R Examples Repository

Home (http://www.uni-kiel.de/psychologie/rexrepos/index.html)
» RBasics (http://www.uni-kiel.de/psychologie/rexrepos/categories.html#RBasics-ref) » recode



**Tags**

RBasics (http://www.uni-kiel.de/psychologie/rexrepos/tags.html#RBasics-ref)

# Recode variables

- TODO
- Install required packages
- Recode numerical or character variables
  - Using index vectors
  - Using `recode()` from package `car`
  - Using `ifelse()`
- Cut continuous variables into categorical variables
  - Free recoding of value ranges into categories
  - Turn ordered value ranges into factor levels using `cut()`
- Recode factors
  - Add, combine and remove factor levels
  - Reorder factor levels

- Detach (automatically) loaded packages (if possible)
- Get the article source from GitHub

# TODO

- link to factors

# Install required packages

`car` (http://cran.r-project.org/package=car), `DescTools` (http://cran.r-project.org/package=DescTools)

```
wants <- c("car", "DescTools")
has   <- wants %in% rownames(installed.packages())
if(any(!has)) install.packages(wants[!has])
```

# Recode numerical or character variables

## Using index vectors

```
myColors <- c("red", "purple", "blue", "blue", "orange", "red", "orange")
farben   <- character(length(myColors))
farben[myColors == "red"]    <- "rot"
farben[myColors == "purple"] <- "violett"
farben[myColors == "blue"]   <- "blau"
farben[myColors == "orange"] <- "orange"
farben
```

```
[1] "rot"     "violett" "blau"    "blau"    "orange"  "rot"     "orange"
```

```
replace(c(1, 2, 3, 4, 5), list=c(2, 4), values=c(200, 400))
```

```
[1]   1 200   3 400   5
```

## Using `recode()` from package `car`

```
library(car)
recode(myColors, "'red'='rot'; 'blue'='blau'; 'purple'='violett'")
```

```
[1] "rot"     "violett" "blau"    "blau"    "orange"  "rot"     "orange"
```

```
recode(myColors, "c('red', 'blue')='basic'; else='complex'")
```

```
[1] "basic"   "complex" "basic"   "basic"   "complex" "basic"   "complex"
```

### Using `ifelse()`

```
orgVec <- c(5, 9, 11, 8, 9, 3, 1, 13, 9, 12, 5, 12, 6, 3, 17, 5, 8, 7)
cutoff <- 10
(reVec <- ifelse(orgVec <= cutoff, orgVec, cutoff))
```

```
 [1]  5  9 10  8  9  3  1 10  9 10  5 10  6  3 10  5  8  7
```

```
targetSet <- c("A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K")
response  <- c("Z", "E", "O", "W", "H", "C", "I", "G", "A", "O", "B")
(respRec  <- ifelse(response %in% targetSet, response, "other"))
```

```
 [1] "other" "E"     "other" "other" "H"     "C"     "I"     "G"
 [9] "A"     "other" "B"
```

## Cut continuous variables into categorical variables

### Free recoding of value ranges into categories

```
set.seed(123)
IQ <- rnorm(20, mean=100, sd=15)
ifelse(IQ >= 100, "hi", "lo")
```

```
 [1] "lo" "lo" "hi" "hi" "hi" "hi" "hi" "lo" "lo" "lo" "hi" "hi" "hi" "hi"
[15] "lo" "hi" "hi" "lo" "hi" "lo"
```

```
library(car)
recode(IQ, "0:100=1; 101:115=2; else=3")
```

```
 [1] 1 1 3 2 2 3 2 1 1 1 3 2 2 2 1 3 2 1 2 1
```

### Turn ordered value ranges into factor levels using `cut()`

```
IQfac <- cut(IQ, breaks=c(0, 85, 115, Inf), labels=c("lo", "mid", "hi"))
head(IQfac)
```

```
[1] mid mid hi  mid mid hi
Levels: lo mid hi
```

```
medSplit <- cut(IQ, breaks=c(-Inf, median(IQ), Inf))
summary(medSplit)
```

```
(-Inf,102] (102, Inf]
        10         10
```

```
IQdiscr <- cut(IQ, quantile(IQ), include.lowest=TRUE)
summary(IQdiscr)
```

```
[70.5,92.6]  (92.6,102]   (102,108]   (108,127]
          5           5           5           5
```

# Recode factors

## Add, combine and remove factor levels

### Add factor levels

```
(status <- factor(c("hi", "lo", "hi")))
```

```
[1] hi lo hi
Levels: hi lo
```

```
status[4] <- "mid"
status
```

```
[1] hi   lo   hi   <NA>
Levels: hi lo
```

```
levels(status) <- c(levels(status), "mid")
status[4] <- "mid"
status
```

```
[1] hi  lo  hi  mid
Levels: hi lo mid
```

### Combine factor levels

```
hiNotHi <- status
levels(hiNotHi) <- list(hi="hi", notHi=c("mid", "lo"))
hiNotHi
```

```
[1] hi    notHi hi    notHi
Levels: hi notHi
```

```
library(car)
(statNew <- recode(status, "'hi'='high'; c('mid', 'lo')='notHigh'"))
```

```
[1] high    notHigh high    notHigh
Levels: high notHigh
```

### Remove factor levels

```
status[1:2]
```

```
[1] hi lo
Levels: hi lo mid
```

```
(newStatus <- droplevels(status[1:2]))
```

```
[1] hi lo
Levels: hi lo
```

### Reorder factor levels

#### Using `reorder.factor()` from package `DescTools`

```
(facGrp <- factor(rep(LETTERS[1:3], each=5)))
```

```
 [1] A A A A A B B B B B C C C C C
Levels: A B C
```

```
library(DescTools)
(facRe <- reorder.factor(facGrp, new.order=c("C", "B", "A")))
```

```
 [1] A A A A A B B B B B C C C C C
Levels: C B A
```

#### Reorder group levels according to group statistics

```
vec <- rnorm(15, rep(c(10, 5, 15), each=5), 3)
tapply(vec, facGrp, FUN=mean)
```

```
       A         B         C
 7.800560  4.652087 16.635740
```

```
reorder(facGrp, vec, FUN=mean)
```

```
 [1] A A A A A B B B B B C C C C C
Levels: B A C
```

## Detach (automatically) loaded packages (if possible)

```
try(detach(package:car))
try(detach(package:DescTools))
```

# Get the article source from GitHub

R markdown (https://github.com/dwoll/RExRepos/raw/master/Rmd/recode.Rmd) - markdown (https://github.com/dwoll/RExRepos/raw/master/md/recode.md) - R code (https://github.com/dwoll/RExRepos/raw/master/R/recode.R) - all posts (https://github.com/dwoll/RExRepos/)

▸ top