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	1.562440	1.542266
1	1.577679	1.626392
2	1.592396	1.645781
3	1.621008	1.641026
4	1.604255	1.689123
5	1.647513	1.672078
6	1.656871	1.732143
7	1.631266	1.668610
8	1.696147	1.731113
9	1.644726	1.714736
10	1.669887	1.660756
11	1.680885	1.748366
12	1.699052	1.720517
13	1.690537	1.790717
14	1.684039	1.709091
15	1.781942	1.751886
16	1.724675	1.735510
17	1.806581	1.739425
18	1.880907	1.744088
19	1.873747	1.824952
20	1.886608	1.866667
21	1.917154	1.850761
22	1.983162	1.859267
23	1.937618	1.851923
24	1.952811	1.879919
25	1.976127	1.902816
26	2.020389	1.862209
27	2.008137	1.924547
28	2.093500	1.886793
29	1.966055	1.885799
30	2.024671	1.948542
31	1.983532	1.922066
32	2.007005	1.952952
33	2.093545	1.982318
34	2.042591	1.964108
35	2.165943	2.010648
36	2.120307	1.954887

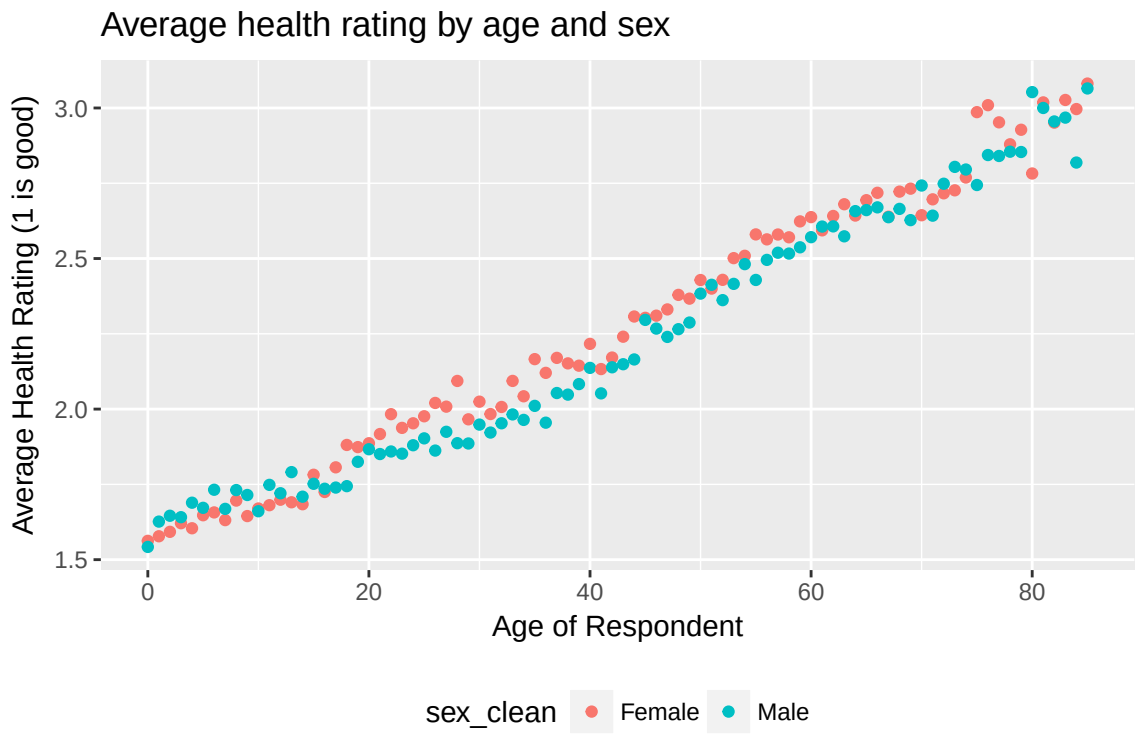
AGE	Female	Male
37	2.170044	2.053388
38	2.151867	2.048077
39	2.144120	2.083015
40	2.216692	2.136885
41	2.132861	2.052165
42	2.171127	2.138807
43	2.240369	2.149016
44	2.307484	2.164825
45	2.303079	2.296203
46	2.310259	2.267345
47	2.331028	2.239700
48	2.379250	2.265363
49	2.366788	2.287465
50	2.428571	2.383491
51	2.400187	2.412808
52	2.429200	2.361849
53	2.501322	2.415832
54	2.509108	2.481367
55	2.580090	2.428856
56	2.563655	2.495506
57	2.579798	2.519253
58	2.570520	2.516459
59	2.623362	2.537278
60	2.637550	2.570918
61	2.593602	2.606183
62	2.641345	2.606815
63	2.680307	2.573771
64	2.642761	2.657407
65	2.693798	2.660844
66	2.718248	2.669919
67	2.638806	2.637255
68	2.722222	2.664922
69	2.731993	2.627523
70	2.643836	2.742481
71	2.696629	2.642398
72	2.716567	2.748359
73	2.726547	2.804348
74	2.769072	2.795640
75	2.986220	2.744186
76	3.009324	2.843666
77	2.952261	2.840625
78	2.879397	2.854786
79	2.927614	2.853741
80	2.782396	3.052632
81	3.018518	3.000000
82	2.951289	2.955157
83	3.026578	2.967914
84	2.996390	2.818750
85	3.081025	3.064838

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

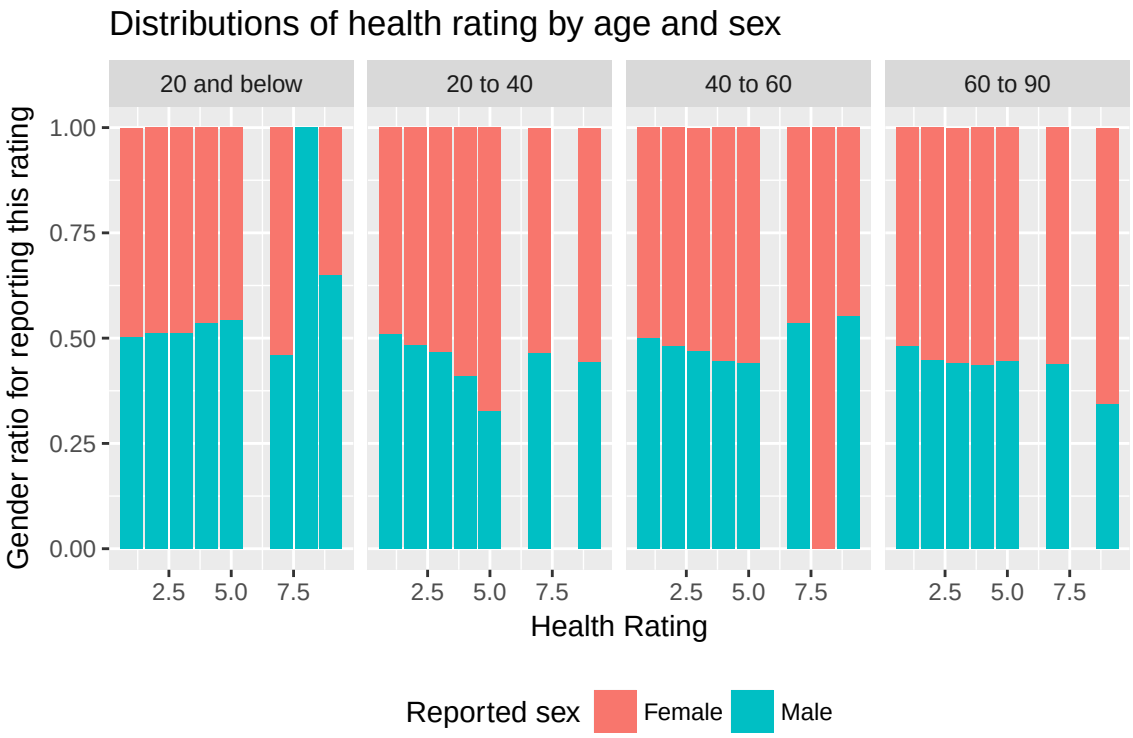
Including ggpplots

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() +
  theme(legend.position = 'bottom') +
  labs(title = "Average health rating by age and sex",
       x = "Age of Respondent",
       fill = "Reported sex",
       y = "Average Health Rating (1 is good)")
```




```
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 - 6:35:07 PM

Table 2:

	Dependent variable:	
	HEALTH	
	(1)	(2)
sex_cleanMale	-0.044^{***}	-0.005
	(0.005)	(0.010)
AGE	0.017^{***}	0.018^{***}
	(0.0001)	(0.0002)
sex_cleanMale:AGE		-0.001^{***}
		(0.0002)
Constant	1.532^{***}	1.513^{***}
	(0.005)	(0.007)
Observations	161,170	161,170
R^2	0.130	0.130
Adjusted R^2	0.130	0.130
Residual Std. Error	1.012 (df = 161167)	1.012 (df = 161166)
F Statistic	$12{,}021.210^{***}$ (df = 2; 161167)	$8{,}023.048^{***}$ (df = 3; 161166)
Note:		*p<0.1; **p<0.05; ***p<0.01