heatmap4 Vignette

Rohan Bhanot

10/31/2019

1. Introduction

This package takes the original heatmap function and reduces the argument complexity.

A heat map is a false color image (basically image(t(x))) with a dendrogram added to the left side and to the top. Typically, reordering of the rows and columns according to some set of values (row or column means) within the restrictions imposed by the dendrogram is carried out.

2. Loading Libraries

The libraries to call are the following:

```r
library(RColorBrewer)
library(marray)
```

```
## Loading required package: limma
```

3. Example Data

We will use the preloaded mtcars data set to map and randomly generate column and row annotations.

```
# Main numeric matrix to heatmap
test_x <- as.matrix(mtcars)

# Column annotation df
set.seed(827)

col_rows <- ncol(mtcars)
col_columns <- 3

column_data <- rnorm(col_rows * col_columns)
column_df <- matrix(data = column_data, nrow = col_rows, ncol = col_columns)
colnames(column_df) <- c('C1', 'C2', 'C3')
column_df <- as.data.frame(column_df)

random3 <- sample(1:3, col_rows, replace = TRUE)
column_df$C4 <- random3

# Row annotation df
set.seed(957)

row_rows <- nrow(mtcars)
row_columns <- 4

row_data <- rnorm(row_rows * row_columns)
row_df <- matrix(data = row_data, nrow = row_rows, ncol = row_columns)
colnames(row_df) <- c('R1', 'R2', 'R3', 'R4')
row_df <- as.data.frame(row_df)
```

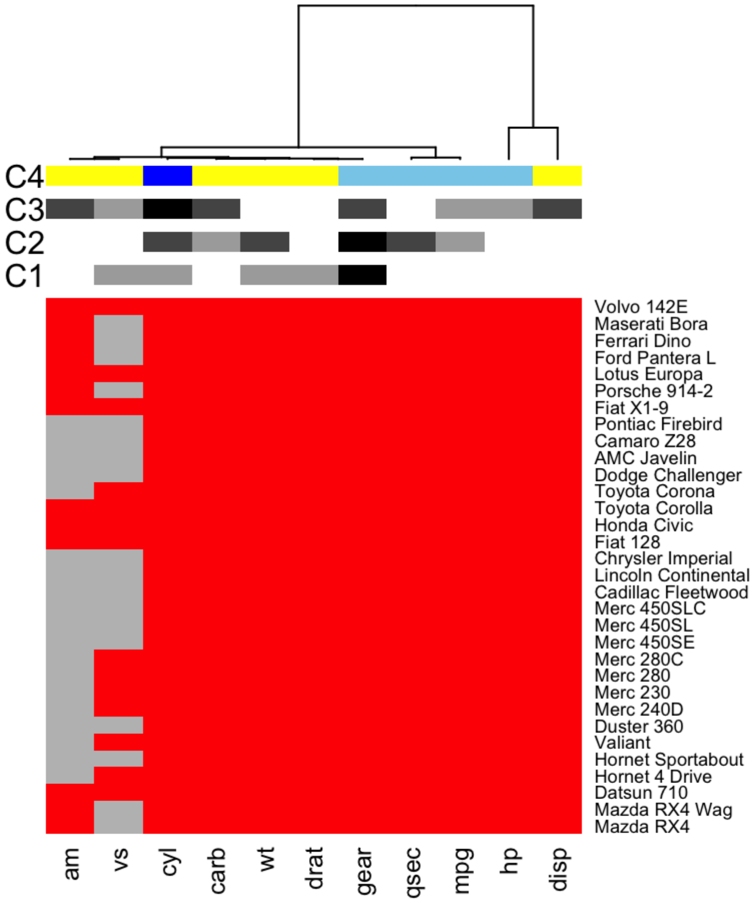
4. Use Cases

> **NOTE**: This is the simplest way to call the heatmap4 function. Defaults are accounted for, see documentation for full list of defualt options for each arguement.

4.1 Annotations and Dendrograms

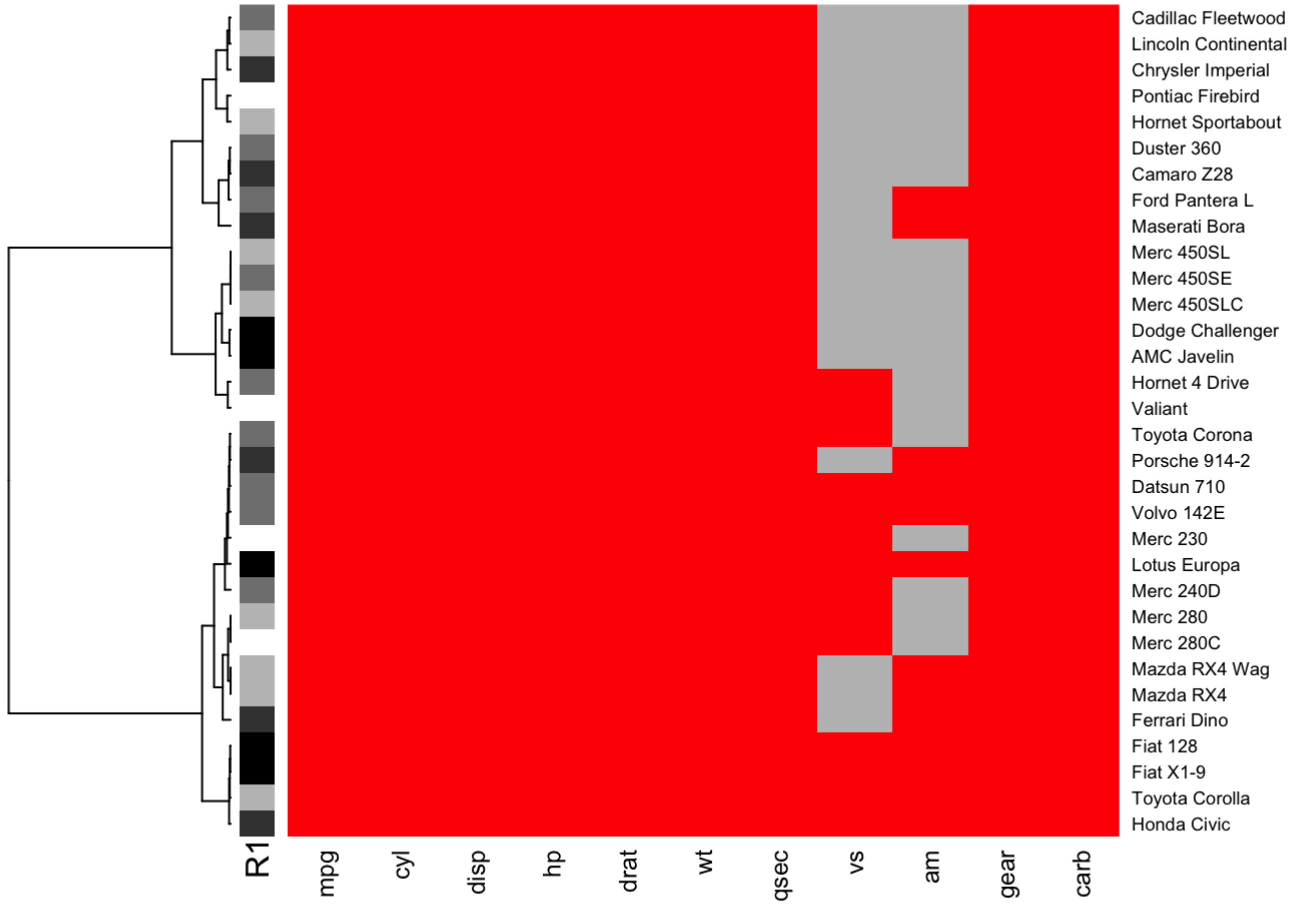
To display the heatmap with just column annotations and dendrograms:

```
generate_heatmap(test_x, row_anno = FALSE, col_info = column_df,  row_dend = FALSE)
```



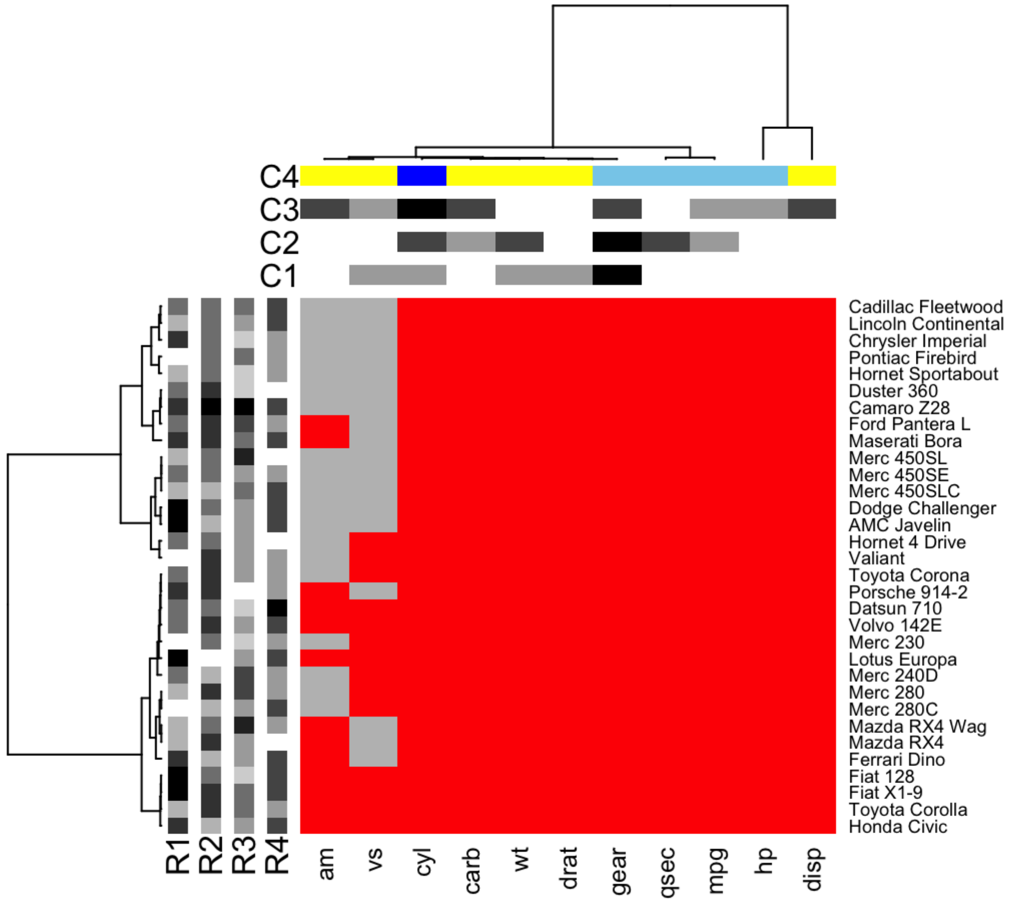
The opposite display of just row annotations and dendrograms:

```
# Needs to be fixed to allow for more than one row annotation -- margins error
generate_heatmap(test_x, col_anno = FALSE, row_info = row_df, col_dend = FALSE, row_anno
_var = c("R1"))
```



To display heatmap with *both* column and row annotations and dendrograms:

```
generate_heatmap(test_x, col_info = column_df, row_info = row_df)
```



If the column or row annotation matrixes contain too many columns to display or there is a need to select only a few, the attribute `col_anno_var` (for column annotation limitations) or `row_anno_var` (for row annotation limitations).

The column names for the column annotaion data.frame is:

```
## [1] "C1" "C2" "C3" "C4"
```

The column names for the row annotaion data.frame is:

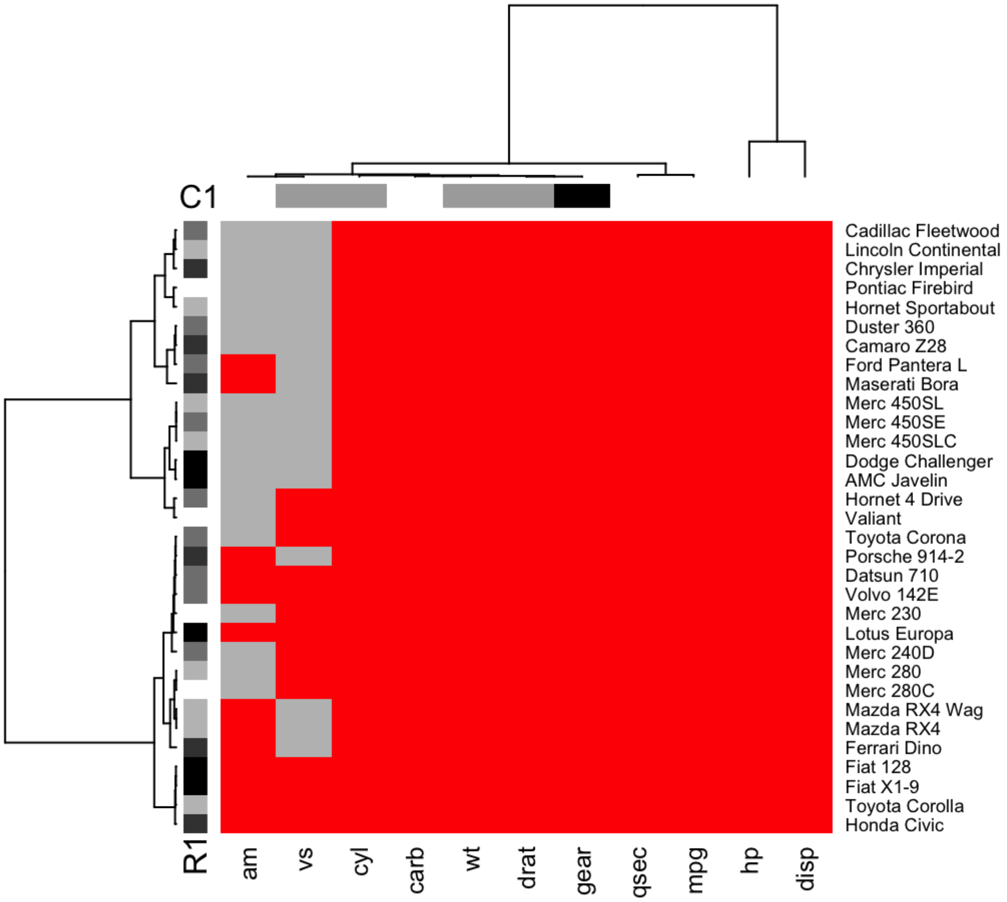
```
## [1] "R1" "R2" "R3" "R4"
```

If we only want to display the first column in each data.frame, we would set:

- `col_anno_var = colnames(column_df)[1]`
- `row_anno_var = colnames(row_df)[1]`

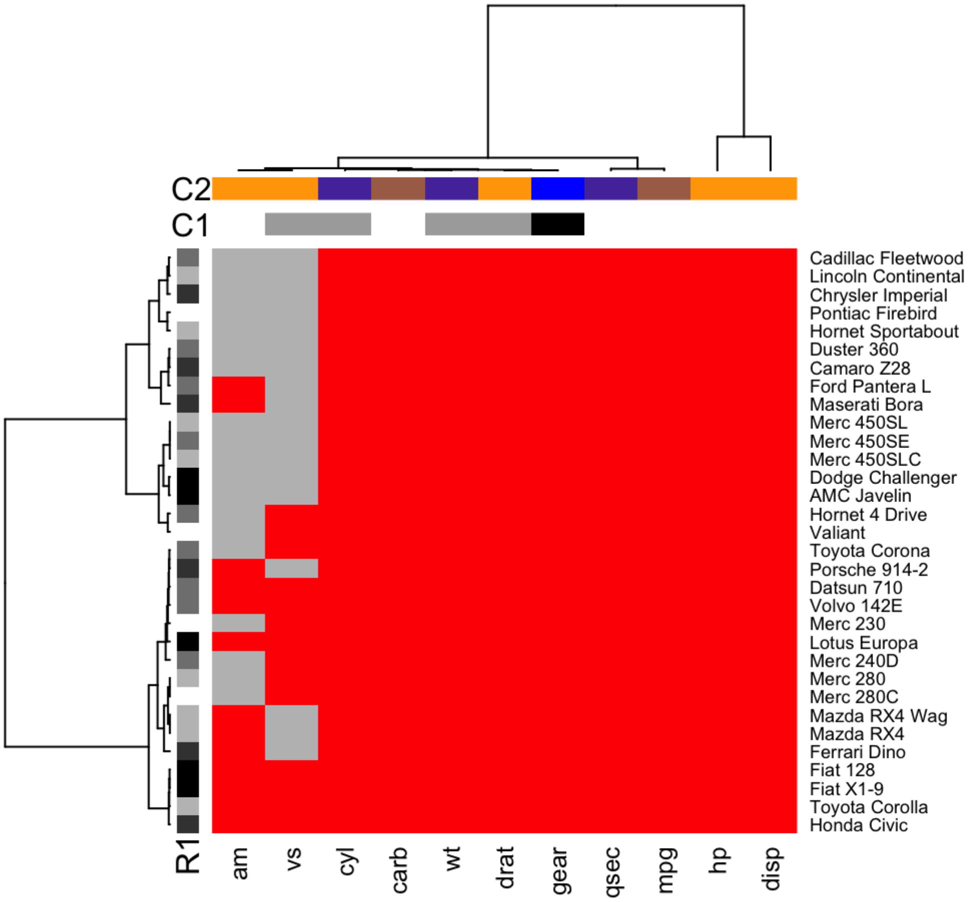
OR

```
generate_heatmap(test_x, col_info = column_df, row_info = row_df, col_anno_var = c("C1"
), row_anno_var = c("R1"))
```



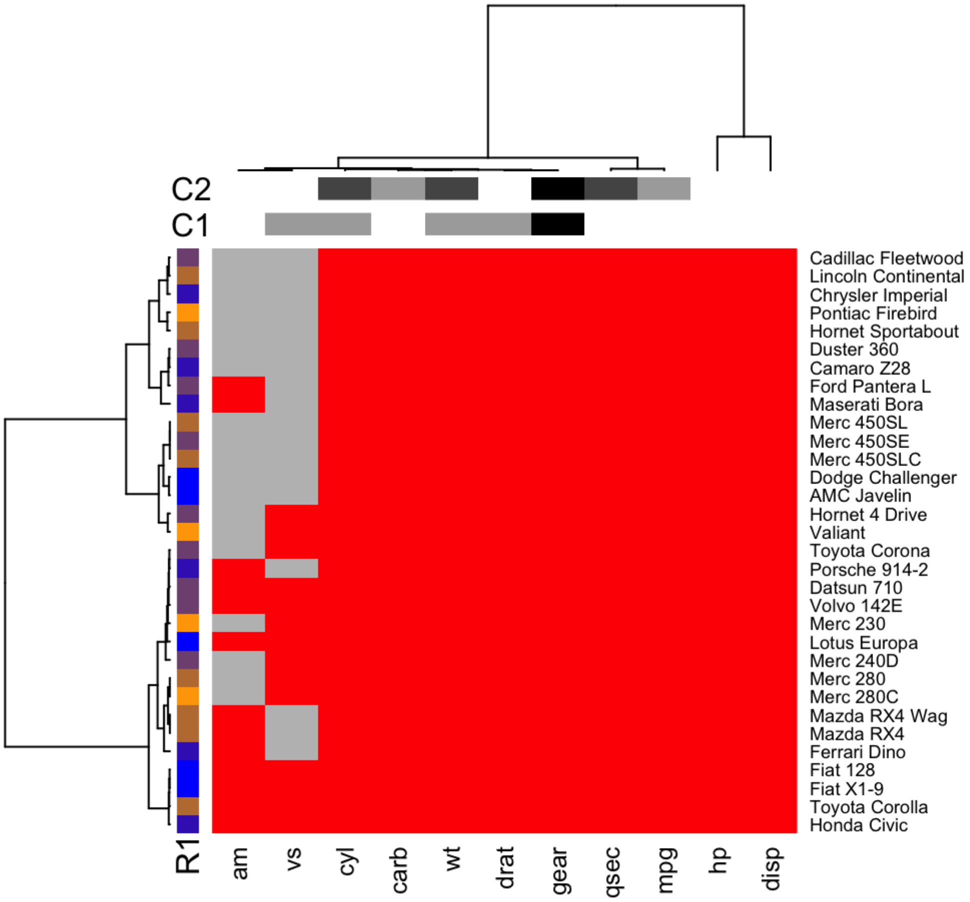
To get even more specific, there is the ability to alter annotation specifications, such as color. As an example, we will isolate the `"C2"` column in `column_df` (column annotation data.frame) and change the color scale from `"white", "black"` to `"orange", "blue"`. When specifying the color of a continuous variable, please note that the first color will represent the “lows” and the second color represents the “highs”.

```
change <- list(C2 = list(color = c("orange", "blue")))

generate_heatmap(test_x, col_info = column_df, row_info = row_df, col_anno_var = c("C1",
"C2"), row_anno_var = c("R1"), col_var_info = change)
```



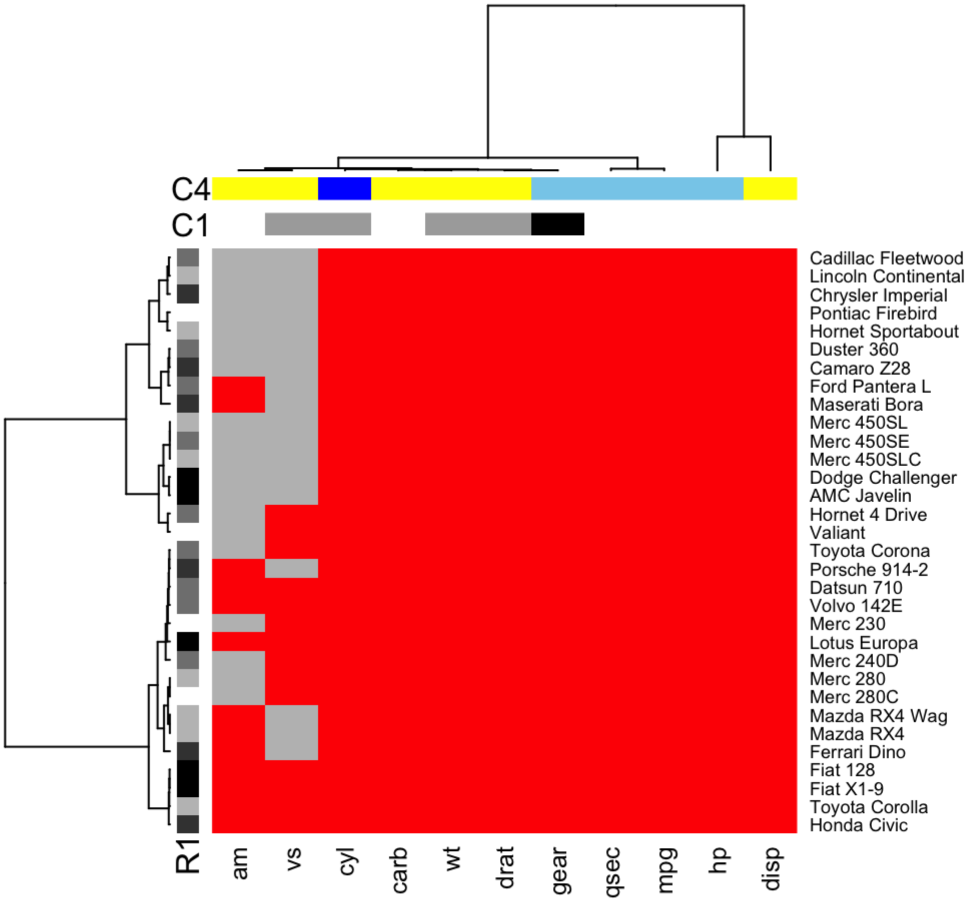
The same can be done with row annotations:

```
change <- list(R1 = list(color = c("orange", "blue")))

generate_heatmap(test_x, col_info = column_df, row_info = row_df, col_anno_var = c("C1",
"C2"), row_anno_var = c("R1"), row_var_info = change)
```



If the annotation is not continuous, instead, categorical (like column annotation C4), It will be displayed with distinct colors for each category, the default being: `column_df$C1` versus `column_df$C4`

```
generate_heatmap(test_x, col_info = column_df, row_info = row_df, col_anno_var = c("C1",
"C4"), row_anno_var = c("R1"))
```

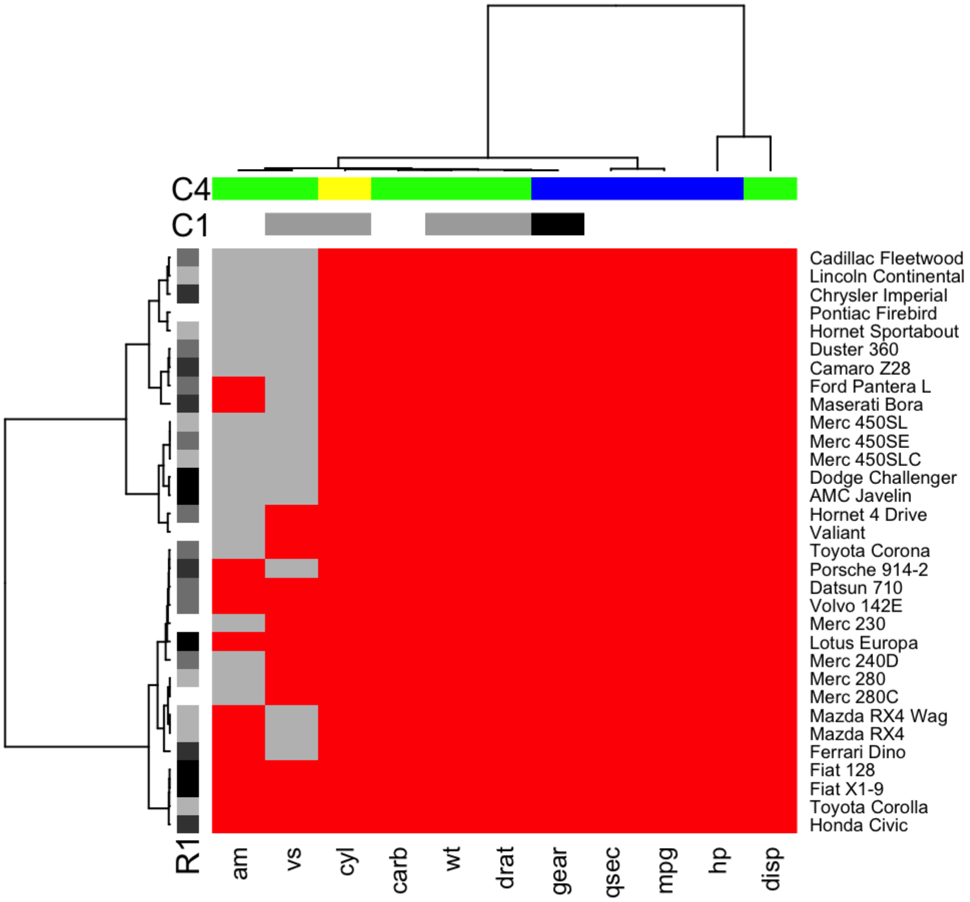


It is also possible to change the color of categorical variables. The order of the colors will match the order of the variables.

```
# Color matches this order
print(unique(column_df$C4))
```

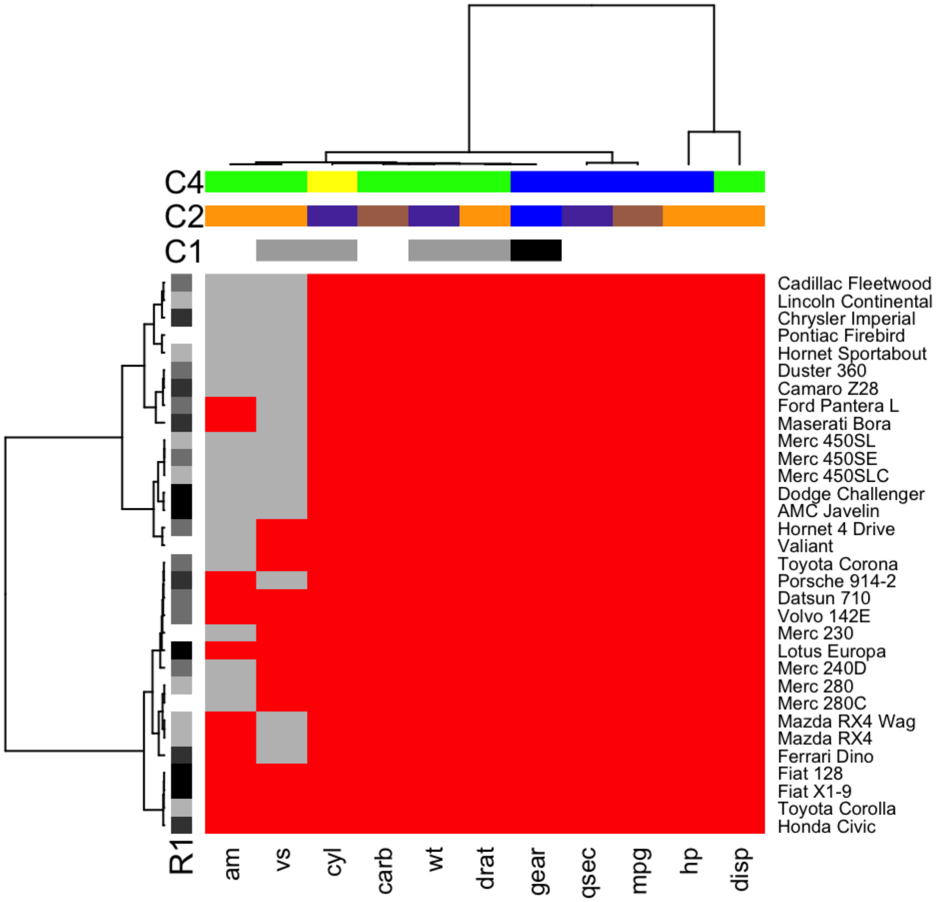
```
## [1] 1 2 3
```

```
categorical_color <- list(C4 = list(color = c("blue", "yellow", "green")))

generate_heatmap(test_x, col_info = column_df, row_info = row_df, col_anno_var = c("C1",
"C4"), row_anno_var = c("R1"), col_var_info = categorical_color)
```



To change the color of both a categorical and continuous variable: In this example, C4 is categorical and C1, C2, R1 are continuous. Only C1 and C4 have been altered.

```
color_change <- list(C4 = list(color = c("blue", "yellow", "green")), C2 = list(color =
 c("orange", "blue")))

generate_heatmap(test_x, col_info = column_df, row_info = row_df, col_anno_var = c("C1",
"C2", "C4"), row_anno_var = c("R1"), col_var_info = color_change)
```

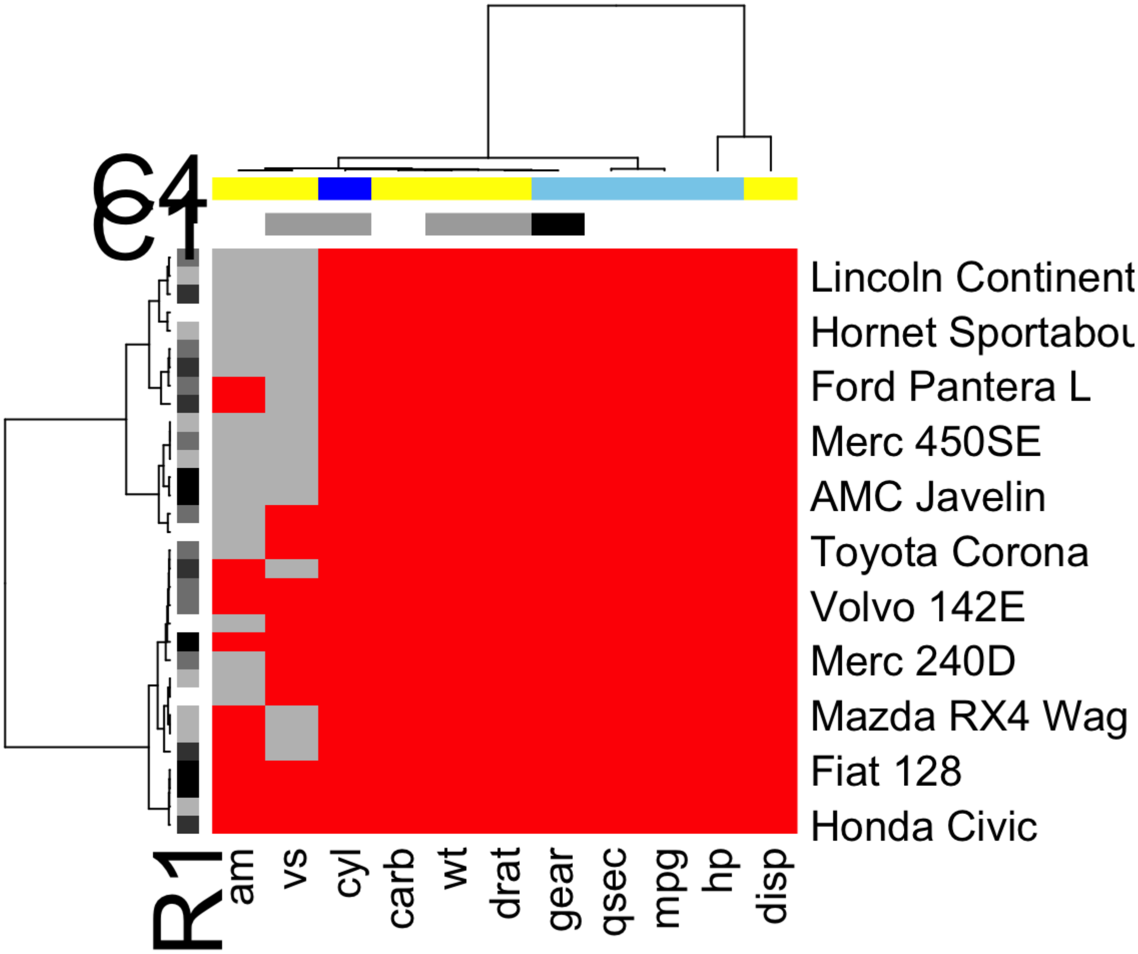


4.2 Main Plot

The heatmap will do its best to account for the size of the plot and the labels associated with it. However, there is an option for the user to adjust the main plot's column and row labels as well as column and row annotation labels.

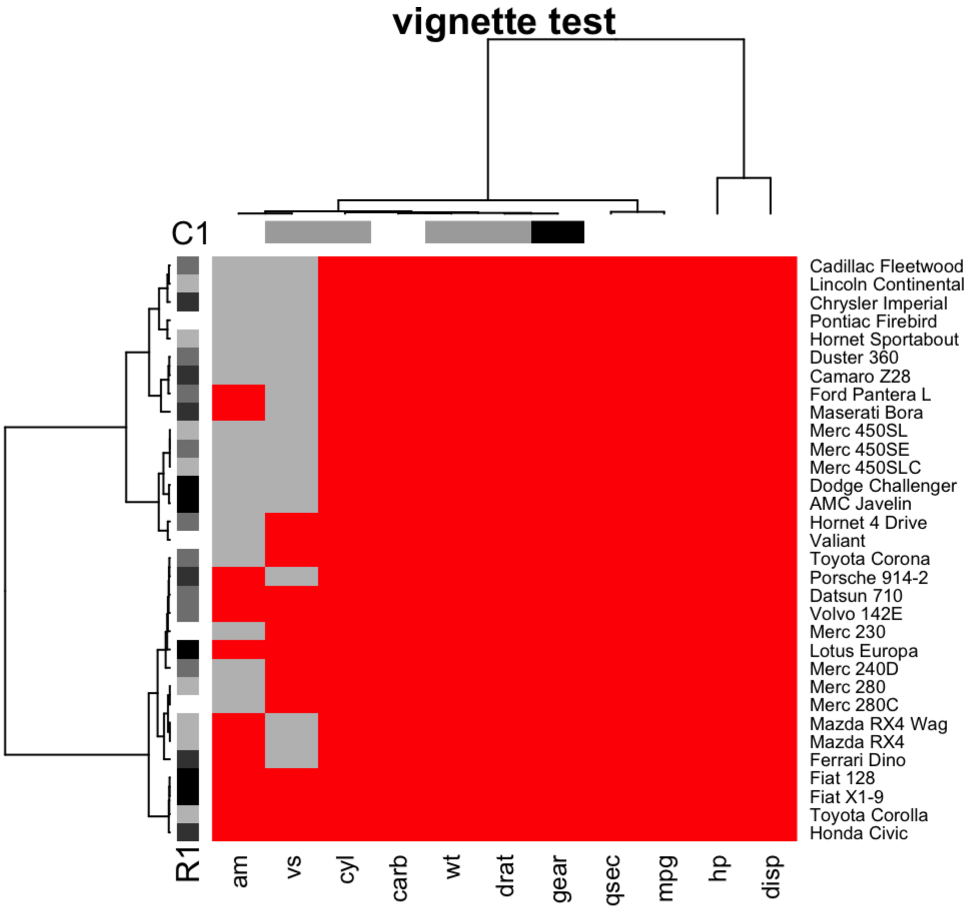
- `sideLabCol and sideLabRow` denote heatmap labels
- `cexColSide and cexRowSide` denote annotation labels

```
main_plot = list("cexCol" = 2, "cexRow" = 2, "cexRowSide" = 3, "cexColSide" = 3)

generate_heatmap(test_x, col_info = column_df, row_info = row_df, col_anno_var = c("C1",
"C4"), row_anno_var = c("R1"), plot_info = main_plot)
```



To add a title to the heatmap, simply call upon `h_title`

```
generate_heatmap(test_x, col_info = column_df, row_info = row_df, col_anno_var = c("C1"
), row_anno_var = c("R1"), h_title = "vignette test")
```



vignette test