Data Wrangling (Data Preprocessing)

Code

Practical assessment 1

Andre Gunawan - s3885488

Setup

Hide

```
#load the nessacary packages.
library(readr) # for importing data.
library(knitr) # for creating tables.
```

Provide a clear description of the data and its source (i.e. URL of the web site). Provide variable descriptions in this section.

ANSWER:

- The data was taken from the Kaggle website. URL: https://www.kaggle.com/prathamtripathi/drug-classification (https://www.kaggle.com/prathamtripathi/drug-classification)
- Data description: The data is about drug classification. It contains various information about age, sex, blood pressure, Cholestrol levels, NA to Potassium ratio related to drug type.
- Variable descriptions:
    - Age: integer variable representing the age of the patient
    - Sex: factor variable representing Gender of the patients
    - BP: factor variable representing blood pressure levels
    - Cholesterol: factor variable representing cholesterol levels
    - NA_to_K: numeric (double) varaible representing sodium to potassium ration in blood
    - Drug: factor variable representing the drug type

Read/Import Data

Hide

```
#read/import the data using read_csv from readr package. Then assign it to a data frame named drug200.
drug200 <- read_csv("C:/Users/andre/OneDrive/Desktop/Data Wrangling/Practical Assessment 1/Assessment 1/drug200.csv")
```

```
-- Column specification --------------------------------------------------------
-----------------------------------------------
cols(
  Age = col_double(),
  Sex = col_character(),
  BP = col_character(),
  Cholesterol = col_character(),
  Na_to_K = col_double(),
  Drug = col_character()
)
```

Hide

```
#Use head(drug200) function to show the head of data set.
head(drug200)
```

Age <dbl>	Sex <chr>	BP <chr>	Cholesterol <chr>	Na_to_K <dbl>	Drug <chr>
23	F	HIGH	HIGH	25.355	DrugY
47	M	LOW	HIGH	13.093	drugC
47	M	LOW	HIGH	10.114	drugC
28	F	NORMAL	HIGH	7.798	drugX
61	F	LOW	HIGH	18.043	DrugY
22	F	NORMAL	HIGH	8.607	drugX

6 rows

Step explaination:

- The first step I did was save the csv file in my working directory and use read_csv() to read/import and store it in the drug200 object in R as data frame.
- Then I used head(drug200) function to show the head of data set.
- Data description: The data is about drug classification. It contains various information about age, sex, blood pressure, Cholestrol levels, NA to Potassium ratio related to drug type.
- There are six variables in this dataset with different data types:
    - Age: integer variable representing the age of the patient
    - Sex: factor variable representing Gender of the patients
    - BP: factor variable representing blood pressure levels
    - Cholesterol: factor variable representing cholesterol levels
    - NA_to_K: numeric (double) variable representing sodium to potassium ration in blood
    - Drug: factor variable representing the drug type

Inspect and Understand

Hide

```
#Applying proper type conversion from double to integer and from character to factor.
drug200$Age <- as.integer(drug200$Age)
drug200$Sex <- as.factor(drug200$Sex)
drug200$BP <- as.factor(drug200$BP)
drug200$Cholesterol <- as.factor(drug200$Cholesterol)
drug200$Drug <- as.factor(drug200$Drug)

#renamed and reordered the levels of factor variables.
drug200$Sex <- factor(drug200$Sex, levels = c("M", "F"), labels = c("Male", "Female"))
drug200$BP <- factor(drug200$BP, levels = c("LOW", "NORMAL", "HIGH"), ordered = TRUE)
drug200$Cholesterol <- factor(drug200$Cholesterol, levels = c("NORMAL", "HIGH"), ordered = TR
UE)

#Check the levels of factor variables.
levels(drug200$Sex)
```

```
[1] "Male"   "Female"
```

Hide

```
levels(drug200$BP)
```

```
[1] "LOW"    "NORMAL" "HIGH"
```

Hide

```
levels(drug200$Cholesterol)
```

```
[1] "NORMAL" "HIGH"
```

Hide

```
levels(drug200$Drug)
```

```
[1] "drugA" "drugB" "drugC" "drugX" "DrugY"
```

Hide

```
#check the dimensions, column names, and data types.
str(drug200)
```

```
spec_tbl_df [200 x 6] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ Age        : int [1:200] 23 47 47 28 61 22 49 41 60 43 ...
 $ Sex        : Factor w/ 2 levels "Male","Female": 2 1 1 2 2 2 2 1 1 1 ...
 $ BP         : Ord.factor w/ 3 levels "LOW"<"NORMAL"<..: 3 1 1 2 1 2 2 1 2 1 ...
 $ Cholesterol: Ord.factor w/ 2 levels "NORMAL"<"HIGH": 2 2 2 2 2 2 2 2 2 1 ...
 $ Na_to_K    : num [1:200] 25.4 13.1 10.1 7.8 18 ...
 $ Drug       : Factor w/ 5 levels "drugA","drugB",..: 5 3 3 4 5 4 5 3 5 5 ...
 - attr(*, "spec")=
  .. cols(
  ..   Age = col_double(),
  ..   Sex = col_character(),
  ..   BP = col_character(),
  ..   Cholesterol = col_character(),
  ..   Na_to_K = col_double(),
  ..   Drug = col_character()
  .. )
```

Hide

```
#check the attributes.
attributes(drug200)
```

```
$names
[1] "Age"         "Sex"         "BP"          "Cholesterol" "Na_to_K"     "Drug"

$row.names
  [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20  21  22
23  24  25  26  27  28  29  30  31  32  33
 [34]  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55
56  57  58  59  60  61  62  63  64  65  66
 [67]  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88
89  90  91  92  93  94  95  96  97  98  99
[100] 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121
122 123 124 125 126 127 128 129 130 131 132
[133] 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154
155 156 157 158 159 160 161 162 163 164 165
[166] 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187
188 189 190 191 192 193 194 195 196 197 198
[199] 199 200

$spec
cols(
  Age = col_double(),
  Sex = col_character(),
  BP = col_character(),
  Cholesterol = col_character(),
  Na_to_K = col_double(),
  Drug = col_character()
)

$class
[1] "spec_tbl_df" "tbl_df"      "tbl"         "data.frame"
```

Step explaination:

- The dimensions is 200 observations x 6 variables as can be seen in the str() function.

- Each column has a name that represents its data, renaming is not necessary.
- There are six variables in this dataset with different data types:
    - Age: integer variable
    - Sex: factor variable
    - BP: factor variable
    - Cholesterol: factor variable
    - NA_to_K: numeric (double) varaible
    - Drug: factor variable
- Character data types have been converted into factor using as.factor() function.
- Numeric data types (age) have been converted into an integer using as.integer() function.
- Have checked the levels of factor variables using levels() function, also renamed using labels() function and reordered the factor variable using the ordered = TRUE.
- Have checked the attributes using attributes() function.

Subsetting

Hide

```
#subset the data frame.
drug200matrix <- drug200[1:10, ]

#convert to a matrix.
drug200matrix <- as.matrix(drug200matrix)

#check the structure of that matrix.
str(drug200matrix)
```

```
 chr [1:10, 1:6] "23" "47" "47" "28" "61" "22" "49" "41" "60" "43" "Female" "Male" "Male" "Fe
male" "Female" "Female" "Female" "Male" ...
 - attr(*, "dimnames")=List of 2
  ..$ : NULL
  ..$ : chr [1:6] "Age" "Sex" "BP" "Cholesterol" ...
```

Step explaination:

- First, I subset the data frame using the first 10 observations and include all variables and assign it to drug200matrix
- Second, I convert the data frame into a matrix using as.matrix function
- Third, I am using the str() function to check the structure
- The structure ended up as a character because the R default will convert the data to the most flexible types(logical<integer<numeric<character). Matrix can only contain the same data types, in this case character have the first priority in the default, so the data is automaticly converted into a character.

Create a new Data Frame

Hide

```
#create integer variable and ordinal variable with 10 observations each.
student_number <- c(1L, 2L, 3L, 4L, 5L, 6L, 7L, 8L, 9L, 10L)
grade <- factor(c("HD","P", "P", "C", "D", "N", "D", "HD", "C", "D"), levels = c("N", "P",
"C", "D", "HD"), ordered = TRUE)

#checking the ordinal variable levels
levels(grade)
```

```
[1] "N"  "P"  "C"  "D"  "HD"
```

Hide

```
#Combine the vector to a data frame and assign it to variable student_grade.
student_grade <- data.frame(student_number, grade)

#check the structure of new data frame (student grade).
str(student_grade)
```

```
'data.frame':   10 obs. of  2 variables:
 $ student_number: int  1 2 3 4 5 6 7 8 9 10
 $ grade         : Ord.factor w/ 5 levels "N"<"P"<"C"<"D"<..: 5 2 2 3 4 1 4 5 3 4
```

Hide

```
#create new vector (numeric) and bind the column using cbind(). Then, assign it to variable n
amed student_overall.
score <- c(83.7, 52.3, 56.5, 63.2, 73.5, 38.2, 77.8, 88.8, 63.6, 72.5)
student_overall <- cbind(student_grade, score)

#check the structure of student overall.
str(student_overall)
```

```
'data.frame':   10 obs. of  3 variables:
 $ student_number: int  1 2 3 4 5 6 7 8 9 10
 $ grade         : Ord.factor w/ 5 levels "N"<"P"<"C"<"D"<..: 5 2 2 3 4 1 4 5 3 4
 $ score         : num  83.7 52.3 56.5 63.2 73.5 38.2 77.8 88.8 63.6 72.5
```

Step explaination:

- First, I created a vector with an integer using c() function. Then assign it to the student number variable.
- Second, I created an ordered factor using factor() function. Then assign it to the grade variable.
- Third, check the levels of the ordinal variable with levels().
- Fourth, I combined both student number and grade variable into a data frame named student_grade using data.frame() function.
- Fifth, check the structure of the data frame using str() function.
- Sixth, create a numeric vector named score and bind it with the student_grade variable using cbind(). Then, I assign it to a new variable named student_overall
- Lastly, I check the structure of the student_overall variable using str() function.