

Data Science - Exercises

Holger Wache





Exercise D

Variable Transformation





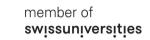
Prerequisites

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

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

As usual we make a copy...

```
my_titanic <- titanic</pre>
```





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

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

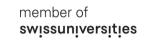
```
> f <- factor(titanic$Survived)
```

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)

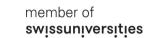
Now we only need to transform f to an integer ...

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

... and write it into the column/variable "Survived"

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

Or everything in one line





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".

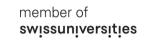


Transforming the ordinal variable Age (1/2)

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 Adult Child Child Child Child Child Adult A
```

Converting it into integers:



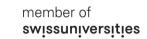


Transforming the ordinal variable Age (2/2)

Now we can replace the values in columns Age

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

... resulting into:





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

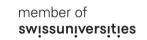
Class is qualitative but partly ordered. While the three values ("1st", "2nd", "3rd") are obviously ordered, is the value "Crew" a little bit separated from that. Therefore we need to have a Boolean column for "Crew" but an ordered column for the other three values.

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:

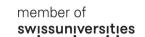
```
> ordered(my_titanic$Class, levels= c("Crew", "3rd", "2nd", "1st"))
[1] 1st 2nd 3rd Crew Levels: Crew < 3rd < 2nd < 1st
```

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
```

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
```





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
...
```