

Making Tables for Lott and Mustard

Here is some advice for making tables for the Lott and Mustard replication. I'll use the `mtcars` dataset as an example, but all the concepts will translate.

Kable takes a data frame

The `kableExtra` package can be used to turn data frames into LaTeX tables. See the code in my hidden curriculum GitHub (R/output/table_arrests_by_racegender.R lines 20-28) for an example of how to write the output to a .tex file, and read this pdf [here](#) for an in-depth explanation of the `kableExtra` package. You may want to look into the “booktabs”, “longtable”, and “grouped columns/rows” sections of that document.

In general, you just need to supply a properly shaped data frame to the `kbl` function, and control the LaTeX output with the `kbl` and `kable_styling` functions. Here's a table with 10 random rows from `mtcars`:

```
mtcars %>%
  sample_n(10) %>%
  kbl(
    caption = "sample\\_n(10) gives me ten random rows",
    format = "latex"
  ) %>%
  kable_styling(latex_options = c("striped", "hold_position"))
```

Table 1: sample_n(10) gives me ten random rows

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Lotus Europa	30.4	4	95.1	113	3.77	1.513	16.90	1	1	5	2
Toyota Corolla	33.9	4	71.1	65	4.22	1.835	19.90	1	1	4	1
Merc 450SL	17.3	8	275.8	180	3.07	3.730	17.60	0	0	3	3
Pontiac Firebird	19.2	8	400.0	175	3.08	3.845	17.05	0	0	3	2
Merc 280C	17.8	6	167.6	123	3.92	3.440	18.90	1	0	4	4
Maserati Bora	15.0	8	301.0	335	3.54	3.570	14.60	0	1	5	8
Honda Civic	30.4	4	75.7	52	4.93	1.615	18.52	1	1	4	2
Valiant	18.1	6	225.0	105	2.76	3.460	20.22	1	0	3	1
Merc 450SLC	15.2	8	275.8	180	3.07	3.780	18.00	0	0	3	3
Cadillac Fleetwood	10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4

What do you mean “properly shaped data frame”?

The hard part of making Table 2 from Lott and Mustard is shaping the data frame in the right way. We are summarizing different columns of the dataset, so the natural `dplyr` operation to use is `summarize`. On one hand, it's straightforward to summarize some columns like Lott and Mustard do. The code below will do it. You need to know what `across` does (see [here](#)) and what `pivot_longer` does (see [here](#)). Full disclosure, I did steal the `pivot_longer` piece from this [Stack Overflow](#). Try it without the `pivot_longer` command and see what happens.

```
summary_table <-
  mtcars %>%
  summarize(across(
    c(mpg, cyl, hp),
    list(
      "N. Obs" = length,
      "Mean" = mean,
      "Std. Dev" = sd
    )
  )) %>%
  pivot_longer(
    cols = everything(),
    names_sep = "_",
    names_to = c("Variable", ".value")
  )

summary_table %>%
  kbl(caption = "I summarized some variables",
      format = "latex") %>%
  kable_styling(latex_options = c("striped", "hold_position"))
```

Table 2: I summarized some variables

Variable	N. Obs	Mean	Std. Dev
mpg	32	20.09062	6.026948
cyl	32	6.18750	1.785922
hp	32	146.68750	68.562868

On the other hand, it's hard to select the appropriate columns and then rename them. The expression `c(mpg, cyl, hp)` in my code above selects the three columns that I want to summarize, but in your replication you will need to select many more. I encourage you to read more about the “tidy select” syntax [here](#) to figure out how to select the appropriate columns for your table.

Finally, you probably want to rename the columns to something more readable. You can use joins. Here's an example below. I strongly encourage you to read the documentation for [tribble](#), [left_join](#), and [coalesce](#) to understand what this code means. What's my translation table for? What does left join do? Why am I coalescing? You should be able to explain these things.

```
translation_table <- tribble(
  ~short_name, ~long_name,
  "mpg", "Miles per Gallon",
  "cyl", "Cylinders"
)

renamed_table <-
  summary_table %>%
  left_join(translation_table, c("Variable" = "short_name")) %>%
  mutate(Variable = coalesce(long_name, Variable)) %>%
  select(-long_name)

renamed_table %>%
  kbl(caption = "I summarized some variables with prettier names",
      format = "latex") %>%
  kable_styling(latex_options = c("striped", "hold_position"))
```

Table 3: I summarized some variables with prettier names

Variable	N. Obs	Mean	Std. Dev
Miles per Gallon	32	20.09062	6.026948
Cylinders	32	6.18750	1.785922
hp	32	146.68750	68.562868