

Each level adds more details; consider this example:

```
C - Manufacturing
C10 - Manufacture of food products
C10.1 - Processing and preserving of meat and production of meat products
C10.1.1 - Processing and preserving of meat
C10.1.2 - Processing and preserving of poultry meat
C10.1.3 - Production of meat and poultry meat products
```

So a company producing meat and poultry meat products would have NACE code level 4 C10.1.3 with it. Today for work I had to create a nice visualisation of the hierarchy of the NACE classification. It took me a bit of time to find a nice solution, so that's why I'm posting it here. Who knows, it might be useful for other people. First let's get the data. Because finding it is not necessarily very easy if you're not used to navigating Eurostat's website, I've put the CSV into a gist:

```
library(tidyverse)
library(data.tree)
library(igraph)
library(GGally)

nace_code <- read_csv("https://gist.githubusercontent.com/b-rodrigues/
4218d6daa8275acce80ebef6377953fe/raw/99bb5bc547670f38569c2990d2acad
a65bb744b3/nace_rev2.csv")

## Parsed with column specification:
## cols(
##   Order = col_double(),
##   Level = col_double(),
##   Code = col_character(),
##   Parent = col_character(),
##   Description = col_character(),
##   `This item includes` = col_character(),
##   `This item also includes` = col_character(),
##   Rulings = col_character(),
##   `This item excludes` = col_character(),
##   `Reference to ISIC Rev. 4` = col_character()
## )

head(nace_code)

## # A tibble: 6 x 10
##   Order Level Code Parent Description `This item incl...` `This item also...`
##
## 1 398481     1 A     AGRICULTUR... "This section i...
## 2 398482     2 01     A      Crop and a... "This division ... This division a...
## 3 398483     3 01.1   01     Growing of... "This group inc...
## 4 398484     4 01.11 01.1   Growing of... "This class inc...
## 5 398485     4 01.12 01.1   Growing of... "This class inc...
## 6 398486     4 01.13 01.1   Growing of... "This class inc...
## # ... with 3 more variables: Rulings , `This item excludes` ,
## #   `Reference to ISIC Rev. 4`
```

So there's a bunch of columns we don't need, so we're going to ignore them. What I'll be doing is transforming this data frame into a data tree, using the {data.tree} package. For this, I need columns that provide the hierarchy. I'm doing this with the next chunk of code. I won't explain each step, but the idea is quite simple. I'm using the `Level` column to create new columns called `Level1`, `Level2`, etc. I'm then doing some cleaning:

```
nace_code <- nace_code %>%
  select(Level, Code)

nace_code <- nace_code %>%
  mutate(Level1 = ifelse(Level == 1, Code, NA)) %>%
  fill(Level1, .direction = "down") %>%
  mutate(Level2 = ifelse(Level == 2, Code, NA)) %>%
  fill(Level2, .direction = "down") %>%
  mutate(Level3 = ifelse(Level == 3, Code, NA)) %>%
  fill(Level3, .direction = "down") %>%
  mutate(Level4 = ifelse(Level == 4, Code, NA)) %>%
  filter(!is.na(Level4))
```

Let's take a look at how the data looks now:

```
head(nace_code)

## # A tibble: 6 x 6
##   Level Code  Level1 Level2 Level3 Level4
##
## 1      4 01.11 A      01      01.1  01.11
## 2      4 01.12 A      01      01.1  01.12
## 3      4 01.13 A      01      01.1  01.13
## 4      4 01.14 A      01      01.1  01.14
## 5      4 01.15 A      01      01.1  01.15
## 6      4 01.16 A      01      01.1  01.16
```

I can now create the hierarchy using by creating a column called `pathString` and passing that data frame to `data.tree::as.Node()`. Because some sections, like C (manufacturing) are very large, I do this separately for each section by using the `group_by()`-`nest()` trick. This way, I can create a `data.tree` object for each section. Finally, to create the plots, I use `igraph::as.igraph()` and pass this to `GGally::ggnet2()`, which takes care of creating the plots. This took me quite some time to figure out, but the result is a nice looking PDF that the colleagues can now use:

```
nace_code2 <- nace_code %>%
  group_by(Level1, Level2) %>%
  nest() %>%
  mutate(nace = map(data, ~mutate(., pathString = paste("NACE2",
    Level1,
    Level2,
    Level3,
    Level4,
    sep = "/")))) %>%
  mutate(plots = map(nace, ~as.igraph(as.Node(.)))) %>%
  mutate(plots = map(plots, ggnet2, label = TRUE))

pdf("nace_maps.pdf")
pull(nace_code2, plots)
dev.off()
```