

# R Markdown Template

## MPS Theme Package Demo

Written By: [Morrigan Hughes](#)

April 2024

## Testing a Level 1 Heading

Testing a Level 2 Heading

Testing a Level 3 Heading

## Displaying text in a document

### Font styles

This is normal text.

This is **bolded** text.

This is *italicized* text.

This is <sup>superscript</sup> text.

This is ~~strikethrough~~ text.

### Long Paragraphs:

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Morbi a odio in purus dapibus auctor. Aliquam dictum, diam sed molestie vestibulum, tellus lectus consequat magna, vel condimentum eros lectus et mauris. Duis quis rhoncus neque. Suspendisse ac eros in nisl ultricies egestas ac in metus. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Pellentesque et ex lorem. Pellentesque commodo purus id justo malesuada, quis venenatis mi fringilla. Nam egestas ipsum nisl, placerat fringilla mauris pretium quis. Nam fermentum, mi sollicitudin suscipit efficitur, orci nisl accumsan felis, sit amet dapibus diam ante nec eros. Donec malesuada pretium mi, sed commodo massa auctor vel. Nunc blandit nisi dolor, ut pulvinar erat rhoncus luctus. Aliquam id ornare nunc. Vivamus dignissim, risus eu dictum gravida, enim ipsum euismod lorem, vel consectetur lectus orci at tellus.

## R Code Chunk:

```
1 summary(mtcars$mpg)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    10.40   15.43   19.20   20.09   22.80   33.90
```

## Block Quote:

*Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean sed tincidunt urna. Sed scelerisque sit amet nunc et maximus. Maecenas et posuere quam. Pellentesque aliquam volutpat auctor. Duis non neque luctus felis varius vehicula at quis mi. Etiam sed sem neque. Nam odio nulla, suscipit eu erat non, feugiat posuere massa.*

## Non-evaluated code blocks:

```
summary(mtcars$mpg)
```

## R Code blocks with eval=FALSE

```
1 summary(mtcars$mpg)
```

## Inline Code

You can use `scale_color_mps()` with your plots now!

## Special notes, warnings, and other callouts

Use `\notebox{Some Note}`, `\warnbox{Some warning}`, `\impbox{Something important}`, and `\specbox{Box Title}{Box Content}` for the special block notes and warnings seen below:

## Notes

```
\notebox{This is a note- something interesting worth extracting and pointing out to  
↪ the reader! We, of course, should show what happens when our notes get pretty  
↪ long, too! Gotta make sure text wrapping actually works.}
```

### NOTE

This is a note- something interesting worth extracting and pointing out to the reader! We, of course, should show what happens when our notes get pretty long, too! Gotta make sure text wrapping actually works.

## Warnings

```
\warnbox{This is a warning: Something critical you should know before taking action!}
```

### WARNING

This is a warning: Something critical you should know before taking action!

## Important Messages

```
\impbox{This is an important notice- Something is important!}
```

### IMPORTANT

This is an important notice- Something is important!

## Other Messages

```
\specbox{EXAMPLE}{This is some example text.}
```

### EXAMPLE

This is some example text.

## Adding captions to figures, plots, and tables

You can use `mps_caption(caption, type)` to caption your plots, tables, and figures with an automatically incremented number and whatever caption you want to give the item. You do this in the body text **after** the code chunk that displays your table, figure, or plot.

```
1 ## Usage
2 # Plots
3 mps_caption("Some Plot Caption/Alt Text", "plots")
4
5 # Tables
6 mps_caption("Some Table Caption", "tables")
7
8 # Figures
9 mps_caption("Some figure caption", "figures")
```

Figure 1. Usage of the new `mps_caption` function.

## Changing the color of inline text

You can use `mps_tcolor(color, text, mps = TRUE/FALSE)` to change the color of text inline. `mps = TRUE` is the default behavior of the function. For example, **this is wine-colored**, while **this is teal-colored**.

Available colors when argument `mps = TRUE`: blue, darkblue, gray, green, orange, purple, red, teal, wine, yellow.

And when argument `mps = FALSE`: black, blue, brown, cyan, darkgray, gray, green, lightgray, lime, magenta, olive, orange, pink, purple, red, teal, violet, , yellow.

## MPS Theme Package and `ggplot()`

There are, as of 18-Apr-2024, **8** user-available `ggplot()` functions in the new `mpstheme` package. The functions are below, and their documentation can be retrieved using the R Console by typing in `?function` [e.g. `?mps_cols`]. The currently available `ggplot()` functions are as follows:

- `mps_cols()`
- `scale_color_mps()`
- `scale_fill_mps()`
- `theme_mps()`
- `theme_mps_donut()`

- title\_mps()
- mps\_fonts()
- geom\_label\_mps()

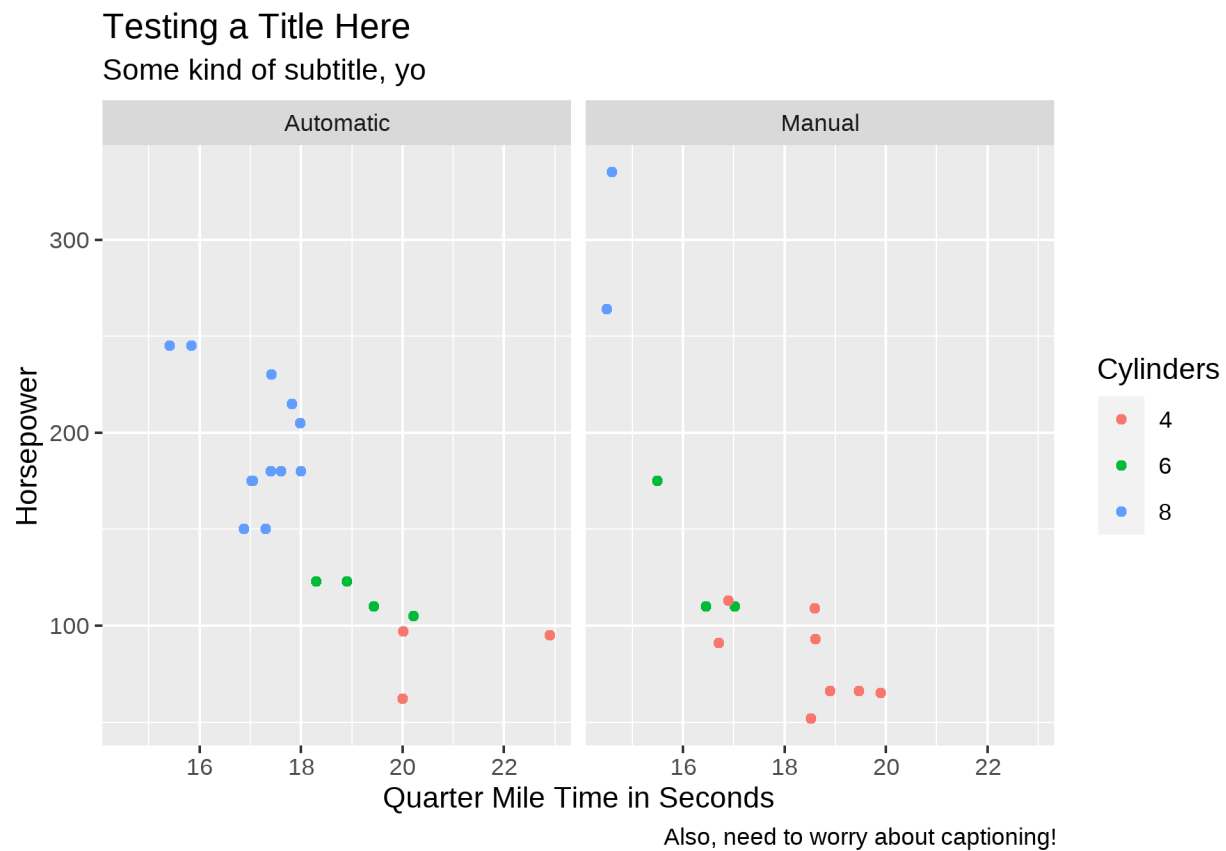
## Demonstration of Functions

We will begin by using the base `mtcars` dataset with a small mutate function included. Then, using this data, we will plot quarter-mile time in seconds against vehicle horsepower, and use the number of engine cylinders as the color scale. We'll also facet the output by transmission style.

```

1 p <- mtcars %>% mutate(
2   transmission = case_when(
3     am == 0 ~ "Automatic",
4     am == 1 ~ "Manual"
5   )
6 ) %>%
7   ggplot(aes(qsec, hp, color = as.factor(cyl))) +
8     geom_point() +
9     labs(
10      title = "Testing a Title Here",
11      subtitle = "Some kind of subtitle, yo",
12      caption = "Also, need to worry about captioning!",
13      x = "Quarter Mile Time in Seconds",
14      y = "Horsepower",
15      color = "Cylinders"
16    ) +
17     facet_grid(~transmission)
18
19 p

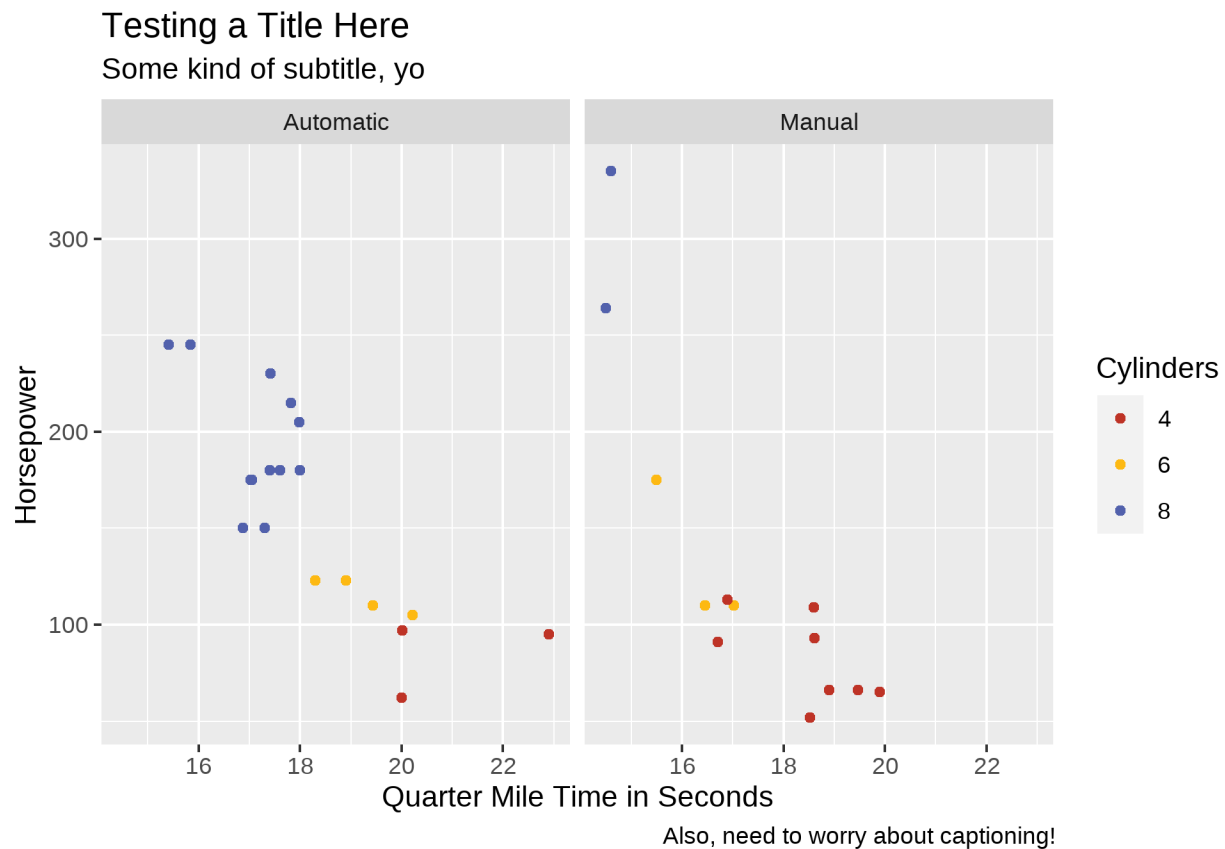
```



*Plot 1. Default ggplot theme and color scale.*

This is the default `ggplot` theme. There are other themes available in the basic `ggplot` package, and even more themes available in `ggthemes`. None of them, however, provided anything which matches REA's and OFDA's typical graphs. So, let's add in `scale_color_mps()` to change the color scale to MPS primary colors.

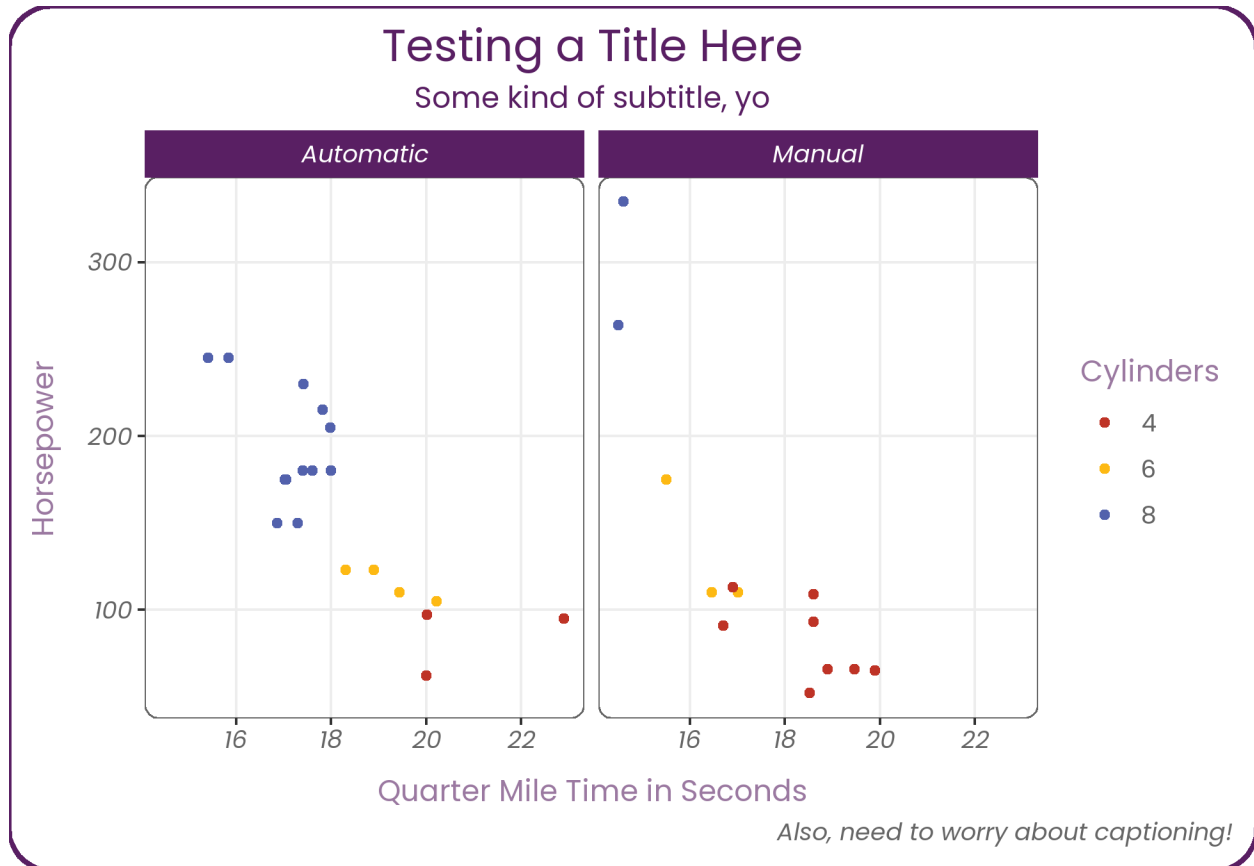
```
1 p + scale_color_mps(palette = "primary", discrete = TRUE)
```



*Plot 2. Default ggplot theme with MPS primary colors.*

This already looks much more like an MPS-style, but it's still very much a basic `ggplot` look. So, let's change that by adding `theme_mps()`. **NOTE** In the setup chunk of your document, you need to have `showtext_auto()` included after loading the `showtext` package.

```
1 p + scale_color_mps(palette = "primary", discrete = TRUE) +
2 theme_mps()
```



*Plot 3. Default MPS theme with MPS primary colors.*

This is much better! So, what exactly is `theme_mps()` doing to the base `ggplot()` theme? Well, see below for all the changes.

```
1 theme(
2   # Set all text-based elements of the theme
3   # Base text definition
4   text = element_text(size = 14, color = pri.color, family = "Poppins"),
5   # Plot titles, subtitles, and captions
6   plot.title = element_text(hjust = 0.5),
7   plot.title.position = "panel",
8   plot.subtitle = element_text(size = 11, hjust = 0.5),
9   plot.caption = element_text(size = 9, face = "italic", color = ter.color, hjust = 1),
10  plot.caption.position = "plot",
11  # Axis titles and text
12  axis.title = element_text(size = 11),
13  axis.text = element_text(size = 9, face = "italic", color = ter.color),
```



```

14 axis.title.y = element_text(margin = margin(r = 10), color = sec.color, angle = 90),
15 axis.title.x = element_text(margin = margin(t = 10), color = sec.color),
16 # Legend titles, text, and formatting
17 legend.title = element_text(size = 11, color = sec.color),
18 legend.text = element_text(size = 9, color = ter.color),
19 # Plot aesthetics
20 plot.background = element_rect(color = pri.color, size = 1, r = .05),
21 plot.margin = margin(.125, .125, .125, .125, unit = "inches"),
22 # Panel aesthetics
23 panel.background = element_rect(fill = "white"),
24 panel.border = element_rect(fill = NA, color = ter.color, r = .03),
25 panel.grid = element_line(color = "gray93", linewidth = 0.5, linetype = 1),
26 panel.grid.minor = element_blank(),
27 # Legend aesthetics
28 legend.key = element_rect(fill = "white"),
29 # Facet aesthetics
30 strip.background = element_rect(fill = pri.color, color = pri.color, r = .03),
31 strip.text = element_text(size = 9, face = "italic", color = "white"),
32 ...
33 )

```