Controlling Table Output

Dr. Austin Brown

Besides ggplot2 output, another common way of communicating the results of statistical analyses, whether descriptive or inferential, is through tables. Let’s say we want to print out a table of the first 5 rows and 10 columns of the classic mtcars dataframe in our document.

datasets::mtcars[1:5,1:10]

## mpg cyl disp hp drat wt qsec vs am gear  
## Mazda RX4 21.0 6 160 110 3.90 2.620 16.46 0 1 4  
## Mazda RX4 Wag 21.0 6 160 110 3.90 2.875 17.02 0 1 4  
## Datsun 710 22.8 4 108 93 3.85 2.320 18.61 1 1 4  
## Hornet 4 Drive 21.4 6 258 110 3.08 3.215 19.44 1 0 3  
## Hornet Sportabout 18.7 8 360 175 3.15 3.440 17.02 0 0 3

Notice that the output here looks exactly the same as what we would see if we ran this in the console. Okay, well this may be fine for internal or informal types of documents/presentations, but what if this table was to be included in a company-wide memo? We need the ability to make a more professional appearing table. Fortuantely, there’s a function for that called knitr::kable. Let’s check out how we can beautify that table from before.

knitr::kable(  
 mtcars[1:5,1:10],  
 caption = "More Beautiful Car Data"  
)

More Beautiful Car Data

|  | mpg | cyl | disp | hp | drat | wt | qsec | vs | am | gear |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Mazda RX4 | 21.0 | 6 | 160 | 110 | 3.90 | 2.620 | 16.46 | 0 | 1 | 4 |
| Mazda RX4 Wag | 21.0 | 6 | 160 | 110 | 3.90 | 2.875 | 17.02 | 0 | 1 | 4 |
| Datsun 710 | 22.8 | 4 | 108 | 93 | 3.85 | 2.320 | 18.61 | 1 | 1 | 4 |
| Hornet 4 Drive | 21.4 | 6 | 258 | 110 | 3.08 | 3.215 | 19.44 | 1 | 0 | 3 |
| Hornet Sportabout | 18.7 | 8 | 360 | 175 | 3.15 | 3.440 | 17.02 | 0 | 0 | 3 |

This looks better! But the column names aren’t super clear! Let’s look at just the first four columns and see how we can change their names:

More Beautiful Car Data

| Miles per Gallon | Number of Cylinders | Engine Displacement | Horsepower |
| --- | --- | --- | --- |
| 21.0 | 6 | 160 | 110 |
| 21.0 | 6 | 160 | 110 |
| 22.8 | 4 | 108 | 93 |
| 21.4 | 6 | 258 | 110 |
| 18.7 | 8 | 360 | 175 |

There are lots of other packages which can help control aesthetic features of a table as well such as DT and xtable among many others. For example, with DT, we can create interactive tables when we render to HTML.

Now, besides just showing tables of raw data, let’s say we wanted to present the mean, median, and standard deviations of penguin bill length across the species of penguins in a nice table.

library(tidyverse)  
summary\_statistics <- palmerpenguins::penguins |>  
 dplyr::group\_by(species) |>  
 dplyr::summarize(Mean = mean(bill\_length\_mm,na.rm=T),  
 Median = median(bill\_length\_mm,na.rm=T),  
 `Standard Deviation` = sd(bill\_length\_mm,na.rm=T))  
  
knitr::kable(  
 palmerpenguins::penguins |>  
 dplyr::group\_by(species) |>  
 rstatix::get\_summary\_stats(bill\_length\_mm,type="common") |>  
 dplyr::select(species,mean,median,sd),  
 caption="Summary Stats via RStatix",  
 digits=4)

Summary Stats via RStatix

| species | mean | median | sd |
| --- | --- | --- | --- |
| Adelie | 38.791 | 38.80 | 2.663 |
| Chinstrap | 48.834 | 49.55 | 3.339 |
| Gentoo | 47.505 | 47.30 | 3.082 |

knitr::kable(summary\_statistics,  
 caption="Summary Statistics for Penguin Bill Length by Species",  
 digits = 4)

Summary Statistics for Penguin Bill Length by Species

| species | Mean | Median | Standard Deviation |
| --- | --- | --- | --- |
| Adelie | 38.7914 | 38.80 | 2.6634 |
| Chinstrap | 48.8338 | 49.55 | 3.3393 |
| Gentoo | 47.5049 | 47.30 | 3.0819 |

Now suppose we wanted to run a simple linear regression to assess the relationship between penguin bill length and bill depth. Typically, the results of regression are presented in an ANOVA type of table:

slr <- summary(lm(bill\_length\_mm~bill\_depth\_mm,data=palmerpenguins::penguins))  
knitr::kable(slr$coefficients)

|  | Estimate | Std. Error | t value | Pr(>|t|) |
| --- | --- | --- | --- | --- |
| (Intercept) | 55.0673698 | 2.5159514 | 21.887295 | 0.00e+00 |
| bill\_depth\_mm | -0.6498356 | 0.1457327 | -4.459093 | 1.12e-05 |

Let’s clean this up a bit! Typically, if we have really tiny p-values, instead of putting it in scientific notation, we just say “< 0.0001.”

coefs <- slr$coefficients  
pvals <- matrix(ifelse(coefs[,4] < 0.0001, "< 0.0001",coefs[,4]),ncol=1)  
coefs1 <- cbind(round(coefs[,-4],2),pvals)  
rownames(coefs1) <- c("Intercept","Bill Length (in mm)")  
knitr::kable(coefs1,  
 caption = "Results of Simple Linear Regression",  
 col.names = c("Estimate","Standard Error","Test Statistic","P-Value"),  
 row.names = T  
 )

Results of Simple Linear Regression

|  | Estimate | Standard Error | Test Statistic | P-Value |
| --- | --- | --- | --- | --- |
| Intercept | 55.07 | 2.52 | 21.89 | < 0.0001 |
| Bill Length (in mm) | -0.65 | 0.15 | -4.46 | < 0.0001 |