# LaTeX Output, Table Borders, and `hhline`

*Benjamin Nutter*

*2016-04-08*

## Contents

When rendering to LaTeXoutput, `pixiedust` offers two styles of table borders. Borders may be drawn using either the facilities of the `xcolor` and `arydshln` packages, or using the `hhline` packages (Note: these are LaTeXpackages). By default, `pixiedust` uses `xcolor` and `arydshln`.

## Comparison of Styles

| Feature | `xcolor` | `hhline` |
| --- | --- | --- |
| Dashed borders | Yes | No |
| Colored borders | Yes | No |
| Double borders | No | Yes |
| Works with background colors | No | Yes |

Before getting started, please note that the YAML front matter for this documents is as follows:

```
---
title: "LaTeX Output, Table Borders, and `hhline`"
author: "Benjamin Nutter"
date: "2016-04-08"
output: pdf_document
header-includes:
- \usepackage{amssymb}
- \usepackage{arydshln}
- \usepackage{caption}
- \usepackage{graphicx}
- \usepackage{hhline}
- \usepackage{longtable}
- \usepackage{multirow}
- \usepackage[dvipsnames,table]{xcolor}
- \makeatletter
- \newcommand*\vdashline{\rotatebox[origin=c]{90}{\$\dabar@\dabar@\dabar@\$}}
- \makeatother
---
```

For these illustrations, we will use our usual example linear model. We will also apply `medley_model` to the table before applying any other customizations.

```r
library(pixiedust)
options(pixiedust_print_method = "latex")


fit <- lm(mpg ~ wt + qsec + factor(am),
          data = mtcars)
```

## A Common Shortcoming

In both styles, vertical borders can become thicker than expected when adjoining cells have the adjoining borders defined. For example, if column 2 has a right border and column 3 has a left border, the adjoining borders appear as one thick border.

```r
dust(fit,
     caption = "Table borders using xcolor",
     hhline = FALSE,
     float = FALSE) %>%
  medley_model() %>%
  sprinkle(border = "all")
```

Table 2: Table borders using xcolor

| term | estimate | std.error | statistic | p.value |
|---|---|---|---|---|
| (Intercept) | 9.62 | 6.96 | 1.38 | 0.18 |
| wt | -3.92 | 0.71 | -5.51 | < 0.001 |
| qsec | 1.23 | 0.29 | 4.25 | < 0.001 |
| factor(am)1 | 2.94 | 1.41 | 2.08 | 0.047 |

```r
dust(fit,
     caption = "Table borders using hhline",
     hhline = TRUE,
     float = FALSE) %>%
  medley_model() %>%
  sprinkle(border = "all")
```

Table 3: Table borders using hhline

| term | estimate | std.error | statistic | p.value |
|---|---|---|---|---|
| (Intercept) | 9.62 | 6.96 | 1.38 | 0.18 |
| wt | -3.92 | 0.71 | -5.51 | < 0.001 |
| qsec | 1.23 | 0.29 | 4.25 | < 0.001 |
| factor(am)1 | 2.94 | 1.41 | 2.08 | 0.047 |

This can be avoided by only assigning one vertical border.

```r
dust(fit,
     caption = "Table borders using xcolor",
     hhline = FALSE,
     float = FALSE) %>%
  medley_model() %>%
  #* Left border on all columns
  sprinkle(cols = 1:5,
```

```
        border = c("left", "top", "bottom")) %>%
#* Right border on last column
sprinkle(cols = 5,
         border = "right")
```

Table 4: Table borders using xcolor

| term | estimate | std.error | statistic | p.value |
|---|---|---|---|---|
| (Intercept) | 9.62 | 6.96 | 1.38 | 0.18 |
| wt | -3.92 | 0.71 | -5.51 | < 0.001 |
| qsec | 1.23 | 0.29 | 4.25 | < 0.001 |
| factor(am)1 | 2.94 | 1.41 | 2.08 | 0.047 |

pixiedust provides a medley (`medley_all_borders`) that performs this task for you.

```
dust(fit,
     caption = "Table borders using the all borders medley",
     hhline = FALSE,
     float = FALSE) %>%
  medley_model() %>%
  medley_all_borders()
```

Table 5: Table borders using the all borders medley

| term | estimate | std.error | statistic | p.value |
|---|---|---|---|---|
| (Intercept) | 9.62 | 6.96 | 1.38 | 0.18 |
| wt | -3.92 | 0.71 | -5.51 | < 0.001 |
| qsec | 1.23 | 0.29 | 4.25 | < 0.001 |
| factor(am)1 | 2.94 | 1.41 | 2.08 | 0.047 |

**Dashed Borders**

**Colored Borders**

**Double Borders**

**Background Colors**