Lab 8: Tables in R

2024-05-24

We have talked previously about summarizing data into aesthetic visuals, or graphs. But sometimes, what is important cannot be transformed into a pleasing picture, and requires a *table.*

Today we will discuss how to create tables that succinctly convey the desired information.

Tables

We will focus on the following types of tables today:

1. Summary tables: show summary statistics of data
2. Regression tables: report the results of one or multiple regressions

Summary Tables

Frequency Statistics

```
# Use the fivethirtyeight package
library(pacman)
p_load(fivethirtyeight, tidyverse) # tidyverse for cleaning

# Data on Congress members from the 113th congress (2013 - 2015)
age_data = congress_age %>% filter(congress == 113)

table(age_data$party, age_data$chamber)
```

```
##
##     house senate
##   D   202     57
##   I     0      2
##   R   237     46
```

This is nice! But it can look better. Let's make a table reporting the average age for each party within each chamber of Congress.

```
# new package: gt
p_load(gt)

age_table = age_data %>%
  group_by(chamber, party) %>%
  summarize(average_age = mean(age)) %>%
  pivot_wider(
    names_from = chamber,
    values_from = average_age
  )
```

```
## `summarise()` has grouped output by 'chamber'. You can override using the
## `.groups` argument.
```

```
age_table
```

```
## # A tibble: 3 x 3
##   party house senate
##   <chr> <dbl>  <dbl>
## 1 D      59.1   61.2
## 2 R      54.8   61.0
## 3 I      NA     70.0
```

```
# make a gt object
age_table = age_table %>% gt()

age_table
```

party	house	senate
D	59.08020	61.21930
R	54.77173	60.98913
I	NA	70.05000

Much better! We can make more changes:

- Add a title
- Capitalize the first letter of each word
- Write out the full name of each party (D → Democrat, etc.)
- Change the number of decimals to 2 for each entry

```
# from the gt packages
age_table %>%

  # title
  tab_header( title = "Average Age of Congress Members by Party and Chamber",
    subtitle = "In the 133th Congress") %>%

  # capitalize first word
  cols_label(party = "Party", house = "House", senate = "Senate") %>%

  # full party names
  text_case_match("D" ~ "Democrat", "R" ~ "Republican", "I" ~ "Independent") %>%

  # decimals
  fmt_number(decimals = 2)
```

Average Age of Congress Members by Party and Chamber

In the 133th Congress

Party	House	Senate
Democrat	59.08	61.22
Republican	54.77	60.99
Independent	NA	70.05

Regression Tables

One of the most common types of tables you'll have in presentations, reports, or research papers are those reporting the results of a regression (or multiple regressions).

For this example, we will use the `candy_rankings` dataset also from the `fivethirtyeight` package. It contains survey data from over 250,000 random match-ups between candies, the percent that each candy "won," and information on each candy (including if it contains chocolate, comes in a bar form, is fruity, etc.)

Suppose we want to know if having chocolate or peanuts/almonds makes a candy more likely to be chosen, or if the combination of chocolate and peanuts is more important. We can do this with two regressions:

$$winpercent \sim chocolate + peanutyalmondy$$

$$winpercent \sim chocolate + peanutyalmondy + chocolate \times peanutyalmondy$$

```
candy = candy_rankings

reg_1 = lm(data = candy, formula = winpercent ~ chocolate + peanutyalmondy)

reg_2 = lm(data = candy,
           formula = winpercent ~ chocolate + peanutyalmondy + chocolate:peanutyalmondy)
```

tab_model() function This produces an HTML output, so it won't look pretty if you knit to a PDF. However, it looks good when using it in an R script file or knitting to HTML.

```
# From package: sjPlot
p_load(sjPlot)

# Basic table
tab_model(reg_1, reg_2)
```

winpercent

winpercent

Predictors

Estimates

CI

p

Estimates

CI

p

(Intercept)

41.82

38.60 – 45.05

<0.001

42.46

39.25 – 45.67

<0.001

chocolateTRUE

16.62

11.37 – 21.88

<0.001

14.82

9.42 – 20.23

<0.001

peanutyalmondyTRUE

7.62

0.60 – 14.64

0.034

-7.60

-23.32 – 8.11

0.339

chocolateTRUE ×peanutyalmondyTRUE

18.82

1.35 – 36.30

0.035

Observations

85

85

R2 / R2 adjusted

0.437 / 0.423

0.467 / 0.448

Multiple adjustments can be made

- show.se = (TRUE / FALSE, whether to show standard errors of estimates)
- show.stat = (TRUE / FALSE, whether to show t-statistic)
- show.ci = (TRUE / FALSE, whether to show confidence interval)
- show.p = (TRUE / FALSE, whether to show p-value)
- pred.labels = c("list names of predictor variable(s)")
- dv.labels = c("list names of *dependent* variable(s)")
- string.pred = "Name for *Predictors* column"
- string.ci = "Name for *CI* column"
- string.p = "Name for *p* column"

Let's use some of these options to make our table prettier

1. Show estimates, standard errors, and p-value
2. Rename variables to "Chocolate," "Nuts," and "Chocolate and Nuts." Also change the dependent variable to "Win Percent"
3. Rename columns appropriately

```
tab_model(reg_1, reg_2,
        # automatically reports estimate, CI, p)
          show.ci = F,
          show.se = T,
        # rename variables
          pred.labels = c("Intercept", "Chocolate", "Nuts", "Chocolate and Nuts"),
          dv.labels = c("Win Percent", "Win Percent"), # wants it twice bc 2 models
        # Rename predictors column
          string.pred = "Variables",
          string.se = "Std. Err."
)
```

Win Percent

Win Percent

Variables

Estimates

Std. Err.

p

Estimates

Std. Err.

p

Intercept

41.82

1.62

<0.001

42.46

1.61

<0.001

Chocolate

16.62

2.64

<0.001

14.82

2.72

<0.001

Nuts

7.62

3.53

0.034

-7.60

7.90

0.339

Chocolate and Nuts

18.82

8.78

0.035

Observations

85

85

R2 / R2 adjusted

0.437 / 0.423

0.467 / 0.448

stargazer package Also produces multiple types of output: text, html

```
p_load(stargazer)

stargazer(reg_1, reg_2, type='text')
```

```
##
## ===============================================================================
##                                         Dependent variable:
##                                 ----------------------------------------------
##                                                winpercent
##                                       (1)                     (2)
## -------------------------------------------------------------------------------
## chocolate                           16.625***               14.823***
##                                      (2.640)                 (2.717)
##
## peanutyalmondy                       7.623**                 -7.602
##                                      (3.529)                 (7.899)
##
## chocolateTRUE:peanutyalmondy                                 18.824**
##                                                              (8.783)
##
## Constant                            41.825***               42.459***
##                                      (1.619)                 (1.612)
##
## -------------------------------------------------------------------------------
## Observations                           85                      85
## R2                                    0.437                   0.467
```

```
## Adjusted R2                          0.423                  0.448
## Residual Std. Error            11.173 (df = 82)       10.936 (df = 81)
## F Statistic                 31.848*** (df = 2; 82) 23.693*** (df = 3; 81)
## ============================================================================
## Note:                                          *p<0.1; **p<0.05; ***p<0.01
```

To get nice regression tables in an Rmarkdown file, include `results = 'asis'` in the brackets at the start of the chunk

```
# include results='asis' in {}

# type = html
stargazer(reg_1, reg_2, type='html')
```

Dependent variable:

winpercent

(1)

(2)

chocolate

16.625***

14.823***

(2.640)

(2.717)

peanutyalmondy

7.623**

-7.602

(3.529)

(7.899)

chocolateTRUE:peanutyalmondy

18.824**

(8.783)

Constant

41.825***

42.459***

(1.619)

(1.612)

Observations

85

85

R2

0.437

0.467

Adjusted R2

0.423

0.448

Residual Std. Error

11.173 (df = 82)

10.936 (df = 81)

F Statistic

31.848*** (df = 2; 82)

23.693*** (df = 3; 81)

Note:

p<0.1; ***p<0.05;*** p<0.01

There are a lot of options for changing the table within the stargazer package.

```
stargazer(reg_1, reg_2, type='html',
          # add a title
          title = "Candy Regressions",

          # label variables (covariates)
          covariate.labels = c("Chocolate", "Nuts", "Chocolate and Nuts", "Intercept"),

          dep.var.labels = "Win Percent",

          # number of digits in each number
          digits = 2,

          # choose which statistics to keep (n and r-squared)
          keep.stat = c("n","rsq"),

          ci.level=0.90
          )
```

Candy Regressions

Dependent variable:

Win Percent

(1)

(2)

Chocolate

16.62***

14.82***

(2.64)

(2.72)

Nuts

7.62**

-7.60

(3.53)

(7.90)

Chocolate and Nuts

18.82**

(8.78)

Intercept

41.82***

42.46***

(1.62)

(1.61)

Observations

85

85

R2

0.44

0.47

Note:

p<0.1; ***p<0.05;*** p<0.01

Can also set type = 'latex' to get the table in latex. Speaking of which....

```
stargazer(reg_1, reg_2, type='latex')
```

```
##
## % Table created by stargazer v.5.2.3 by Marek Hlavac, Social Policy Institute. E-mail: marek.hlavac a
## % Date and time: Thu, May 23, 2024 - 10:58:42
## \begin{table}[!htbp] \centering
##   \caption{}
##   \label{}
## \begin{tabular}{@{\extracolsep{5pt}}lcc}
## \\[-1.8ex]\hline
## \hline \\[-1.8ex]
##  & \multicolumn{2}{c}{\textit{Dependent variable:}} \\
## \cline{2-3}
## \\[-1.8ex] & \multicolumn{2}{c}{winpercent} \\
## \\[-1.8ex] & (1) & (2)\\
## \hline \\[-1.8ex]
##  chocolate & 16.625$^{***}$ & 14.823$^{***}$ \\
##   & (2.640) & (2.717) \\
##   & & \\
##  peanutyalmondy & 7.623$^{**}$ & $-$7.602 \\
##   & (3.529) & (7.899) \\
##   & & \\
##  chocolateTRUE:peanutyalmondy &  & 18.824$^{**}$ \\
##   &  & (8.783) \\
```

```
##   & & \\
##  Constant & 41.825$^{***}$ & 42.459$^{***}$ \\
##   & (1.619) & (1.612) \\
##   & & \\
## \hline \\[-1.8ex]
## Observations & 85 & 85 \\
## R$^{2}$ & 0.437 & 0.467 \\
## Adjusted R$^{2}$ & 0.423 & 0.448 \\
## Residual Std. Error & 11.173 (df = 82) & 10.936 (df = 81) \\
## F Statistic & 31.848$^{***}$ (df = 2; 82) & 23.693$^{***}$ (df = 3; 81) \\
## \hline
## \hline \\[-1.8ex]
## \textit{Note:}  & \multicolumn{2}{r}{$^{*}$p$<$0.1; $^{**}$p$<$0.05; $^{***}$p$<$0.01} \\
## \end{tabular}
## \end{table}
```

Tables in LaTeX

There may be cases in the future where you need a table for something unrelated to data, or it is showing qualitative information that can't be easily turned into a data frame and then into the table. Luckily, LaTeX has us covered! We can make all sorts of tables. Here's how the tables work:

- Start by centering your table with \begin { center } and at the end: \end { center }
- Create the table: \begin { tabular } { }
  - In the second set of brackets, use a *c* to denote an individual column, and a | (pipe symbol?) to denote a vertical line in the table.
- Use \hline to create a horizontal line
- Use ampersands & to *align* the lines of your columns
- End each line with \\ to let it know that's the end of that line