R Markdown Tutorial

Leah

25 February, 2018

Contents

Now let's say that we want to have a nice PDF readout of some of our tables and analyses in R. The tool we will use is Rmarkdown. This allows you to generate a PDF document with output from R code with very little effort. There are lots of options for how you run your code chunks so that they turn out nicely.

Including or not including code chunks

Our first code chunk brings in our dataset, so it's not useful to include that in the markdown (and you don't see it below in the PDF output).

Now the below code ins included in our PDF, but I have opted not to have to see any of the warning messages that come with viewing it.

```r
sex_codebook <-
  tibble(SEX = c(1,2),
         sex_clean = c("Male","Female"))

educ_codebook <-
  tibble(EDUC = c(0:22, 97:99),
         educ_clean = c(rep("No Degree", 14),
                        rep("HS Diploma",2),
                        rep("Some College", 3),
                        rep("College Degree", 4),
                        rep(NA, 3)
                        )
         )

slim_df <-
  df %>%
  select(AGE, SEX, EDUC, HEALTH, HEIGHT, WEIGHT) %>%
  sample_frac(.1, replace = F) %>%
  mutate(BMI = WEIGHT/HEIGHT) %>%
  left_join(sex_codebook, by = "SEX") %>%
  left_join(educ_codebook, by = "EDUC")
```

Including tables

Let's say I now want to tell the story of how men and women get less healthy as they get older. The first thing I might want to show would be a table with the average health ratings of men and women compared with how old they are. I use the kable command to include a nice looking table in the markdown.

```r
age_health_gender <-
  slim_df %>%
  group_by(AGE, sex_clean) %>%
  summarise(avg = mean(HEALTH)) %>%
  spread(key = sex_clean, value = avg)

kable(age_health_gender, caption = "Average health as respondents age by gender")
```

Table 1: Average health as respondents age by gender

AGE	Female	Male
0	1.514064	1.550552
1	1.568058	1.626402
2	1.611626	1.660315
3	1.613460	1.673043
4	1.627810	1.669959
5	1.606327	1.655483
6	1.673001	1.652789
7	1.677172	1.759514
8	1.695079	1.675541
9	1.645816	1.721732
10	1.656040	1.720030
11	1.721260	1.748663
12	1.644670	1.722862
13	1.738215	1.766202
14	1.732648	1.700483
15	1.800495	1.736707
16	1.816170	1.732787
17	1.801630	1.747619
18	1.824030	1.822962
19	1.919372	1.786935
20	1.879278	1.879495
21	1.963706	1.848241
22	1.960340	1.883651
23	1.905550	1.877331
24	1.983240	1.891615
25	1.995563	1.917897
26	1.990557	1.951295
27	1.998183	1.924303
28	2.037819	1.931455
29	1.995467	1.930596
30	2.088306	1.927586
31	2.025431	1.934627
32	2.029514	1.960239
33	2.037267	1.984601
34	2.038095	1.985479
35	2.076078	1.989547
36	2.169611	2.089641

AGE	Female	Male
37	2.056764	2.016505
38	2.192180	2.002094
39	2.204953	2.097816
40	2.233083	2.098859
41	2.206747	2.112512
42	2.191057	2.148745
43	2.246339	2.084130
44	2.251304	2.217520
45	2.268235	2.235342
46	2.315057	2.218750
47	2.349870	2.265858
48	2.314159	2.322804
49	2.350443	2.383233
50	2.446082	2.425743
51	2.468158	2.345667
52	2.401229	2.363289
53	2.517114	2.413586
54	2.551818	2.356467
55	2.591025	2.490336
56	2.553875	2.523663
57	2.547150	2.532941
58	2.598969	2.563119
59	2.601383	2.510843
60	2.614228	2.629067
61	2.594315	2.519036
62	2.679860	2.621151
63	2.578880	2.666211
64	2.670782	2.661515
65	2.622525	2.633491
66	2.633094	2.627787
67	2.691834	2.707038
68	2.696063	2.646840
69	2.721477	2.751923
70	2.729290	2.772727
71	2.779863	2.683857
72	2.771203	2.798165
73	2.845339	2.790000
74	2.846316	2.775194
75	2.912955	2.829208
76	2.912951	2.817143
77	2.855792	2.796491
78	2.960784	2.888489
79	2.941177	2.956364
80	2.839793	3.014440
81	2.961652	2.950820
82	2.997024	2.985646
83	2.957746	2.984694
84	3.003846	2.978261
85	3.105702	3.035156

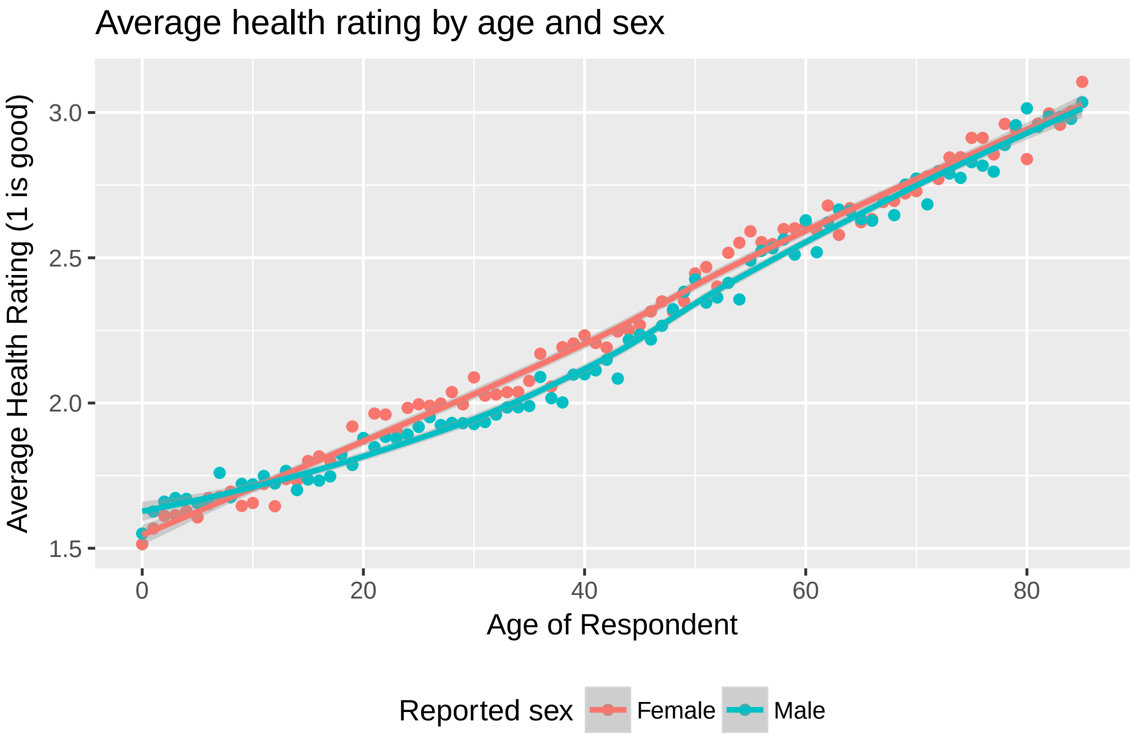
This table is so long it is useless, so let’s represent this information differently.

Including ggpplots

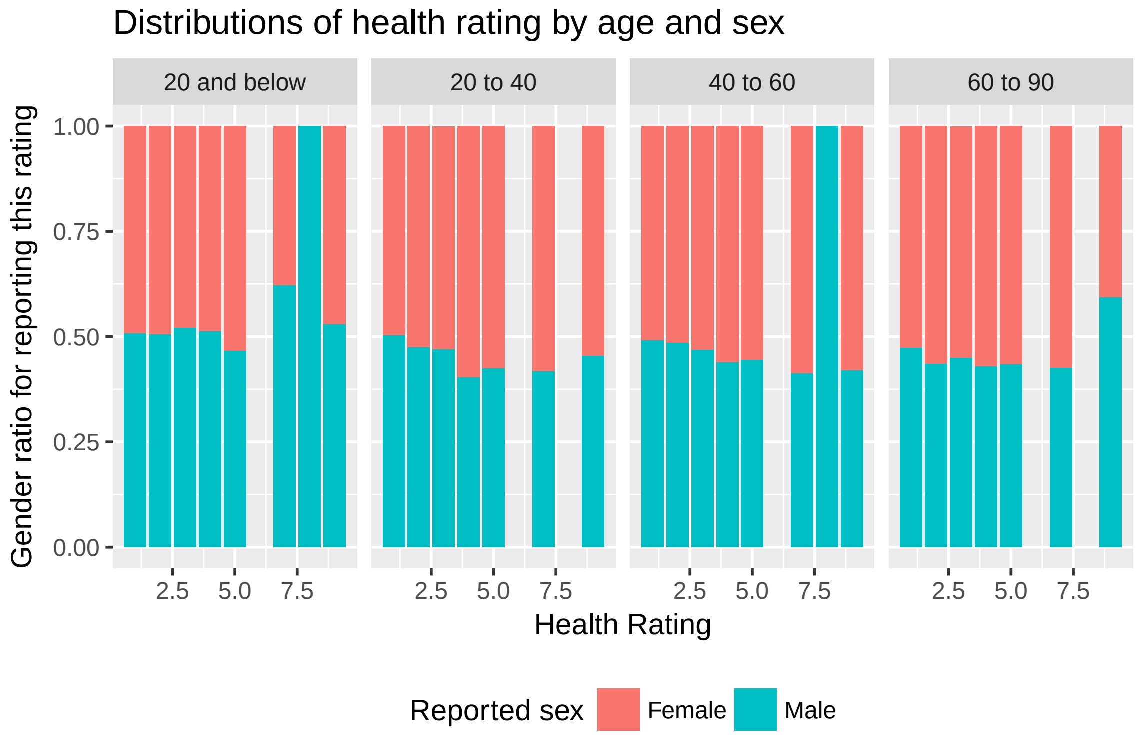
Now I want to further convince my readers by showing the difference in the distributions of our data for health rating by age and gender. To do this I will create a categorical age variable.

```
slim_df %>%
  group_by(AGE, sex_clean) %>%
  summarise(avg = mean(HEALTH)) %>%
  ggplot(aes(x = AGE, y = avg, color = sex_clean)) +
  geom_point() +
  geom_smooth() +
  theme(legend.position = 'bottom') +
  labs(title = "Average health rating by age and sex",
       x = "Age of Respondent",
       color = "Reported sex",
       y = "Average Health Rating (1 is good)")
```

```
## `geom_smooth()` using method = 'loess'
```



```
slim_df %>%
  mutate(age = case_when(AGE < 20 ~ "20 and below",
                         AGE < 40 ~ "20 to 40",
                         AGE < 60 ~ "40 to 60",
                         AGE < 90 ~ "60 to 90")) %>%
  ggplot(aes(x = HEALTH, fill = sex_clean)) +
  geom_bar(position = 'fill')+
  facet_grid(. ~ age) +
  theme(legend.position = 'bottom') +
  labs(title = "Distributions of health rating by age and sex",
       x = "Health Rating",
       fill = "Reported sex",
       y = "Gender ratio for reporting this rating")
```



Including regression output

Now the below code chunk has a few options. The `results='asis'` option is there to make the output of stargazer render nicely. The `echo = F` prevents the source code form being inlcuded, and the `warning = F` prevents any warnings form being included in the PDF.

% Table created by stargazer v.5.2 by Marek Hlavac, Harvard University. E-mail: hlavac at fas.harvard.edu % Date and time: Sun, Feb 25, 2018 - 13:40:13

Table 2:

	Dependent variable:	
	HEALTH	
	(1)	(2)
sex_cleanMale	−0.040***	−0.006
	(0.005)	(0.010)
AGE	0.017***	0.018***
	(0.0001)	(0.0002)
sex_cleanMale:AGE		−0.001***
		(0.0002)
Constant	1.534***	1.518***
	(0.005)	(0.007)
Observations	161,170	161,170
R^2	0.129	0.129
Adjusted R^2	0.129	0.129
Residual Std. Error	1.014 (df = 161167)	1.014 (df = 161166)
F Statistic	11,975.610*** (df = 2; 161167)	7,990.362*** (df = 3; 161166)
Note:		*p<0.1; **p<0.05; ***p<0.01