

# Problem Set #1

Sven Svensson

I'm basically going to teach you R Markdown by way of example. I obviously can't do your homework for you, but I can show you what an R Markdown document would look like and how you could use it to make nice documents for you. In the R Markdown document, do you see that above “chunk”? The one titled **setup**. You can load libraries in there and do all sorts of stuff in there as a type of “setup”. There are myriad advanced uses, but let's keep it simple and use it to make sure are libraries are loaded and that—for this purpose—each successive code chunk “echoes” (i.e. you see the code generating the result).

Let's look at the `eurostat_codes` data in `{stevedata}`. These are all states that have some kind of formal classification in Eurostat's glossary. Overall, there are 56 rows and 3 columns. Did you see what I did there?

Maybe we can check how many different “categories” there are in Eurostat's glossary.

```
n_distinct(eurostat_codes$cat)
```

```
## [1] 8
```

Interesting, there are 8 distinct categories. What are they? And how many are there, by category?

```
eurostat_codes %>%  
  count(cat)
```

```
## # A tibble: 8 x 2  
##   cat      n  
##   <chr> <int>  
## 1 EFTA      4  
## 2 ENP-E      3  
## 3 ENP-S     10  
## 4 EU       27  
## 5 EUCC      8  
## 6 OEC       1  
## 7 PC        2  
## 8 UK        1
```

Very cool. I can get all sorts of useful summary information from just the `count()` code. Notice `distinct()` does basically the same, but doesn't give me a count.

```
eurostat_codes %>%  
  distinct(cat)
```

```
## # A tibble: 8 x 1  
##   cat  
##   <chr>  
## 1 EU  
## 2 EFTA  
## 3 UK  
## 4 EUCC  
## 5 PC  
## 6 ENP-E
```

```
## 7 ENP-S
## 8 OEC
```

It doesn't even order them alphabetically. How about this?

```
eurostat_codes %>%
  summarize(n = n(),
            .by = cat)
```

```
## # A tibble: 8 x 2
##   cat      n
##   <chr> <int>
## 1 EU      27
## 2 EFTA     4
## 3 UK       1
## 4 EUCC     8
## 5 PC       2
## 6 ENP-E    3
## 7 ENP-S   10
## 8 OEC      1
```

Neat, another way of doing it. Not quite the same as `count()`. How about this?

```
eurostat_codes %>%
  summarize(n = n(),
            .by = cat) %>%
  arrange(cat)
```

```
## # A tibble: 8 x 2
##   cat      n
##   <chr> <int>
## 1 EFTA     4
## 2 ENP-E    3
## 3 ENP-S   10
## 4 EU      27
## 5 EUCC     8
## 6 OEC      1
## 7 PC       2
## 8 UK       1
```

Ah, yes, I did the thing, but needed more code to do it. Whatever, at least I see there are multiple ways of doing it.

One last thing, before I go: how might I create identify some “like” countries? For example, where are Armenia, Azerbaijan, and Georgia at?

```
eurostat_codes %>%
  filter(iso2c %in% c("AZ", "AM", "GE"))
```

```
## # A tibble: 3 x 3
##   country iso2c cat
##   <chr>    <chr> <chr>
## 1 Georgia  GE      PC
## 2 Armenia  AM      ENP-E
## 3 Azerbaijan AZ      ENP-E
```

Interesting. Despite all three being former SSRs from the same basic area, Georgia is a potential candidate and Armenia and Azerbaijan are in the neighborhood.

I won't do your homework for you, but I will link you to a `flygresor.se` jingle. Hela Sverige sjunger med!  
*flyg-, flyg-, flyg, flygresor-punkt-se!*