# Making sense of census microdata

## Tutorial 3: Creating aggregated variables and visualisations

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**: Joining two datasets

Very often, we need to use more than one source of data to get all the information required for our analysis. In those cases, instead of analysing those sources separately, we can incorporate the pieces of information into our master dataset. For those cases, the function join of **dplyr** is the best. Make sure you have **dplyr and haven** loaded into your session.

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

We need to add the variables region and local authority to our dataset in order to understand the patterns of unpaid care provisions in England and Wales. So join these two variables, which are available in the census_workshop.dta dataset (it is also the first dataset we imported into R)

**Hint**: Use the variable caseno as the key variable

**a.** Create a new dataset with the variables: Caseno, region and la_group (use the function "select") called **geo**.

```
# Solution:
geo<-select(censusd, caseno, region, la_group)
```

**b.** Join your dataset with the new one that contains the variables caseno, region and la_group (use the function "left_join").

```
# Solution
carer_geo<- left_join(carer_res, geo, by = "caseno")

# Now, convert into factor the variable region so you can see the
categories

carer_geo$region<-as_factor(carer_geo$region)
```

Your dataset now should have 15 variables.

**Task 2**: Create aggregated variables

Data aggregation is any process in which information is gathered and expressed in a summary form, for purposes such as statistical analysis. A common aggregation purpose is to get more information about particular groups based on specific variables such as age, profession, or income.

Creating aggregated variables is especially relevant when we are analysing census data, since it gives us the opportunity to obtain summary information at different levels, such as region or local authority.

The following example shows how to create aggregated summaries for each region in England and Wales, using some of the functions that we have already learned.

**Example**: Create a new variable that can show the proportion of females who provide unpaid care by region.

First, we group the dataset by region using the function `group_by`

```
census_regions<- group_by(carer_geo, region)
```

Next, we create an aggregated variable for regions, using the function `summarise`

```
example<- summarise(census_regions,
    tot_car=sum(unpaidcare=="Yes"),
    fem_carer = sum(sexf=="Female" & unpaidcare=="Yes"),
    pop=length(caseno),
    fem_carerpc = fem_carer/pop*100)

View(example)
```

You can see that a new dataset has been created with the name “example”. That dataset contains a summary with the following information: total number of individuals in the sample who provide unpaid care by region; total number in the sample of unpaid female carers; total number of individuals in the sample; and the proportion of females in the sample that provide unpaid care.

**Important about saving data**

You can save the table with your aggregated statistiscs as a comma-separated values (CSV) file that can be opened in Excel. You can also save it as a Stata or SPSS file. The code below shows how to save data into three different formats. This is valid for the datasets with microdata as well as for the dataset with the ageggregated variables. The files will be saved in your working directory.

```
### saving as csv file

write.csv(example, "example_aggregated.csv")

#the first argument is the object (i.e. data or variable) in R, the second
argument is the object as we want to save it; you can change the name here as
I did i.e. "example_aggregated.csv". Remember to put the new name within
"quotation marks" and write the file extension (.csv)

###  Saving as Stata

# The same instructions as before, but using a different function (read_dta)
and different extension (.dta)
write_dta(example, "example_aggregated.dta") # the aggregated variable
write_dta(censusd, "census_subsample.dta")  # the census dataset used

### Saving as SPSS

# The same instructions as before, but using a different function (read_sav)
and different extension (.sav)
write_sav(example, "example_aggregated.sav")
write_sav(censusd, "census_subsample.sav")
```

**Note**: This is optional

We can skip the step of creating a new dataset for the aggregated variables if we use pipes %>%

```
example_byregion<- carer_geo %>%
  group_by(region) %>%
  summarise(tot_car=sum(unpaidcare=="Yes"),
            fem_carer = sum(sexf=="Female" & unpaidcare=="Yes"),
            pop=length(caseno),
            fem_carerpc = fem_carer/pop*100)
```

**2.a**. Now add the proportion of males who also provide unpaid care by region. You can use pipes %>% if you prefer.

**Hint**: you need to edit the above code; we included the variables created in the example so we can have all the aggregated variables in the same dataset.

```
# Solution:

task2a<- summarise(census_regions,
                   tot_car=sum(unpaidcare=="Yes"),
                   fem_carer = sum(sexf=="Female" & unpaidcare=="Yes"),
                   male_carer = sum(sexf=="Male" & unpaidcare=="Yes")   ,
                   pop=length(caseno),
                   fem_carerpc = fem_carer/pop*100,
                   male_carerpc = male_carer/pop*100)
```

**now with pipes %>%**

```
task2aPipes<- carer_geo %>%
  group_by(region) %>%
  summarise(tot_car=sum(unpaidcare=="Yes"),
                   fem_carer = sum(sexf=="Female" & unpaidcare=="Yes"),
                   male_carer = sum(sexf=="Male" & unpaidcare=="Yes")   ,
                   pop=length(caseno),
                   fem_carerpc = fem_carer/pop*100,
                   male_carerpc = male_carer/pop*100)
```

**Note**: This is optional!

**2.b**. Include now the proportion of males and females that are "Economically Inactive, Looking after home/family".

**Hint**: you need to edit the above code.

```
task2b<-
  summarise(census_regions,
            tot_car=sum(unpaidcare=="Yes"),
            fem_carer = sum(sexf=="Female" & unpaidcare=="Yes"),
            male_carer = sum(sexf=="Male" & unpaidcare=="Yes" ),
            pop=length(caseno),
            fem_carerpc = fem_carer/pop*100,
            male_carerpc = male_carer/pop*100,
            fem_inac= sum (sexf=="Female" & ecopuk11==12, na.rm=TRUE),
            male_inac= sum (sexf=="Male" & ecopuk11==12, na.rm=TRUE),
            fem_inacpc = fem_inac/pop*100,
            male_inacpc = male_inac/pop*100)
```

**Note**: We have included the expression `na.rm=TRUE` which means that we are asking R to remove all the missing values, which are indentified as NA in R.

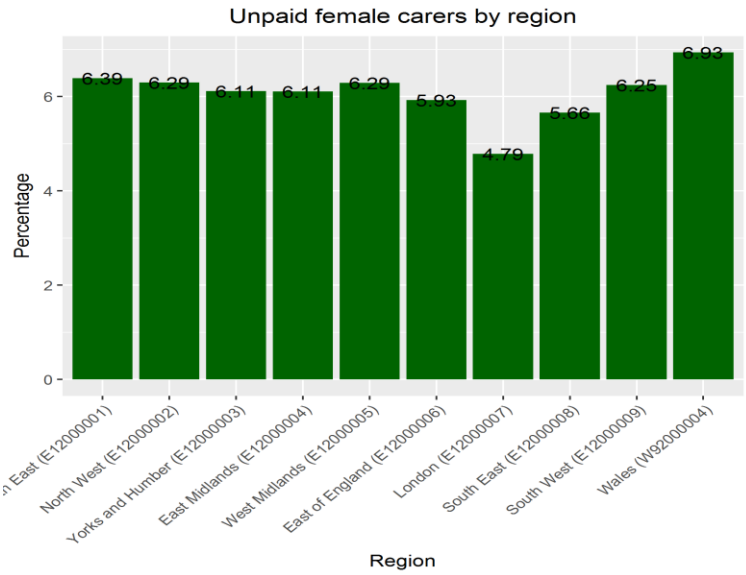
If we don't include this expresion, we will end up with a table with NA in those variables; you can check it for yourself if you want!

**Task 3**: Data visualization

**3a.** First install and load the package `ggplot2`.

```
install.packages("ggplot2")
library(ggplot2)
```

We are going to produce this plot that shows the proportion of females providing unpaid care in England and Wales using the aggregated variable from the example in task 2.




**Code in R:**

You can copy this code in the R Studio console

```
plot1 <- ggplot(example, aes(x=region, y= fem_carerpc))+
  geom_bar(stat="identity", fill= "Dark Green") +
  geom_text(aes(label=round(fem_carerpc, digits = 2))) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+
  ylab("Percentage")+
  xlab("Region")+
  ggtitle("Unpaid female carers by region")+
  theme(plot.title = element_text(hjust = 0.5))

print(plot1) # to see our plot

ggsave("plot1.png") # to save it into png format, you can use other formats
as .pdf
```

**Plot 1 explained**

**Basic Plot:** Only the two first lines of the above code are necessary to make a simple graph, the rest of the code are options to make it look better.

```
ggplot(example, aes(x=region, y= fem_carerpc))+
     geom_bar(stat="identity", fill= "Dark Green")
```

**1st line:** ggplot(task2a, aes(x=region, y= fem_carerpc))+

*Here we put on the x axis the variable region and on the y axis the variable percentage of female carers by region.*

`ggplot()`:
example is the name of the data that we are using.
`aes`: stands for aesthetics, it describes how the variables are used in the plot: x for the variable on the x axis and y for the variable on the y axis.

**2nd line:** `geom_bar(stat="identity", fill= "Dark Green)`

We add geom_bar (it can be: geom_points, geom_lines, etc) to ask for a bar graph. We could have left it empty, and we would get a plot with default options.

`geom_bar()`
`stat="identity",` leaves the data `(y= fem_carerpc)` as they are in the dataset (percentages). If not specified, ggplot will produce a plot with the count of cases (not useful in our case).
`fill= "Dark Green",` this is just for adding colours, by default the bars are grey.

**Additional functions:** The following options allow us to manipulate the look of our plot
**3rd line:** Here we are telling R that we want the percentages to be on top of the bars

```
geom_text(aes(label=round(fem_carerpc, digits = 2)))
```

**4th line***:* Here we are telling R that we want the x axis labels to display in a tilted angle (45 degrees).

```
theme(axis.text.x = element_text(angle = 45, hjust = 1))+
```

**5th line:** This is to give titles to each axis

```
ylab("Percentage")+ xlab("Region")+
```

**6th line:** Here we are giving a title to our graph

```
ggtitle("Unpaid female carers by region")+
```

**7th line:** Here we are centering the title of the graph. By default, it is positioned on the left-hand side

```
theme(plot.title = element_text(hjust = 0.5))
```

**More options**

Editing the x axis labels

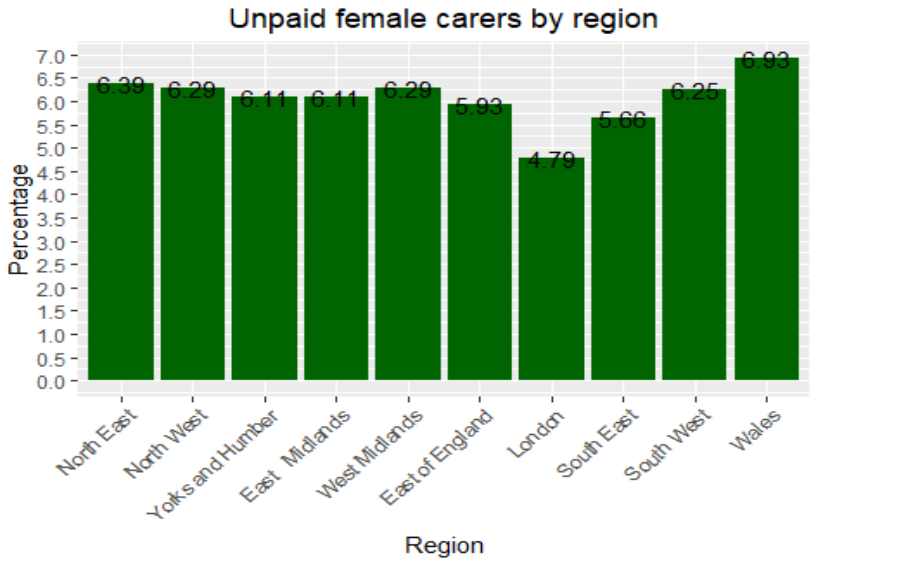
```
scale_x_discrete(labels = c("North East", "North West",
                            "Yorks and Humber",
                            "East   Midlands",
                            "West Midlands","East of England",
                            "London", "South East",
                            "South West", "Wales"))
```

Editing the y axis labels

```
scale_y_continuous(breaks = seq(0, 7, by = 0.))
```

This is an example of plot1 with the additional functions

```
plot1b<- ggplot(example, aes(x=region, y= fem_carerpc))+
  geom_bar(stat="identity", fill= "Dark Green") +
  geom_text(aes(label=round(fem_carerpc, digits = 2))) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+
  ylab("Percentage")+
  xlab("Region")+
  ggtitle("Unpaid female carers by region")+
  theme(plot.title = element_text(hjust = 0.5))+
  scale_x_discrete(labels = c("North East", "North West",
                              "Yorks and Humber",
                              "East   Midlands",
                              "West Midlands","East of England",
                              "London", "South East",
                              "South West", "Wales"))+
  scale_y_continuous(breaks = seq(0, 7, by = 0.5))
print(plot1b)
```



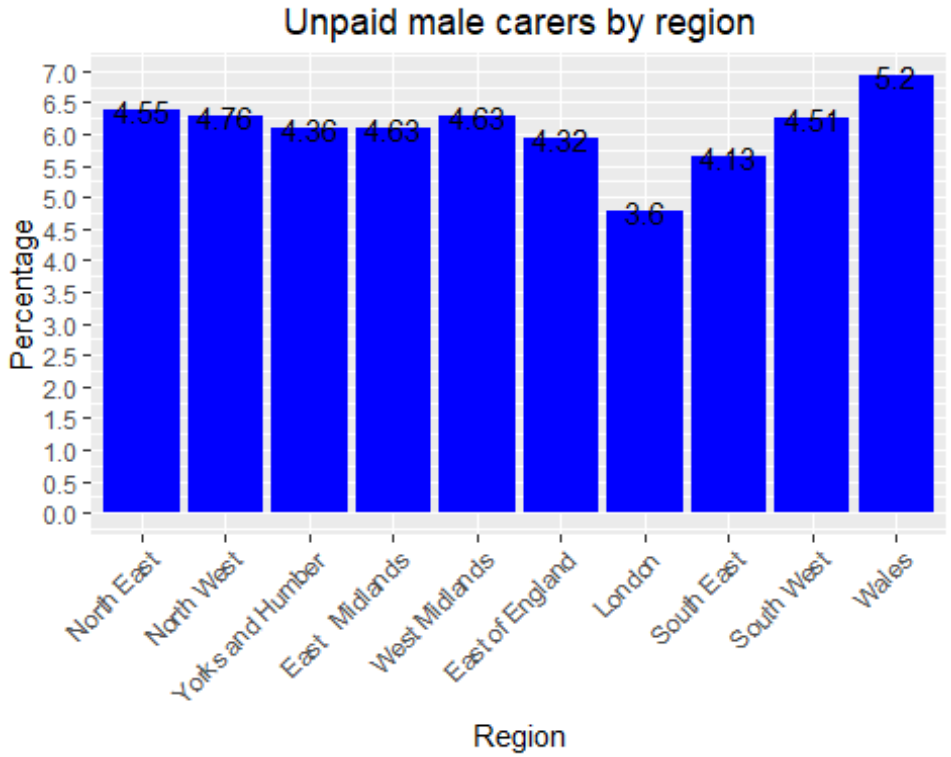
**3.b**. Use the code from the example above to create a bar plot of the proportion of males providing unpaid care.

Edit plot 1 to plot the proportion of males that provide unpaid care by regions.

```
# Solution

plot2<- ggplot(task2b, aes(x=region, y= fem_carerpc))+
  geom_bar(stat="identity", fill= "blue") +
  geom_text(aes(label=round(male_carerpc, digits = 2))) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+
  ylab("Percentage")+
  xlab("Region")+
  ggtitle("Unpaid male carers by region")+
  theme(plot.title = element_text(hjust = 0.5))+
  scale_x_discrete(labels = c("North East", "North West",
                              "Yorks and Humber",
                              "East   Midlands",
                              "West Midlands","East of England",
                              "London", "South East",
                              "South West", "Wales"))+
  scale_y_continuous(breaks = seq(0, 7, by = 0.5))

plot2
```



**Note:** Check this website: http://www.cookbook-r.com/Graphs/ for more examples making graphs with ggplot

## Bonus

In your own time, try to reproduce the map about the provision of unpaid care in England and Wales available here