows containing missing values
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 16 rows containing missing values
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 16 rows containing missing values
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 16 rows containing missing values
```

```
## Warning: Removed 16 rows containing missing values or values outside the scale range
## (`geom_point()`).
## Removed 16 rows containing missing values or values outside the scale range
## (`geom_point()`).
## Removed 16 rows containing missing values or values outside the scale range
## (`geom_point()`).
## Removed 16 rows containing missing values or values outside the scale range
## (`geom_point()`).
## Removed 16 rows containing missing values or values outside the scale range
## (`geom_point()`).
## Removed 16 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
## Warning: Removed 16 rows containing non-finite outside the scale range
## (`stat_density()`).
```

```
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 16 rows containing missing values
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 16 rows containing missing values
## Warning in ggally_statistic(data = data, mapping = mapping, na.rm = na.rm, :
## Removed 16 rows containing missing values
```

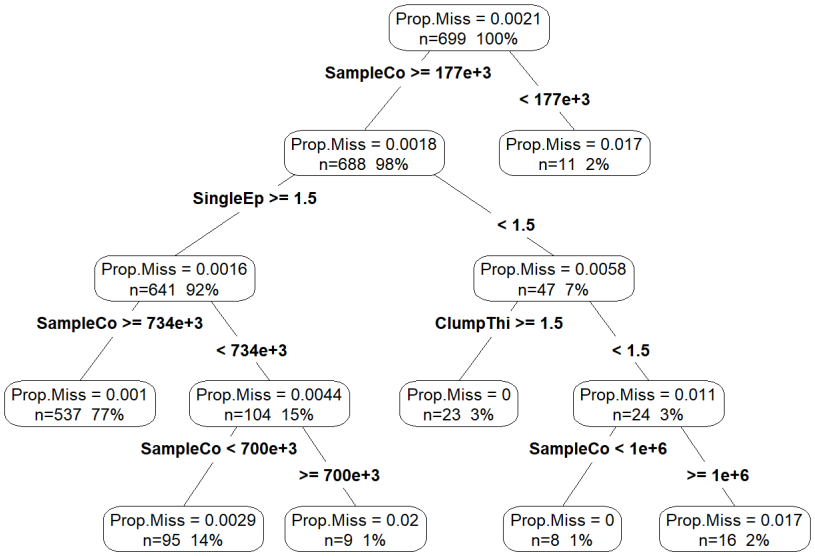
```
## Warning: Removed 16 rows containing missing values or values outside the scale range
## (`geom_point()`).
## Removed 16 rows containing missing values or values outside the scale range
## (`geom_point()`).
## Removed 16 rows containing missing values or values outside the scale range
## (`geom_point()`).
```




```
# Look at reference 4
skew(df_cleaning_x)
```

```
##  [1] 13.6166895  0.5903165  1.2278492  1.1568774  1.5179315  1.7048304
##  [7]  0.9856714  1.0952527  1.4161630  3.5453906
```

```
df_cleaning %>%
  add_prop_miss() %>%
  rpart(prop_miss_all ~ ., data=.) %>%
  prp(type=4, extra = 101, roundint=FALSE, prefix="Prop.Miss = ")
```



Imputation - mean/mode

```
#---------------------- Mean imputation
df_cleaning.mean<-df_cleaning
df_cleaning.mean<-df_cleaning.mean %>% mutate_at(vars(BareNuclei),~ifelse(is.na(.x), mean(.x, na.rm = TRUE), .x))

# Double check mean imputation
head(df_cleaning.mean,24)
```

```
##     SampleCode ClumpThickness CellSize CellShape Adhesion SingleEpithelialSize
##          <int>          <int>    <int>     <int>    <int>                <int>
##  1:    1000025              5        1         1        1                    2
##  2:    1002945              5        4         4        5                    7
##  3:    1015425              3        1         1        1                    2
##  4:    1016277              6        8         8        1                    3
##  5:    1017023              4        1         1        3                    2
##  6:    1017122              8       10        10        8                    7
##  7:    1018099              1        1         1        1                    2
##  8:    1018561              2        1         2        1                    2
##  9:    1033078              2        1         1        1                    2
## 10:    1033078              4        2         1        1                    2
## 11:    1035283              1        1         1        1                    1
## 12:    1036172              2        1         1        1                    2
## 13:    1041801              5        3         3        3                    2
## 14:    1043999              1        1         1        1                    2
## 15:    1044572              8        7         5       10                    7
## 16:    1047630              7        4         6        4                    6
## 17:    1048672              4        1         1        1                    2
## 18:    1049815              4        1         1        1                    2
## 19:    1050670             10        7         7        6                    4
## 20:    1050718              6        1         1        1                    2
## 21:    1054590              7        3         2       10                    5
## 22:    1054593             10        5         5        3                    6
## 23:    1056784              3        1         1        1                    2
## 24:    1057013              8        4         5        1                    2
##     SampleCode ClumpThickness CellSize CellShape Adhesion SingleEpithelialSize
##     BareNuclei BlandChromatin NormalNucleoli Mitoses Diagnosis
##          <num>          <int>          <int>   <int>    <fctr>
##  1:   1.000000              3              1       1         0
##  2:  10.000000              3              2       1         0
##  3:   2.000000              3              1       1         0
##  4:   4.000000              3              7       1         0
##  5:   1.000000              3              1       1         0
##  6:  10.000000              9              7       1         1
##  7:  10.000000              3              1       1         0
##  8:   1.000000              3              1       1         0
##  9:   1.000000              1              1       5         0
## 10:   1.000000              2              1       1         0
## 11:   1.000000              3              1       1         0
## 12:   1.000000              2              1       1         0
## 13:   3.000000              4              4       1         1
## 14:   3.000000              3              1       1         0
## 15:   9.000000              5              5       4         1
## 16:   1.000000              4              3       1         1
## 17:   1.000000              2              1       1         0
## 18:   1.000000              3              1       1         0
## 19:  10.000000              4              1       2         1
## 20:   1.000000              3              1       1         0
## 21:  10.000000              5              4       4         1
## 22:   7.000000              7             10       1         1
## 23:   1.000000              2              1       1         0
## 24:   3.544656              7              3       1         1
##     BareNuclei BlandChromatin NormalNucleoli Mitoses Diagnosis
```

```
#---------------------- Mode imputation
# Find mode - look at reference 3
getmode <- function(v) {
   uniqv <- unique(v)
   uniqv[which.max(tabulate(match(v, uniqv)))]
}

df_cleaning.mode<-df_cleaning
mode.result <- getmode(df_cleaning.mode$BareNuclei)
print(mode.result)
```

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

```
# Fill NAs with mode of 1s
df_cleaning.mode$BareNuclei[is.na(df_cleaning.mode$BareNuclei)] <- mode.result

# Double check mode imputation
head(df_cleaning.mode,24)
```

```
##     SampleCode ClumpThickness CellSize CellShape Adhesion SingleEpithelialSize
##          <int>          <int>    <int>     <int>    <int>                <int>
##  1:    1000025              5        1         1        1                    2
##  2:    1002945              5        4         4        5                    7
##  3:    1015425              3        1         1        1                    2
##  4:    1016277              6        8         8        1                    3
##  5:    1017023              4        1         1        3                    2
##  6:    1017122              8       10        10        8                    7
##  7:    1018099              1        1         1        1                    2
##  8:    1018561              2        1         2        1                    2
##  9:    1033078              2        1         1        1                    2
## 10:    1033078              4        2         1        1                    2
## 11:    1035283              1        1         1        1                    1
## 12:    1036172              2        1         1        1                    2
## 13:    1041801              5        3         3        3                    2
## 14:    1043999              1        1         1        1                    2
## 15:    1044572              8        7         5       10                    7
## 16:    1047630              7        4         6        4                    6
## 17:    1048672              4        1         1        1                    2
## 18:    1049815              4        1         1        1                    2
## 19:    1050670             10        7         7        6                    4
## 20:    1050718              6        1         1        1                    2
## 21:    1054590              7        3         2       10                    5
## 22:    1054593             10        5         5        3                    6
## 23:    1056784              3        1         1        1                    2
## 24:    1057013              8        4         5        1                    2
##     SampleCode ClumpThickness CellSize CellShape Adhesion SingleEpithelialSize
##     BareNuclei BlandChromatin NormalNucleoli Mitoses Diagnosis
##          <int>          <int>          <int>   <int>    <fctr>
##  1:          1              3              1       1         0
##  2:         10              3              2       1         0
##  3:          2              3              1       1         0
##  4:          4              3              7       1         0
##  5:          1              3              1       1         0
##  6:         10              9              7       1         1
##  7:         10              3              1       1         0
##  8:          1              3              1       1         0
##  9:          1              1              1       5         0
## 10:          1              2              1       1         0
## 11:          1              3              1       1         0
## 12:          1              2              1       1         0
## 13:          3              4              4       1         1
## 14:          3              3              1       1         0
## 15:          9              5              5       4         1
## 16:          1              4              3       1         1
## 17:          1              2              1       1         0
## 18:          1              3              1       1         0
## 19:         10              4              1       2         1
## 20:          1              3              1       1         0
## 21:         10              5              4       4         1
## 22:          7              7             10       1         1
## 23:          1              2              1       1         0
## 24:          1              7              3       1         1
##     BareNuclei BlandChromatin NormalNucleoli Mitoses Diagnosis
```

Imputation - regression

Explanation:

1. **Identify Missing Values**:
   - `missing.index <- which(is.na(reg_df$BareNuclei), arr.ind=TRUE)` : Finds the indices of missing values in the `BareNuclei` column.
2. **Remove Rows with Missing Values**:
   - `reg_df.1 <- reg_df[-missing.index, 2:10]` : Removes rows with missing `BareNuclei` values and considers only columns 2 to 10 for regression.
3. **Fit Regression Model**:
   - `reg_model <- lm(BareNuclei ~ ClumpThickness + CellSize + CellShape + Adhesion + SingleEpithelialSize + BlandChromatin + NormalNucleoli` Fits a linear regression model to predict `BareNuclei` based on other variables.
4. **Predict Missing Values**:

- `predicted_values <- predict(reg_model, reg_df[missing.index, 2:10])` : Uses the fitted model to predict missing `BareNuclei` values.

5. **Impute Predicted Values**:
   - `reg_df$BareNuclei[missing.index] <- predicted_values` : Replaces the missing values in the `BareNuclei` column with the predicted values.
6. **Check Imputation**:
   - `summary(reg_df)` : Provides a summary of the dataframe after imputation to verify the changes.

```
# For reproducibility
set.seed(123)

# Make a copy of the dataframe to preserve the original data
reg_df <- df_cleaning

# Identify the indices of missing values in the BareNuclei column
missing.index <- which(is.na(reg_df$BareNuclei), arr.ind=TRUE)

# Remove rows with missing BareNuclei values from the dataframe
reg_df.1 <- reg_df[-missing.index, 2:10]

# Fit a linear regression model to predict BareNuclei using other columns
reg_model <- lm(BareNuclei ~ ClumpThickness + CellSize + CellShape + Adhesion +
                SingleEpithelialSize + BlandChromatin + NormalNucleoli + Mitoses, data = reg_df.1)
summary(reg_model)
```

```
##
## Call:
## lm(formula = BareNuclei ~ ClumpThickness + CellSize + CellShape +
##     Adhesion + SingleEpithelialSize + BlandChromatin + NormalNucleoli +
##     Mitoses, data = reg_df.1)
##
## 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 **
## ClumpThickness        0.230156   0.041691   5.521 4.83e-08 ***
## CellSize             -0.067980   0.076170  -0.892  0.37246
## CellShape             0.340442   0.073420   4.637 4.25e-06 ***
## Adhesion              0.339705   0.045919   7.398 4.13e-13 ***
## SingleEpithelialSize  0.090392   0.062541   1.445  0.14883
## BlandChromatin        0.320577   0.059047   5.429 7.91e-08 ***
## NormalNucleoli        0.007293   0.044486   0.164  0.86983
## Mitoses              -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
```

```
# Use the fitted model to predict missing BareNuclei values
predicted_values <- predict(reg_model, reg_df[missing.index, 2:10])

# Impute the predicted values back into the original dataframe
reg_df$BareNuclei[missing.index] <- predicted_values

# Check the dataframe after imputation
summary(reg_df)
```

```
##    SampleCode       ClumpThickness      CellSize        CellShape
##  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
##     Adhesion      SingleEpithelialSize   BareNuclei      BlandChromatin
##  Min.   : 1.000   Min.   : 1.000       Min.   : 0.5687   Min.   : 1.000
##  1st Qu.: 1.000   1st Qu.: 2.000       1st Qu.: 1.0000   1st Qu.: 2.000
##  Median : 1.000   Median : 2.000       Median : 1.0000   Median : 3.000
##  Mean   : 2.807   Mean   : 3.216       Mean   : 3.5266   Mean   : 3.438
##  3rd Qu.: 4.000   3rd Qu.: 4.000       3rd Qu.: 5.9230   3rd Qu.: 5.000
##  Max.   :10.000   Max.   :10.000       Max.   :10.0000   Max.   :10.000
##  NormalNucleoli      Mitoses       Diagnosis
##  Min.   : 1.000   Min.   : 1.000   0:458
##  1st Qu.: 1.000   1st Qu.: 1.000   1:241
##  Median : 1.000   Median : 1.000
##  Mean   : 2.867   Mean   : 1.589
##  3rd Qu.: 4.000   3rd Qu.: 1.000
##  Max.   :10.000   Max.   :10.000
```

Imputation - perturbation

Explanation:

1. **Calculate Residuals and Noise Level**:
   - Get residuals from the regression model and calculate the standard deviation to set the noise level.
2. **Add Noise to Predictions**:
   - Use `rnorm` to add random noise to the predicted values.
3. **Impute Perturbed Values**:
   - Replace the missing values with these perturbed predictions.

```
# For reproducibility
set.seed(123)

# Make a copy of the dataframe to preserve the original data
perturbation_df <- df_cleaning

# Identify the indices of missing values in the BareNuclei column
missing.index <- which(is.na(perturbation_df$BareNuclei), arr.ind=TRUE)

# Remove rows with missing BareNuclei values from the dataframe
perturbation_df.1 <- perturbation_df[-missing.index, 2:10]

# Fit a linear regression model to predict BareNuclei using other columns
reg_model <- lm(BareNuclei ~ ClumpThickness + CellSize + CellShape + Adhesion +
                SingleEpithelialSize + BlandChromatin + NormalNucleoli + Mitoses, data = perturbation_df.1)
summary(reg_model)
```

```
##
## Call:
## lm(formula = BareNuclei ~ ClumpThickness + CellSize + CellShape +
##     Adhesion + SingleEpithelialSize + BlandChromatin + NormalNucleoli +
##     Mitoses, data = perturbation_df.1)
##
## 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 **
## ClumpThickness        0.230156   0.041691   5.521 4.83e-08 ***
## CellSize             -0.067980   0.076170  -0.892  0.37246
## CellShape             0.340442   0.073420   4.637 4.25e-06 ***
## Adhesion              0.339705   0.045919   7.398 4.13e-13 ***
## SingleEpithelialSize  0.090392   0.062541   1.445  0.14883
## BlandChromatin        0.320577   0.059047   5.429 7.91e-08 ***
## NormalNucleoli        0.007293   0.044486   0.164  0.86983
## Mitoses              -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
```

```
# Use the fitted model to predict missing BareNuclei values
predicted_values <- predict(reg_model, perturbation_df[missing.index, 2:10])

# Add some random noise (perturbation) to the predicted values
# Using standard deviation of residuals as noise level
residuals <- residuals(reg_model)
noise_level <- sd(residuals)
perturbed_values <- predicted_values + rnorm(length(predicted_values), mean = 0, sd = noise_level)

# Impute the perturbed values back into the original dataframe
perturbation_df$BareNuclei[missing.index] <- perturbed_values

# Check the dataframe after imputation
summary(perturbation_df)
```

```
##    SampleCode       ClumpThickness      CellSize        CellShape
##  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
##     Adhesion      SingleEpithelialSize   BareNuclei    BlandChromatin
##  Min.   : 1.000   Min.   : 1.000       Min.   :-1.10   Min.   : 1.000
##  1st Qu.: 1.000   1st Qu.: 2.000       1st Qu.: 1.00   1st Qu.: 2.000
##  Median : 1.000   Median : 2.000       Median : 1.00   Median : 3.000
##  Mean   : 2.807   Mean   : 3.216       Mean   : 3.54   Mean   : 3.438
##  3rd Qu.: 4.000   3rd Qu.: 4.000       3rd Qu.: 6.00   3rd Qu.: 5.000
##  Max.   :10.000   Max.   :10.000       Max.   :10.00   Max.   :10.000
##  NormalNucleoli      Mitoses       Diagnosis
##  Min.   : 1.000   Min.   : 1.000   0:458
##  1st Qu.: 1.000   1st Qu.: 1.000   1:241
##  Median : 1.000   Median : 1.000
##  Mean   : 2.867   Mean   : 1.589
##  3rd Qu.: 4.000   3rd Qu.: 1.000
##  Max.   :10.000   Max.   :10.000
```

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?

Optimization is used in descriptive and predictive analytics [1] and there are three parts to it - variables, constraints, and objective function [2]. In November I will be supporting my sister in the NYC Marathon. Applying these three parts to my NYC Marathon cheering strategy, the goal or *objective function* is to maximize the number of times I can cheer her without missing her. The key *variables* are the mile markers (where I can stand and cheer). *Constraints* include traffic, public transportation schedules, and crowd density, which can all affect my adherence to my cheering strategy.

References:

[1] Lecture 1 slides

[2] Lecture 3 slides

[3] R - Mean, Median and Mode - Tutorialspoint. (2019). Tutorialspoint.com. https://www.tutorialspoint.com/r/r_mean_median_mode.htm (https://www.tutorialspoint.com/r/r_mean_median_mode.htm)

[4] RPubs - Vaar R Notebook: Breast Cancer Wisconsin Original. (2023, August 12). Rpubs.com. https://rpubs.com/bi23le/1070975 (https://rpubs.com/bi23le/1070975)

[5] Microsoft. (2024). Copilot: AI companion. Accessed 2024-10-27. Prompt: ‘Add comments to my lines of code.’ Generated using https://www.microsoft.com (https://www.microsoft.com).”

[5] Microsoft. (2024). Copilot: AI companion. Accessed 2024-10-29. Prompt: ‘Here’s my code, read it, understand it, then write a concise explanation the steps in rmd format.’ Generated using https://www.microsoft.com (https://www.microsoft.com).”

[6] Learn how to fill in missing values for regression analysis using different imputation methods and R. Compare the advantages and disadvantages of each method. (2023, September 27). Linkedin.com. https://www.linkedin.com/advice/0/how-can-you-impute-missing-data-regression-analysis (https://www.linkedin.com/advice/0/how-can-you-impute-missing-data-regression-analysis)