

DATA Step

Can I validate a postcode using SAS without a lookup to the postcode database?

While it is possible to obtain a copy of the 1.8M postcodes currently in use in the UK, and merge these against the desired field it may not be the most efficient method.

The following SAS function can be used to validate a text string and identify whether it follows the correct logic for a UK postcode.

```
options cmplib = work.funcs;

*** Return only complete postcodes if valid *** ;

proc fcmp outlib = work.funcs.pcode ;
  function pcode(post $) $ 8 ;
    post2 = upcase(post) ;
    *** Identify and validate incode *** ;
    ic_pos = prxmatch("/(\d[^CIKM0V0123456789,._\/-\(\)'\"s]+)\B/", post2) ;
    if ic_pos > 0 then
      do ;
        incode = substr(post2, ic_pos, 3) ;
        *** Identify possible outcode *** ;
        if ic_pos > 2 then outcode = scan(substr(post2, 1, ic_pos-1), -1) ;

        length vpc $ 8 ;
        vpc = catx(' ', outcode, incode) ;

        *** Validate outcode *** ;
        if not (vpc = 'GIR 0AA' /**** Known exception ****/
or
          /**** AN Outcode Type ****/
          prxmatch("/([A-Z]\d)\b/", outcode) = 1 or
          /**** ANN AAN ANA Outcode Types ****/
          prxmatch("/([A-Z]\d\d)\b/", outcode) = 1 or
          prxmatch("/([A-Z][A-H K-Y]\d)\b/", outcode) = 1 or
          prxmatch("/([A-Z]\d[ABCDEFGHJKSTUW])\b/", outcode) = 1 or
          /**** AANN AANA Outcode Types ****/
          prxmatch("/([A-Z][A-H K-Y]\d\d)\b/", outcode) = 1 or
          prxmatch("/([A-Z][A-H K-Y]\d[ABEHMNPRVWXY])\b/", outcode)
          ) then vpc = ' ' ;

        end ;
        return(vpc) ;
      endsub ;
run ;

data _null_ ;
```

DATA Step

```
postcode1 = 'Ez21yf' ;
postcode2 = 'Howe Street,EH36TD' ;
valid1 = pcode(postcode1) ;
put valid1 = ;
valid2 = pcode(postcode2) ;
put valid2 = ;
run ;
```

```
*** Partial Validation - return outcode only *** ;
```

```
proc fcmp outlib = work.funcs.pcode ;
  function pcode(post $) $ 8 ;
    post2 = upcase(post) ;
    *** Identify and validate incode *** ;
    ic_pos = prxmatch("/(\d[^CIKM0V0123456789,._\/-\(\)\s]+)\B/", post2) ;
    if ic_pos > 0 then incode = substr(post2,ic_pos,3) ;
    *** Identify possible outcode *** ;
    outcode = scan(substr(post2,1,ifn(ic_pos>2,ic_pos-1,max(length(post2),4))),1) ;
    length vpc $ 8 ;
    vpc = catx(' ',outcode,incode) ;

    *** Validate outcode *** ;
    if not (vpc = 'GIR 0AA' /**** Known exception ****/
or
    /**** AN Outcode Type ****/
    prxmatch("/([A-Z]\d)\b/",outcode) = 1 or
    /**** ANN AAN ANA Outcode Types ****/
    prxmatch("/([A-Z]\d\d)\b/",outcode) = 1 or
    prxmatch("/([A-Z][A-H K-Y]\d)\b/",outcode) = 1 or
    prxmatch("/([A-Z]\d[ABCDEFGHJKSTUW])\b/",outcode) = 1 or
    /**** AANN AANA Outcode Types ****/
    prxmatch("/([A-Z][A-H K-Y]\d\d)\b/",outcode) = 1 or
    prxmatch("/([A-Z][A-H K-Y]\d[ABEHMNPVRVWXY])\b/",outcode)
    ) then vpc = '' ;
  if vpc ne '' then return(vpc) ;
  else return(outcode) ;
endsub ;
```

DATA Step

```
run ;
```

```
data _null_ ;  
    postcode = 'EH2 9k3' ;  
    valid = pcode(postcode) ;  
    put valid = ;  
run ;
```

Unique solution ID: #1046

Author: Alan D Rudland

Last update: 2019-10-02 12:20