Making the most of census microdata

Practical 2: Exploring census microdata and data manipulation

In this session, you will be exploring the extent of unpaid caring in England and Wales using data from the 2011 microdata dataset that includes grouped local authorities.

You will need to apply the functions from the package dplyr that were previously explained.
Despite being a sample of the original microdata dataset, this file is still quite big. This may cause our computer to be slow and not very responsive to our requests. To avoid this, we can use a smaller version of the data, now is when `select()` and `filter()` come handy. Make sure you have **dplyr and haven** loaded into your session.

```
#Solution:
library(dplyr)
library(haven)
```

First, open a new script in R studio and save it in your working directory, so you will be able to access this script at a later time if you want to revise or modify a code. In R Studio:

Go to File… New File… R script

Task 1: Create a new dataset named carerdata that contains the following variables:

R is case sensitive so be careful when writing the variable names!

- caseno: Case number
- sex: sex
- ageg: Age of individual (grouped)
- aarer: Provision of unpaid carer
- housecarer: number of people in household who provide care
- ecopuk11: Economically active:
- popbasesec: whether usual resident,student living away or short term resident

```
#Solution:

carerdata<-select(censusd, caseno, sex, carer, housecarer,
ecopuk11, popbasesec, ageg)
```

Now we have a much smaller dataset containing only those selected variables that we need for our analysis.

Task 2: Edit the last dataset created selecting only those respondents that are usual residents

Our analysis will be focused on residents only, this will allow us to avoid the double counting of students living away during term time. In order to do that, we need to include only those cases that are "usual residents". The variable ***"popbasesec"*** classifies respondents into whether they are usual residents, students living away from home during term-time or short-term residents.

a. First, we need to know what code "usual residents" are represented by, so we need to do some manipulations.

Hint: Use the table option to check the categories of responses for ***popbasesec***. As a note, **table()** is not a function of the package **dplyr**.

```
table(carerdata$popbasesec)
```

```
##
##       1       2       3
## 2803842   34859    9448
```

Right now, the variable does not contain any visible labels that help us to identify each category of response, but a further inspection of the type of variable reveals that ***popbasesec*** is a **labelled** variable, which means that the categories or labels are attached, but they are not visible. The class function allows us to know the type of variable.

```
class(carerdata$popbasesec)

## [1] "labelled"
```

We can use the function `as_factor()` from the package **haven** to convert the variable into a factor variable (equivalent to categorical variable). Copy the following code in your console.

```
carerdata$popbasesecf<-as_factor(carerdata$popbasesec)
```

Here we created a new variable with the suffix "f" (for factor). Now use the table function on the new variable ***popbasesecf*** and check the categories.

```
#solution:
table(carerdata$popbasesecf)

##
##                                  Usual resident
##                                         2803842
## Student living away from home during term-time
##                                           34859
##                              Short-term resident
##                                            9448
```

b. Create a subsample of the data of "usual residents" only. Save this dataset under a different name such as **carer_res**

Hint: use the filter function

```
#Solution 1: Using the factor variable popbasesecf
carer_res<- filter(carerdata, popbasesecf=="Usual resident")
```

We can do the same using the "labelled" variable popbasesec. You can choose either solution, but not both!

```
#Solution 2: Using the labelled variables popbasesec
carer_res<- filter(carerdata, popbasesec==1)
```

- What is the number of cases that we now have in our dataset?
  2803842 cases

Task 3: Convert into factor all relevant variables in the last dataset created and check that the conversion went well by running the table function.

```
carer_res$sexf <- as_factor(carer_res$sex)
carer_res$carerf <- as_factor(carer_res$carer)
carer_res$housecarerf <- as_factor(carer_res$housecarer)
carer_res$ecopuk11f <- as_factor(carer_res$ecopuk11)
carer_res$agegf <- as_factor(carer_res$ageg)
```

Task 4: Recoding variables Recode the variable **carerf**: provision of unpaid care, into a binary variable that shows whether a respondent provides unpaid care or not.

```
# Solution:
table(carer_res$carerf)

##
##                No   Yes,1-19 hours Yes, 20-49 hours   Yes, 50+
hours
##          2514522           183000            38563
67757
```

Complete the following code:

```
#Solution:
carer_res$unpaidcare<- recode_factor(carer_res$carerf, "No" = "No",
                                     "Yes, 20-49 hours" = "Yes",
                                     "Yes, 50+ hours" = "Yes",
                                     "Yes,1-19 hours" = "Yes")

# check your new variable
table(carer_res$unpaidcare)

##
##      No     Yes
## 2514522  289320
```

Note: Remember R is case sensitive! The category "Yes,1-19 hours" is missing a space between the "," and "1-19", if we put a space between these two arguments, the code will not work.

Task 5: Get a description of sex, carer, housecarer and ecopuk11 using the functions table() and prop.table()

New function

`prop.table()` gives you the proportion for each category of your variable, it complements the table function that only shows the counts under each category. This is not a **dplyr** function.

Example:

```
#Solution:
table(carer_res$sexf)

##
##    Male  Female
## 1379549 1424293
```

```
prop.table(table(carer_res$sexf))
```

```
##
##      Male    Female
## 0.4920209 0.5079791
```

You can have percentage of this, by multiplying the expression by 100

```
prop.table(table(carer_res$sexf))*100
```

```
##
##     Male   Female
## 49.20209 50.79791
```

You can also run crosstabulations using the function **table** and **prop.table** as follows:

```
prop.table(table(variable1,variable2), 1) # row proportion
prop.table(table(variable1,variable2), 2) # column proportion
```

Now explore other variables and anwer the questions

5.1. What is(are) the age group(s) that provides most unpaid care? **The 55 and older age group**

```
# Solution:
prop.table(table(carer_res$agegf))*100
```

```
##
##  15 or less       16-24       25-34       35-44       45-54         55+
##    18.88206    11.88316    13.41845    13.98160    13.71903    28.11571
```

5.2. What is the proportion of females providing 20-49 hours of unpaid care? **1.56%**

```
# Solutions:
```

```
# standard crosstab
table(carer_res$sexf, carer_res$carerf)
```

```
##
##                No Yes,1-19 hours Yes, 20-49 hours Yes, 50+ hours
##   Male   1256825          79093            16403          27228
##   Female 1257697         103907            22160          40529
```

```
#Row proportion
prop.table(table(carer_res$sexf, carer_res$carerf),1)
```

```
##
##                  No Yes,1-19 hours Yes, 20-49 hours Yes, 50+ hours
##   Male   0.91104049     0.05733251       0.01189012     0.01973689
##   Female 0.88303249     0.07295339       0.01555860     0.02845552
```

```
# In percentages
prop.table(table(carer_res$sexf, carer_res$carerf),1)*100
```

```
##
##                 No Yes,1-19 hours Yes, 20-49 hours Yes, 50+ hours
##   Male   91.104049       5.733251         1.189012       1.973689
##   Female 88.303249       7.295339         1.555860       2.845552
```

5.3. What is the proportion of females providing unpaid care? **11.7%**

```
# Using the variable unpaidcare that was created in task 4:
prop.table(table(carer_res$sexf, carer_res$unpaidcare),1)*100
```

```
##
##                 No       Yes
##   Male   91.104049  8.895951
##   Female 88.303249 11.696751
```

Bonus

Search for more interesting features about the provision of unpaid care according to economic activity.