THE ULTIMATE GUIDE TO DATA MANIPULATION WITH R AND PYTHON

Zakaria Al Azhar, bigdatahabits.com

2017-11-12

Contents

Chapter 1

Klm

Chapter 2

Introduction

Chapter 3

Prerequisites

3.1 Basic knowledge

3.2 Dataset

3.3 Python and R packages

Chapter 4

Data Exploration

4.1 Data Structure

4.2 Data Summary

Chapter 5

Column Formulas

After obtaining a good overview of the data, we can move to the next step: manipulating data. In this chapter we present the most used data manipulaton formulas on one ore more columns.

5.1 Data Binning

Data Binning is about grouping data in intervals - called bins. For example, in the titanic dataset we've measured the age in years, but you wanted to have age categories as follows:

- 1 = Child , age ranges of 0-17
- 2 = Adult, age ranges of 18-39
- 3 = Middle Aged, age ranges of 40-59
- 4 = Over 60, age ranges of 60 and above

R

```
titanic = read.csv("titanic.csv")
#define the left  edges of the age categories and the corresponding labels:
edges <- c(0,18,40,60, 120)
labels <- c("Child","Adult","Middle Aged","Over 60")
# we can break the ages in categories with the cut function
age.categories <- cut(titanic$Age,breaks = edges, right = FALSE, labels = labels)
# print the first 50 age items and the corresponding age categories)
age.categories[1:50]
```

```
##  [1] Adult       Adult       Adult       Adult       Adult
##  [6] <NA>        Middle Aged Child       Adult       Child
## [11] Child       Middle Aged Adult       Adult       Child
## [16] Middle Aged Child       <NA>        Adult       <NA>
## [21] Adult       Adult       Child       Adult       Child
## [26] Adult       <NA>        Adult       <NA>        <NA>
```

```
## [31] Middle Aged <NA>        <NA>        Over 60     Adult
## [36] Middle Aged <NA>        Adult       Adult       Child
## [41] Middle Aged Adult       <NA>        Child       Adult
## [46] <NA>        <NA>        <NA>        <NA>        Adult
## Levels: Child Adult Middle Aged Over 60
```

PYTHON

```
import pandas as pd
titanic = pd.read_csv("titanic.csv")
labels = ["Child","Adult","Middle Aged", "Over 60"]
edges = [0,18,40,60, 120]
age_categories =  pd.cut(titanic["Age"], edges, labels=labels)
print age_categories[0:20]
```

```
## 0           Adult
## 1           Adult
## 2           Adult
## 3           Adult
## 4           Adult
## 5             NaN
## 6     Middle Aged
## 7           Child
## 8           Adult
## 9           Child
## 10          Child
## 11    Middle Aged
## 12          Adult
## 13          Adult
## 14          Child
## 15    Middle Aged
## 16          Child
## 17            NaN
## 18          Adult
## 19            NaN
## Name: Age, dtype: category
## Categories (4, object): [Child < Adult < Middle Aged < Over 60]
```

5.2 Convert & Replace

Convert & Replace is a set of formulas that deal with converting and replacing columns or individual cells.

5.2.1 Category to Number

Category To Number is about converting nominal data to integer. Very often, prediction or machine learning functions don't accept nominal data, making it necessary to convert the field to integer if you want to make predictions. For instance, the column 'Sex' in the titanic dataset is nominal consisting of "male" and "female", which can be encoded to the integers 0/1, as follows:

R

```
titanic = read.csv("titanic.csv")
#define the left  edges of the age categories and the corresponding labels:
gender.encoded <- as.integer(as.factor(titanic$Sex))-1
#print subset
head(gender.encoded)
```

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

Python

```
import pandas as pd
titanic = pd.read_csv("titanic.csv")
gender_encoded =   pd.Categorical(titanic.Sex).codes
print gender_encoded[0:6]
```

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

Chapter 6

Final Words and this is strange

Chapter 7

Placeholder

Bibliography