

---

**recode**

---

syntax: `recode( data, test, newval )`

purpose: Copies the elements in `data`, changing those for which the corresponding `test` is satisfied to have the value `newval`.

example: 

```
>> x = [1 2 3 4 5];
>> y = recode(x, x>3, 999)
y: 1 2 3 999 999
```

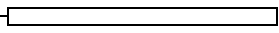
This command is useful when a range of numbers stands for the same event.

For example, consider simulating an insurance company whose policies have a deductible of \$500 and whose records indicate that incidents have a mean amount of damage of \$650 and follow an exponential distribution. You want to compute how much variance there will be in the payout for blocks of 100 incidents.

```
accidents = exponential(650); % create an urn
z = starttally;
for trials = 1:1000
    claims = sample(100, accidents);
    payout = claims - 500; % subtract the deductible
    % payouts cannot be negative
    payout = recode(payout, payout<0, 0);
    a = sum(payout);
    score a z;
end
percentile( z, [.025 .975])
```

for MATLAB experts: `RECODE` is closely related to the built-in `FIND` command. Note that in MATLAB, `if` statements cannot be applied to an entire vector. For instance, the following code does not work for a vector `x`

```
x = [1 2 3 4 5];
if x > 3 % this won't work for a vector x
    x = 999;
end
```



Instead, you can

```
>> newx = recode(x,x>3,999)
newx:  1 2 3 999 999
```

This document is an excerpt from

*Resampling Stats in MATLAB*

Daniel T. Kaplan

Copyright (c) 1999 by Daniel T. Kaplan, All Rights Reserved

This document differs from the published book in pagination and in the omission (unintentional, but unavoidable for technical reasons) of figures and cross-references from the book. It is provided as a courtesy to those

who wish to examine the book, but not intended as a replacement for the published book, which is available from

Resampling Stats, Inc.

[www.resample.com](http://www.resample.com)

703-522-2713

---