University of Applied Sciences and Arts Northwestern Switzerland
School of Business

Data Science - Exercises

Holger Wache

Exercise D

Variable Transformation

Prerequisites

In this example we use a different data set which has much more qualitative data:

```
> titanic <- data.frame(Titanic)
> titanic
   Class    Sex   Age Survived Freq
1    1st   Male Child       No    0
2    2nd   Male Child       No    0
3    3rd   Male Child       No   35
4   Crew   Male Child       No    0
5    1st Female Child       No    0
6    2nd Female Child       No    0
7    3rd Female Child       No   17
…
```

As usual we make a copy…

```
my_titanic <- titanic
```

Transforming the categorial (nominal) variable Survived (1/2)

First we need really to transform the column/variable ”Survived” into a factor:

```
> f <- factor(titanic$Survived)
```

what are the possible values of the factor f?

```
> levels(f)
 [1] "Male"   "Female"
```

The factor f is internally already a number (an integer, in order to be precise)

```
> typeof(f)
[1] "integer"
```

Transforming the categorial (nominal) variable Survived (2/2)

Now we only need to transform f to an integer …

```
> as.integer(f)
```

… and write it into the column/variable “Survived”

```
> my_titanic$Survived <- as.integer(f)
```

Or everything in one line

```
> my_titanic$Survived <- as.integer(factor(titanic$Survived))
> my_titanic
Class    Sex   Age Survived Freq
1    1st   Male Child        1    0
2    2nd   Male Child        1    0
3    3rd   Male Child        1   35
4   Crew   Male Child        1    0
```

Transforming the categorial (nominal) variable Sex

The column/variable ”Sex” is NOT ordered. Therefore we can not transform it into a factor. But we can create a unique (boolean) column for each value.

We use a special package which supports us in this task

```
> install.packages("fastDummies")
> library(fastDummies)
```

“dummy_cols” selects the column “Sex”, remove it, and add for each value a new (0/1) columns, i.e. two colums “Sex_Male” and “Sex_Female”.

```
> my_titanic <- dummy_cols(my_titanic, select_columns="Sex", remove_selected_columns = TRUE)
> my_titanic
   Class   Age Survived Freq Sex_Male Sex_Female
1    1st Child        1    0        1          0
2    2nd Child        1    0        1          0
3    3rd Child        1   35        1          0
4   Crew Child        1    0        1          0
```

Transforming the ordinal variable Age (1/2)

Age is qualitative but ordered. This time we would like to influence how the different values are translated into number. An Adult is older than a child. Therefore Child = 1, Adult = 2

```
> ordered(my_titanic$Age, levels= c("Child", "Adult"))
 [1] Child Child Child Child Child Child Child Child Adult Adult Adult Adult Adult Adult Adult Adult
[17] Child Child Child Child Child Child Child Child Adult Adult Adult Adult Adult Adult Adult Adult
Levels: Child < Adult
```

Converting it into integers:

```
> as.integer(ordered(my_titanic$Age, levels= c("Child", "Adult")))
 [1] 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2
```

Transforming the ordinal variable Age (2/2)

Now we can replace the values in columns Age

```
> my_titanic$Age <- as.integer(ordered(my_titanic$Age, levels= c("Child", "Adult")))
```

… resulting into:

```
> my_titanic
   Class Age Survived Freq Sex_Male Sex_Female
1    1st   1        1    0        1          0
2    2nd   1        1    0        1          0
3    3rd   1        1   35        1          0
4   Crew   1        1    0        1          0
```

Transforming the (partly) ordinal variable Class (1/2)

Class is qualitative but partly ordered. While the three values (”1st”, “2nd”, “3rd”) are obviously ordered, is the value “Crew” a little bit separated from that. Therefore we need to have a Boolean column for “Crew” but an ordered column for the other three values.

Create a column just for the Crew:

```
> ifelse(my_titanic$Class =="Crew",1,0)
```

Add an additional column to the data set. Now the crew is separated:

```
> my_titanic$Class_Crew <- ifelse(my_titanic$Class =="Crew",1,0)
```

Transforming the (partly) ordinal variable Class (2/2)

Now order Class :

```
> ordered(my_titanic$Class, levels= c("Crew", "3rd", "2nd","1st"))
[1] 1st  2nd  3rd  Crew 1st  2nd  3rd  Crew 1st  2nd  3rd  Crew 1st  2nd  3rd  Crew 1st  2nd  3rd  Crew
[21] 1st  2nd  3rd  Crew 1st  2nd  3rd  Crew 1st  2nd  3rd  Crew
Levels: Crew < 3rd < 2nd < 1st
```

Converting it into integers:

```
> as.integer(ordered(my_titanic$Class, levels= c("Crew", "3rd", "2nd","1st")))
[1] 4 3 2 1 4 3 2 1 4 3 2 1 4 3 2 1 4 3 2 1 4 3 2 1 4 3 2 1 4 3 2 1
```

The right ‘Class numbers’ (i.e. subtract 1)

```
> as.integer(ordered(my_titanic$Class, levels= c("Crew", "3rd", "2nd","1st")))-1
[1] 3 2 1 0 3 2 1 0 3 2 1 0 3 2 1 0 3 2 1 0 3 2 1 0 3 2 1 0 3 2 1 0
```

```
> my_titanic$Class <-
      as.integer(ordered(my_titanic$Class, levels= c("Crew", "3rd", "2nd","1st")))-1
[1] 3 2 1 0 3 2 1 0 3 2 1 0 3 2 1 0 3 2 1 0 3 2 1 0 3 2 1 0 3 2 1 0
```

The resulting data set

```
> my_titanic
   Class Age Survived Freq Sex_Male Sex_Female Class_Crew
1      3   1        1    0        1          0          0
2      2   1        1    0        1          0          0
3      1   1        1   35        1          0          0
4      0   1        1    0        1          0          1
5      3   1        1    0        0          1          0
6      2   1        1    0        0          1          0
7      1   1        1   17        0          1          0
…
```