Reproducibility and Literate Programming

Rachel

2022-10-19

For this portfolio I will be using data obtained from the World Bank's DataBank on Gender Statistics [^1] to demonstrate how one might use literate programming to produce code that can easily be recreated by anyone reading this report. We first load the packages that we'll be making use of in our analysis [^1]: https://databank.worldbank.org/reports.aspx?source=gender-statistics#

Since there are approximately 1000 possible metrics and 265 countries to consider for a period of >20 years, I will only be focusing on the employment data for 2020, which I have downloaded into a CSV file directly from the website. To load this into our environment we can run the following (as long as the data is in the same directory as the markdown file)

```
data <- read_csv("Raw_Gender_Data.csv", show_col_types = FALSE)
```

If we look at a summary of the first two columns we can see they are just denoting the year, which is constant and thus not useful, thus we remove it. The fourth column contains a unique series code and the third column contains a longer description of the metric. Since this is the same information, we can remove the less concise third column. If we want to retrieve it, the data comes with a CSV file containing the series name for each series column.

```
data <- data[,-c(1,2,3)]
```

Then we might want to take a brief visual inspection of our data (for simplicity we just look at the first 15 rows and 5 countries):

```
knitr::kable(data[1:15,1:6],booktabs = TRUE) %>%
  kable_styling(font_size = 8, latex_options = "scale_down")
```

Series Code	United Kingdom [GBR]	Afghanistan [AFG]	Albania [ALB]	Algeria [DZA]	American Samoa [ASM]
SG.GET.JOBS.EQ	1	1	1	1	..
SG.NGT.WORK.EQ	1	0	1	0	..
SG.DNG.WORK.DN.EQ	1	0	1	0	..
SG.IND.WORK.EQ	1	0	1	1	..
SH.HIV.ARTC.FE.ZS	..	9	53	87	..
SH.HIV.ARTC.MA.ZS	..	9	45	80	..
SE.PRM.TENR	..	..	..	..	..
SE.PRM.TENR.FE	..	..	..	..	..
SE.PRM.TENR.MA	..	..	..	..	..
SP.ADO.TFRT	11.183	57.509	19.4332	9.3594	..
SH.HIV.PMTC.ZS	..	10	..	34	..
SH.STA.BRTC.ZS	..	..	..	..	..
SH.DTH.COMM.ZS	..	..	..	..	..
SH.DTH.COMM.0004.ZS	..	..	..	..	..
SH.DTH.COMM.0004.FE.ZS	..	..	..	..	..

Number of Missing Values	21	22	23	24	28	30	32	33	38	43	47	49	50	54	57	58	61	63	84	89	90	117	118	119
Frequency	1	1	3	11	30	25	1	2	2	12	1	2	2	3	1	2	1	8	3	3	5	3	3	3

Number of Missing Values	120	121	123	127	132	133	134	135	136	137	138	142	143	145	147	150	153	155	156	159	160	161	163	165
Frequency	5	3	6	1	2	3	1	2	6	4	3	3	5	4	1	1	3	3	2	4	1	1	3	1

Number of Missing Values	172	175	179	181	183	184	185	186	187	188	191	197	201	208	209	210	211	212	213	214	215	216	217
Frequency	1	2	9	4	3	6	3	3	1	2	1	2	7	1	2	1	1	24	37	51	15	221	5

We can see that there are many missing values, although here they are represented by an ellipses. We can then change these to NA, as this is what R recognises as missing values, and then see how many missing values there are in each row

```
data <- data %>% na_if("..")
na_freq <- tibble("Number of Missing Values" = 1:max(rowSums(is.na(data))), "Frequency" = tabulate(rowSu
na_freq <- na_freq[!(na_freq$Frequency == 0),]
knitr::kable(t(na_freq))%>%
  kable_styling(font_size = 8, latex_options = "scale_down")
```