

Colortable

Ellis Hughes

2/16/2020

{{colortable}}

Examples of {{colortable}} in action!

```
library(colortable)
library(tidyverse)
library(knitr)
```

```
cell_sample <- color_vctr(24, text_color = "red", background = "blue")
cell_sample2 <- color_vctr(42, background = "yellow")
cell_sample3 <- color_vctr(68, text_color = "magenta", style = "strikethrough")
cell_sample4 <- color_vctr(70, text_color = "green")
```

cell_sample

```
## [1] 24
```

cell_sample2

```
## [1] 42
```

cell_sample3

```
## [1] 68
```

cell_sample4

```
## [1] 70
```

```
vect_sample <- color_vctr(cell_sample, cell_sample2, cell_sample3, cell_sample4)
vect_sample
```

```
## [1] 24 42 68 70
```

```
vect_sample2 <- vect_sample
vect_sample2[5] <- 422
vect_sample2[20] <- color_vctr(98119, text_color = "yellow", background = "blue", style = "underline")
vect_sample2[6:7] <- c(21,23)
vect_sample2[10:12] <- color_vctr(cell_sample, cell_sample2, cell_sample3)

vect_sample2
```

```
## [1] 24 42 —68 70 422 21 23 NA NA 24 42 —68
## [13] NA NA NA NA NA NA NA 98119
```

```
data.frame(idx = 1:5, z = vect_sample[1:5])
```

idx	z
1	24
2	42
3	68
4	70
5	NA

```
color_tibble <- tibble(idx = 1:5, z = vect_sample[1:5])
```

```
color_tibble
```

idx	z
1	24
2	42
3	68
4	70
5	NA

```
color_tibble %>%
  kable(escape = FALSE)
```

idx	z
1	24
2	42
3	68
4	70
5	NA

Use Cases

The ability to update coloring within the table allows for visualizing the results before printing and rendering.

One use case could be trying to print out p-values and drawing attention to the significant p-values.

Normally, the course of action would be to manually add either the latex or html required to tag the outputs. This requires both knowing how to tag the significant pvalues with the correct latex/html code and also hard codes those results into your code.

```
## Super Great analysis of mtcars!
```

```
lm_fit <- lm(mpg ~ ., mtcars)
```

```
a_lm_fit <- anova(lm_fit)
```

```
df_anova <- data.frame(a_lm_fit)
```

```
# if the output is pdf
df_anova$Pr..F. <- ifelse(
  df_anova$Pr..F. < .05,
  paste0("\\textcolor{green}{", df_anova$Pr..F., "}"),
  df_anova$Pr..F.
)

kable(df_anova)
```

	Df	Sum.Sq	Mean.Sq	F.value	Pr..F.
cyl	1	817.7129524	817.7129524	116.4245456	<u>5.03444973840481e-10</u>
disp	1	37.5939529	37.5939529	5.3525615	<u>0.0309108258078556</u>
hp	1	9.3709293	9.3709293	1.3342166	0.261031043915007
drat	1	16.4674349	16.4674349	2.3446047	0.140643762276576
wt	1	77.4757948	77.4757948	11.0308687	<u>0.00324449159445386</u>
qsec	1	3.9493082	3.9493082	0.5622956	0.461655702242183
vs	1	0.1297687	0.1297687	0.0184762	0.893173302477966
am	1	14.4742372	14.4742372	2.0608167	0.165857678951404
gear	1	0.9717105	0.9717105	0.1383504	0.71365333783354
carb	1	0.4066688	0.4066688	0.0579008	0.812178712952693
Residuals	21	147.4944300	7.0235443	NA	NA

{{colortable}} can resolve this and make your code much easier to understand, and you can add additional styling just as easily. There is also the added benefit that even though we have styling on the cells, the underlying object type still exists and can be modified and edited as needed.

```
tbl_anova <- data.frame(a_lm_fit)

tbl_anova$Pr..F. <-
  set_styling(tbl_anova$Pr..F. , tbl_anova$Pr..F. < 0.05, text_color = "green", style = "underline")

tbl_anova
```

	Df	Sum.Sq	Mean.Sq	F.value	Pr..F.
cyl	1	817.7129524	817.7129524	116.42454564	<u>5.034450e-10</u>
disp	1	37.5939529	37.5939529	5.35256153	<u>3.091083e-02</u>
hp	1	9.3709293	9.3709293	1.33421658	2.610310e-01
drat	1	16.4674349	16.4674349	2.34460470	1.406438e-01
wt	1	77.4757948	77.4757948	11.03086869	<u>3.244492e-03</u>
qsec	1	3.9493082	3.9493082	0.56229561	4.616557e-01
vs	1	0.1297687	0.1297687	0.01847624	8.931733e-01
am	1	14.4742372	14.4742372	2.06081667	1.658577e-01
gear	1	0.9717105	0.9717105	0.13835045	7.136533e-01
carb	1	0.4066688	0.4066688	0.05790079	8.121787e-01
Residuals	21	147.4944300	7.0235443	NA	NA