R Markdown Tutorial

Eric Karsten

24 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.

```
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.

```
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	1.524324	1.583480
1	1.592463	1.597911
2	1.666369	1.616952
3	1.625317	1.625935
4	1.641688	1.686678
5	1.646128	1.680033
6	1.677729	1.675569
7	1.650399	1.670683
8	1.710993	1.718949
9	1.664076	1.766435
10	1.684737	1.718400
11	1.682266	1.769171
12	1.650174	1.759876
13	1.715035	1.754272
14	1.707483	1.717061
15	1.756223	1.697752
16	1.806788	1.716323
17	1.792737	1.752756
18	1.907498	1.820779
19	1.945259	1.794258
20	1.911069	1.849142
21	2.020077	1.841766
22	1.972837	1.845070
23	1.939597	1.820000
24	2.016016	1.897773
25	1.960000	1.926641
26	1.953861	1.905045
27	1.972948	1.931907
28	2.003463	1.920762
29	1.992654	1.918580
30	2.080901	2.008850
31	2.070028	1.921799
32	2.011083	1.998182
33	2.106870	2.006499
34	2.042039	1.996071
35	2.152500	2.051304
36	2.096833	2.051758

AGE	Female	Male
37	2.100091	2.070381
38	2.154812	2.092019
39	2.198594	2.015224
40	2.157044	2.120561
41	2.199454	2.133770
42	2.203231	2.173561
43	2.212479	2.137714
44	2.273713	2.243243
45	2.319759	2.204013
46	2.354866	2.266537
47	2.297814	2.224740
48	2.379715	2.279314
49	2.378893	2.364151
50	2.389319	2.372549
51	2.411817	2.384317
52	2.426372	2.393258
53	2.488160	2.396702
54	2.451072	2.405631
55	2.500921	2.420901
56	2.564659	2.502146
57	2.616438	2.519641
58	2.592857	2.528694
59	2.523913	2.509530
60	2.697314	2.628176
61	2.674392	2.595147
62	2.645910	2.581646
63	2.707466	2.628075
64	2.671937	2.592170
65	2.659873	2.570225
66	2.660688	2.687285
67	2.648773	2.649395
68	2.652106	2.625862
69	2.662630	2.585321
70	2.725938	2.816822
71	2.712032	2.684466
72	2.809187	2.718894
73	2.818367	2.720982
74	2.762279	2.818428
75	2.854962	2.680000
76	2.810234	2.803234
77	2.916244	2.716216
78	2.880319	2.932907
79	2.971084	2.902985
80	2.983871	2.968254
81	2.914110	2.845133
82	2.958991	2.917391
83	3.053872	3.022472
84	2.930894	3.005102
85	3.024356	3.049875

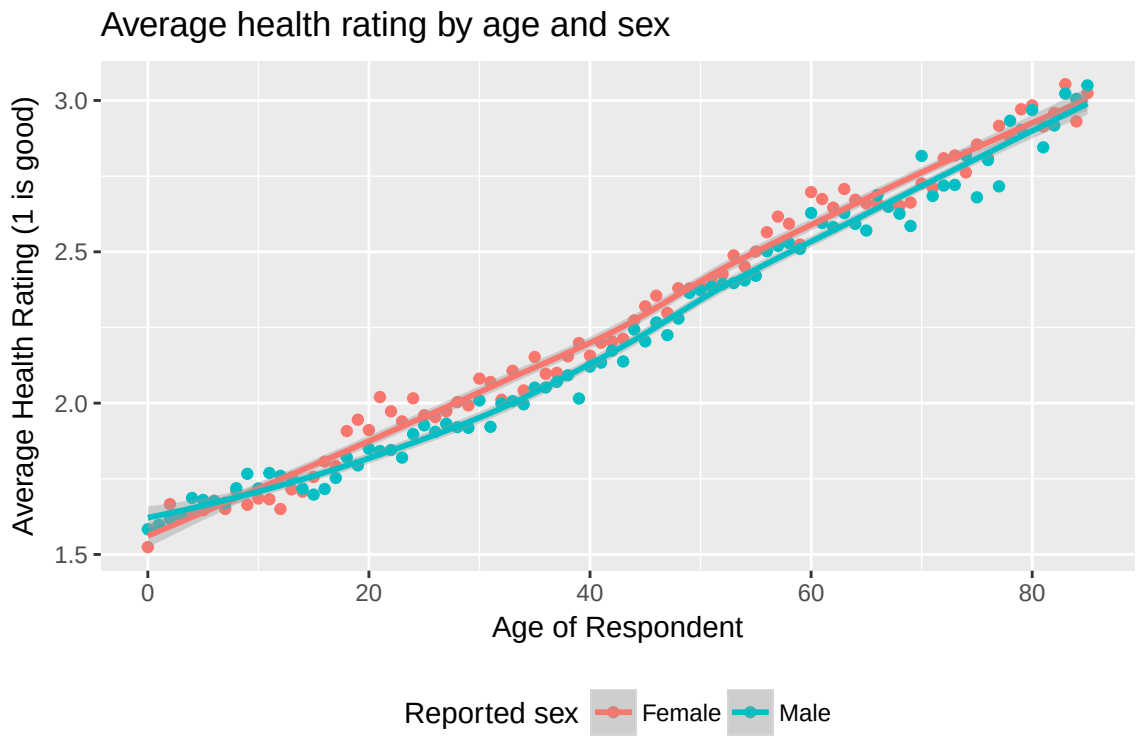
This table is so long it is useless, so let's represent this information differently.

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.

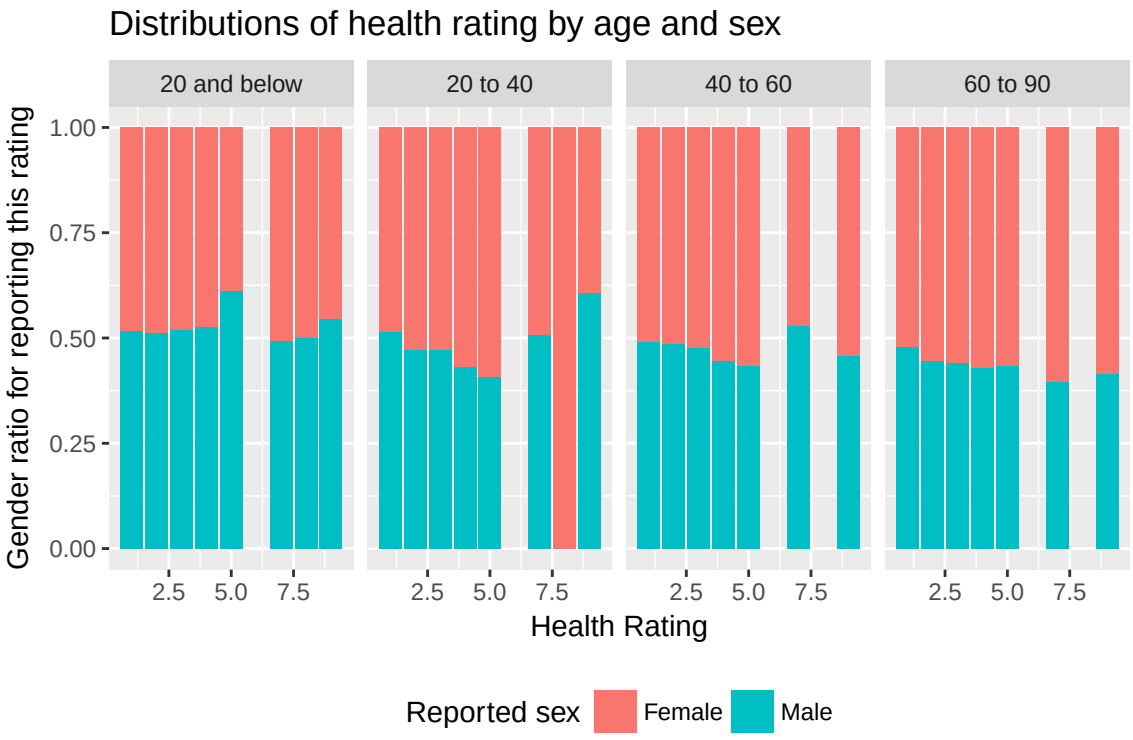
```
slim_df %>%
  group_by(AGE, sex_clean) %>%
  summarise(avg = mean(HEALTH)) %>%
  ggplot(aes(x = AGE, y = avg, color = sex_clean)) +
  geom_point() +
  geom_smooth() +
  theme(legend.position = 'bottom') +
  labs(title = "Average health rating by age and sex",
       x = "Age of Respondent",
       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: Sat, Feb 24, 2018 - 6:37:07 PM

Table 2:

	Dependent variable:	
	HEALTH	
	(1)	(2)
sex_cleanMale	-0.044^{***}	-0.012
	(0.005)	(0.010)
AGE	0.017^{***}	0.017^{***}
	(0.0001)	(0.0002)
sex_cleanMale:AGE		-0.001^{***}
		(0.0002)
Constant	1.547^{***}	1.532^{***}
	(0.005)	(0.007)
Observations	161,170	161,170
R^2	0.125	0.126
Adjusted R^2	0.125	0.125
Residual Std. Error	1.014 (df = 161167)	1.013 (df = 161166)
F Statistic	$11{,}556.770^{***}$ (df = 2; 161167)	$7{,}710.292^{***}$ (df = 3; 161166)
Note:		*p<0.1; **p<0.05; ***p<0.01