Introduction to Interactive Plots

Osama Mahmoud

Website: http://osmahmoud.com
E-mail: o.mahmoud@bristol.ac.uk

15 Novemvber 2019

Goals

In this sesseion, we will:

- Introduce interactive plots in R.
- Show how to use `plotly` in R to produce basic interactive graphics.
- Show useful extension of `ggplot2` and `plotly` for visualising correlation matrices.
- Demonstrate how to save your interactive plots in html format.

Contents

Install and load required packages

Installing packages in R is straightforward. For example:

```
> install.packages(c("ggcorrplot", "plotly"))
```

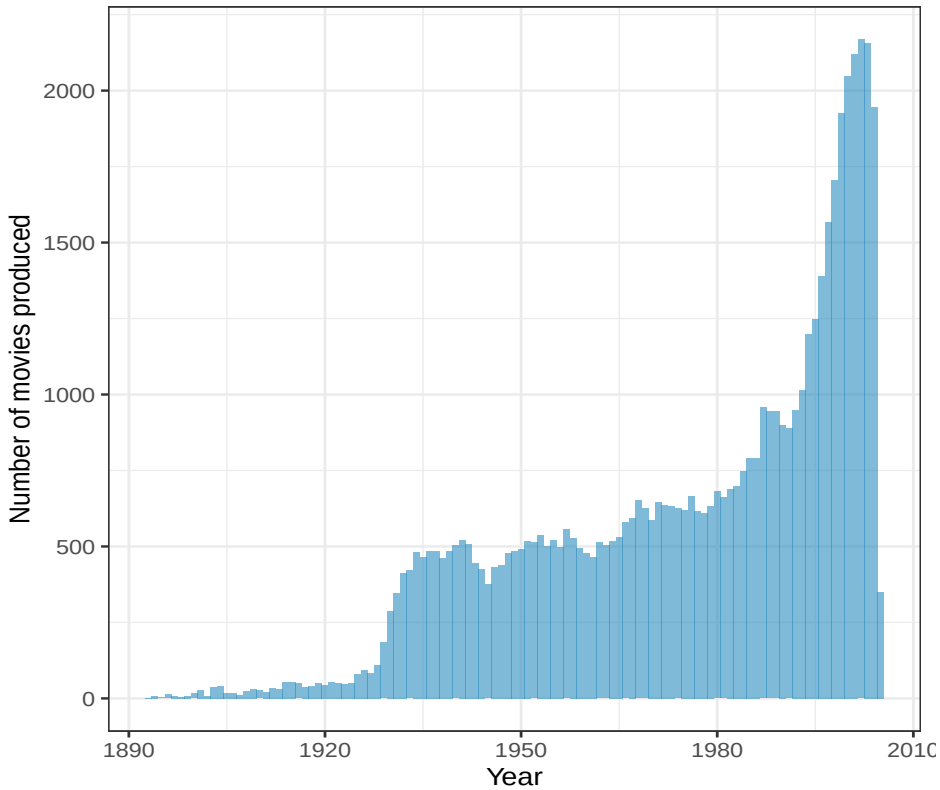
Then, you can simply load it to your R session whenever needed:

```
> library("plotly"); library("plotly")
```

Produce interactive plots using ggplotly

Histograms

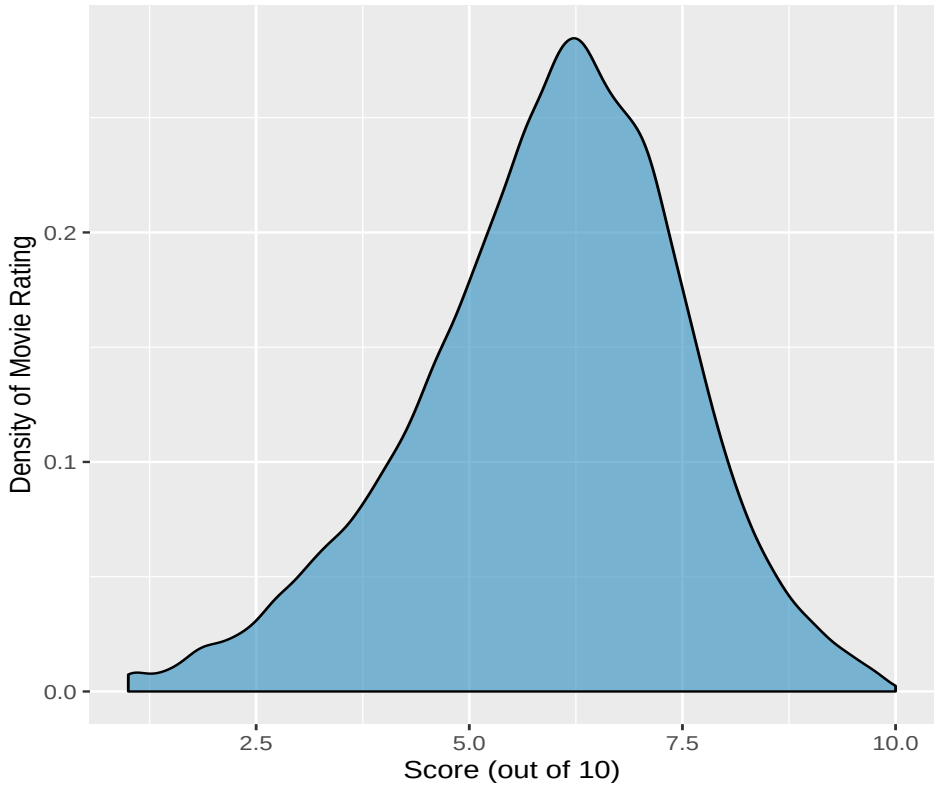
```
> g = ggplot(data=movies, aes(x=year)) + geom_histogram(binwidth = 1,
fill="#2b8cbe", alpha=0.6)
> g + xlab("Year") + ylab("Number of movies produced") + theme_bw()
```




```
> ggplotly(g)
```

Density

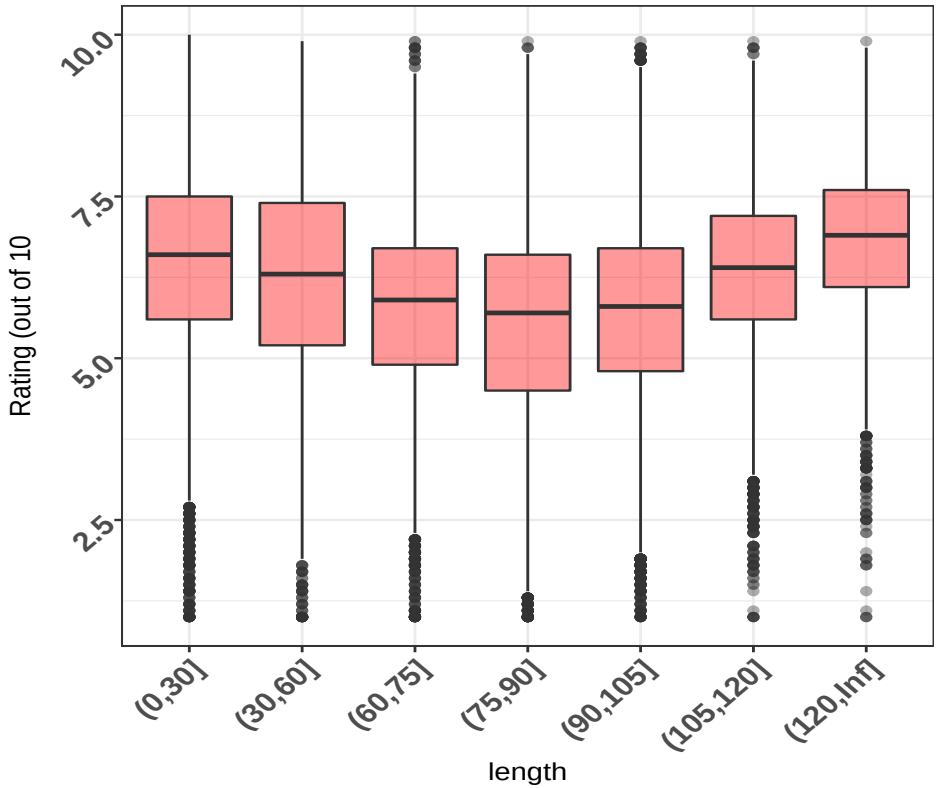
```
> g = ggplot(movies, aes(x=rating)) + geom_density(fill="#2b8cbe",
alpha=0.6)
> g + ylab("Density of Movie Rating") + xlab("Score (out of 10)")
```




```
> ggplotly(g)
```

box-plot

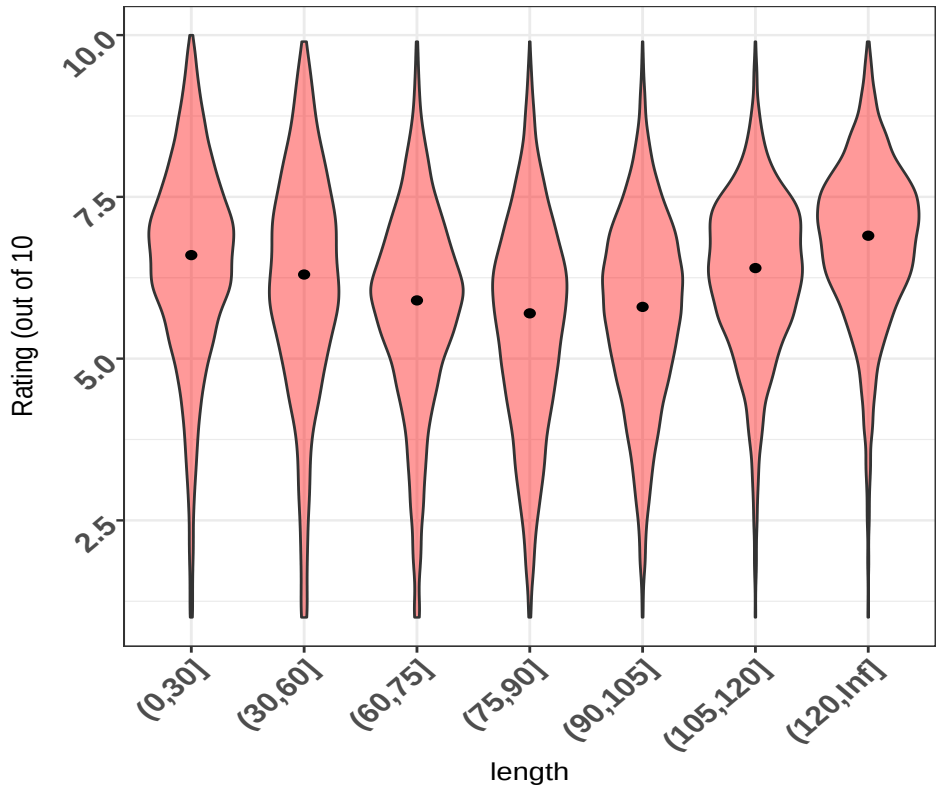
```
> g = ggplot(movies, aes(x=factor(cat_length), y=rating)) +
xlab("length") + ylab("Rating (out of 10")
> g + geom_boxplot(fill="red", alpha=0.4)
```




```
> ggplotly(g)
```

violin

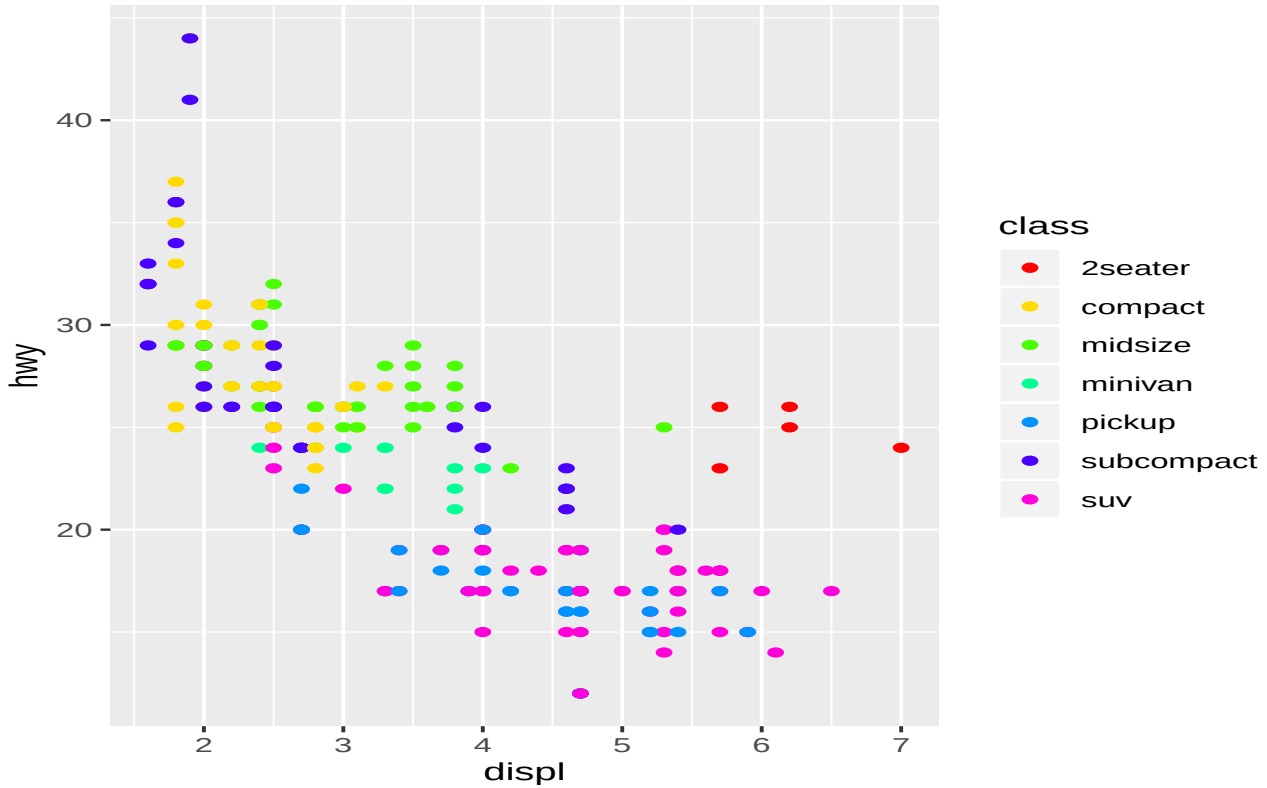
```
> g = g + geom_violin(fill="red", alpha=0.4) + stat_summary(fun.y = median, geom='point')
> g + theme_bw() + theme(axis.text=element_text(face='bold', size = 12, angle = 45, hjust = 1))
```




```
> ggplotly(g)
```

Scatter plot

```
> data(mpg, package = "ggplot2")
> g = ggplot(mpg, aes(displ, hwy, colour=class)) + geom_point()
> g + scale_color_manual(values=rainbow(7))
```




```
> Fig = ggplotly(g)
```

Saving interactive plots

```
> require(htmlwidgets)
> htmlwidgets::saveWidget(widget = Fig, "Scatter_plot.html")
```

Prepare your data

Compute the correlation matrix

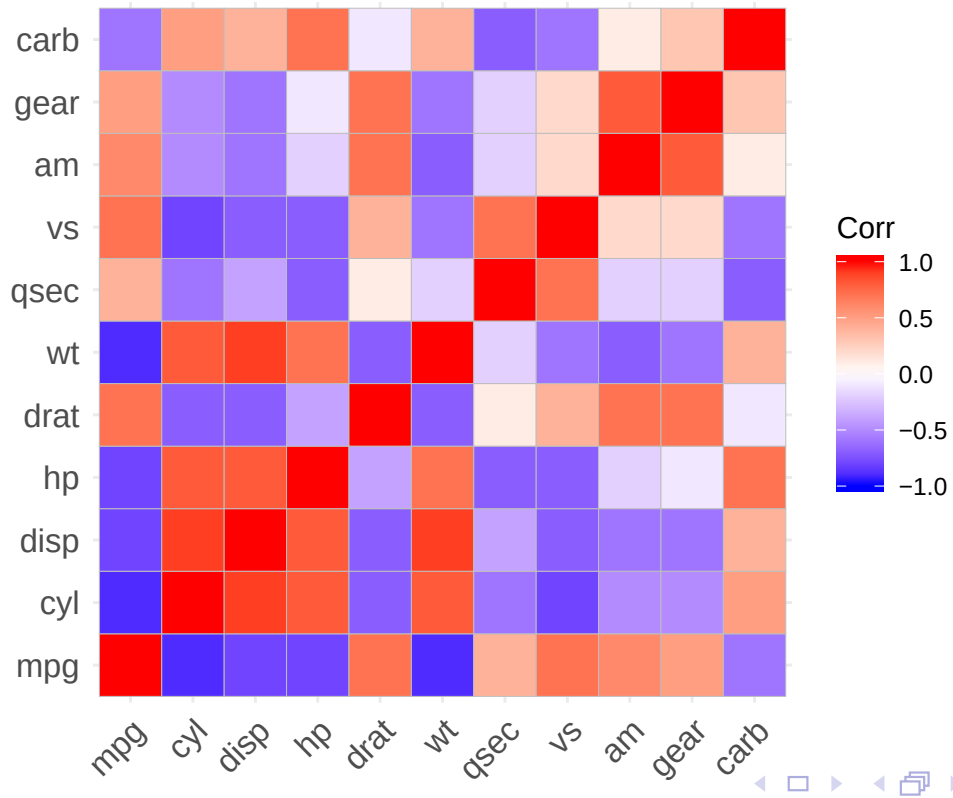
```
> data(mtcars); corr.mat <- round(cor(mtcars), 1)
```

Compute correlation P-values

```
> pval.cor <- cor_pmat(mtcars)
```

Visualize the correlation matrix: method = "square" (default)

```
> (G1 = ggcorrplot(corr.mat))
```

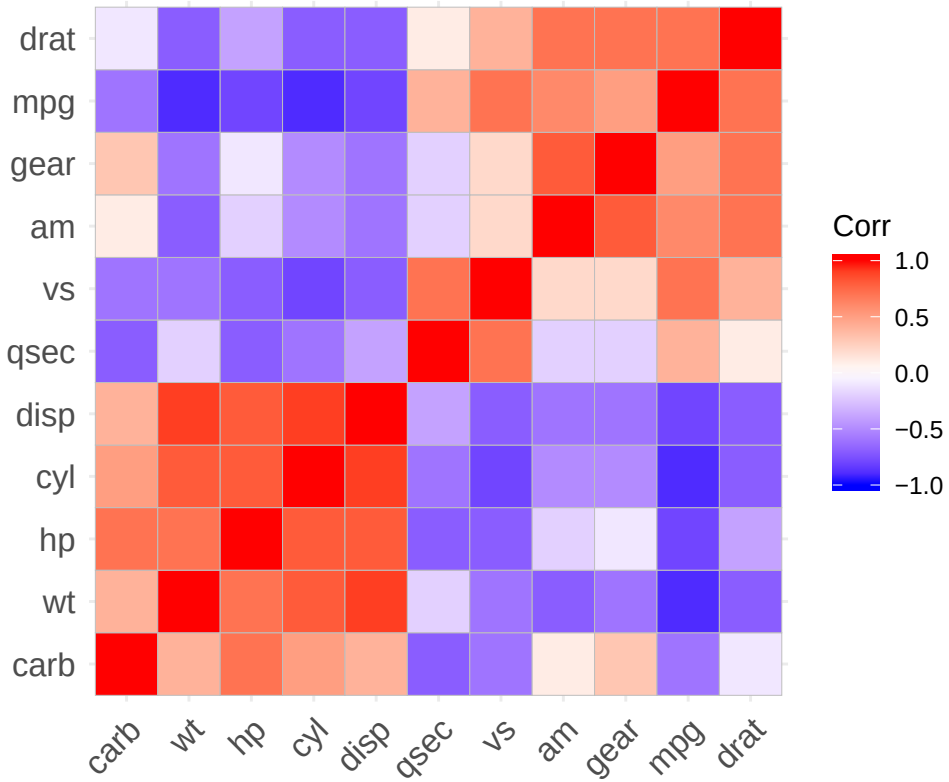


Reordering the correlation matrix using hierarchical clustering

```
> G2 = ggcorrplot(corr.mat, hc.order = TRUE)
```

Interactive form

```
> ggplotly(G2)
```

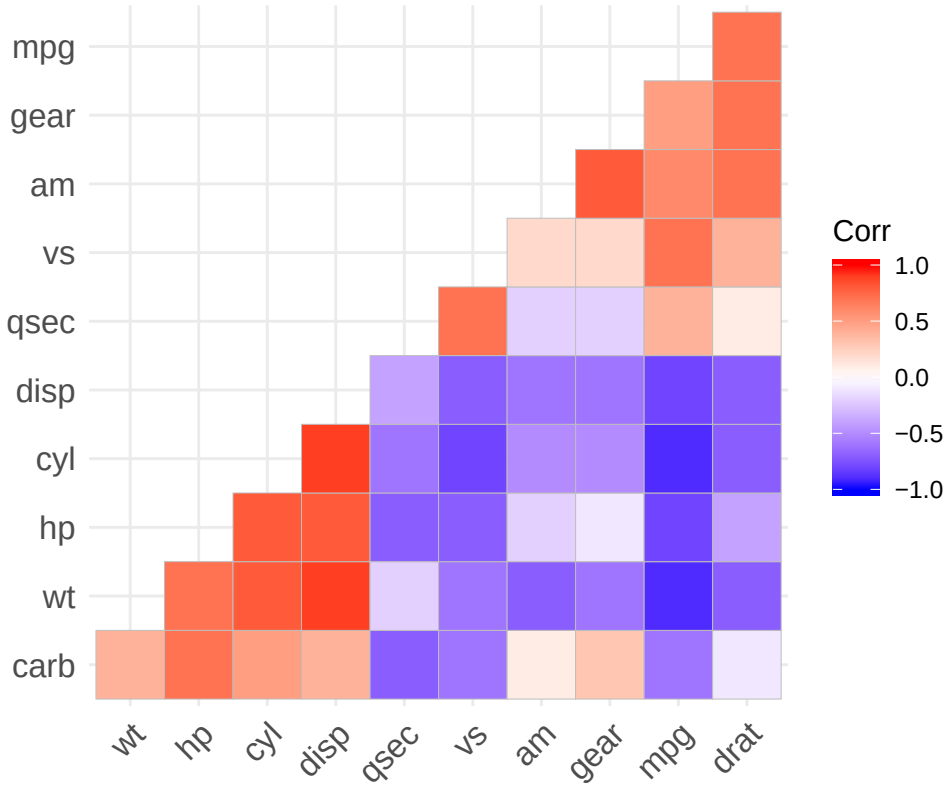


Correlation matrix: Lower triangle

```
> G3 = ggcorrplot(corr.mat, hc.order = TRUE, type = "lower")
```

Interactive form

```
> ggplotly(G3)
```

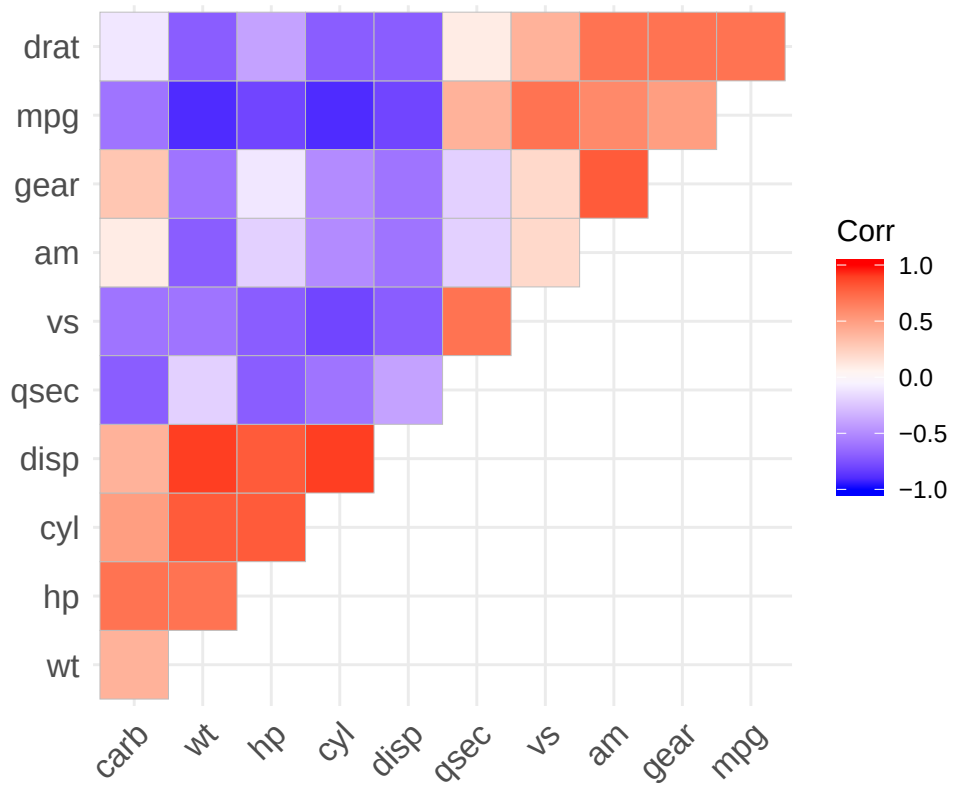


Correlation matrix: Upper triangle

```
> G4 = ggcorrplot(corr.mat, hc.order = TRUE, type = "upper")
```

Interactive form

```
> ggplotly(G4)
```

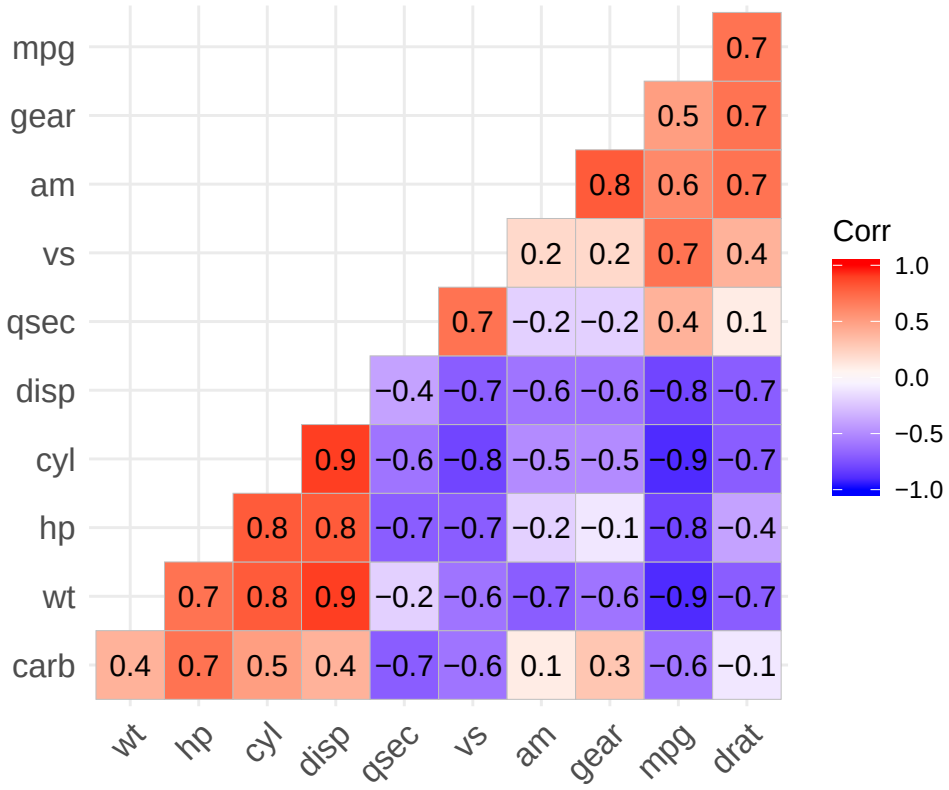


Correlation matrix: Add coeficients

```
> G5 = ggcorrplot(corr.mat, hc.order = TRUE, type = "lower", lab = TRUE)
```

Interactive form

```
> ggplotly(G5)
```

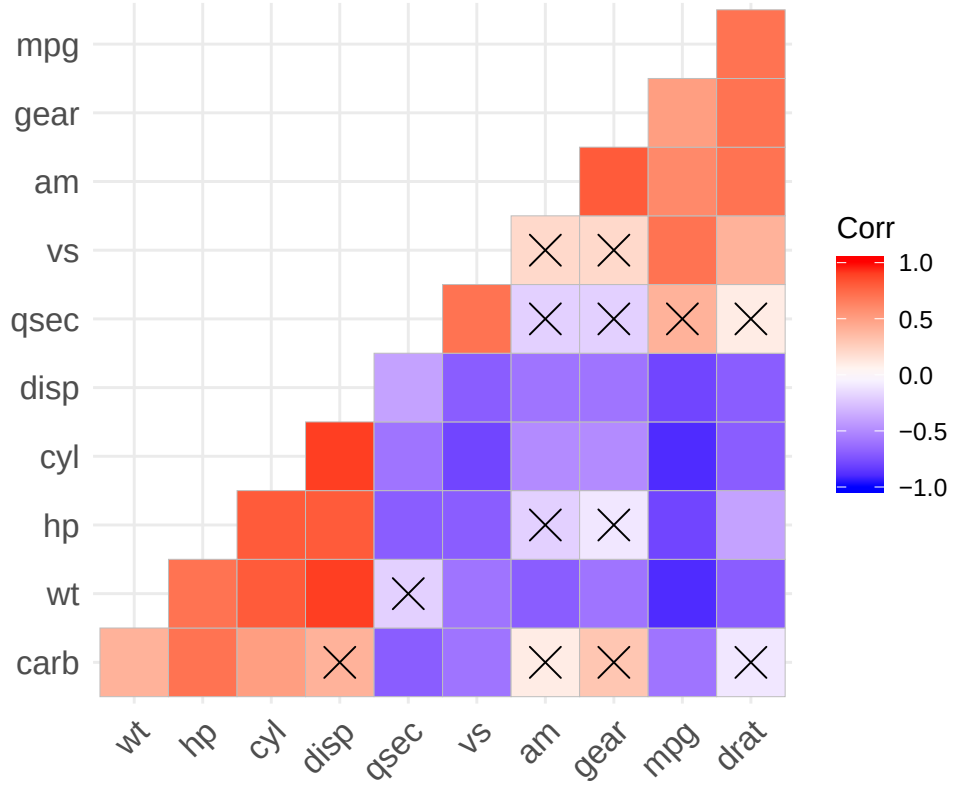


Correlation matrix: Add significance level

```
> G6 = ggcorrplot(corr.mat, hc.order = TRUE, type = "lower", p.mat =
pval.cor, sig.level = 0.01, insig = "pch", pch = 4, pch.col =
"black")
```

Interactive form

```
> ggplotly(G6)
```

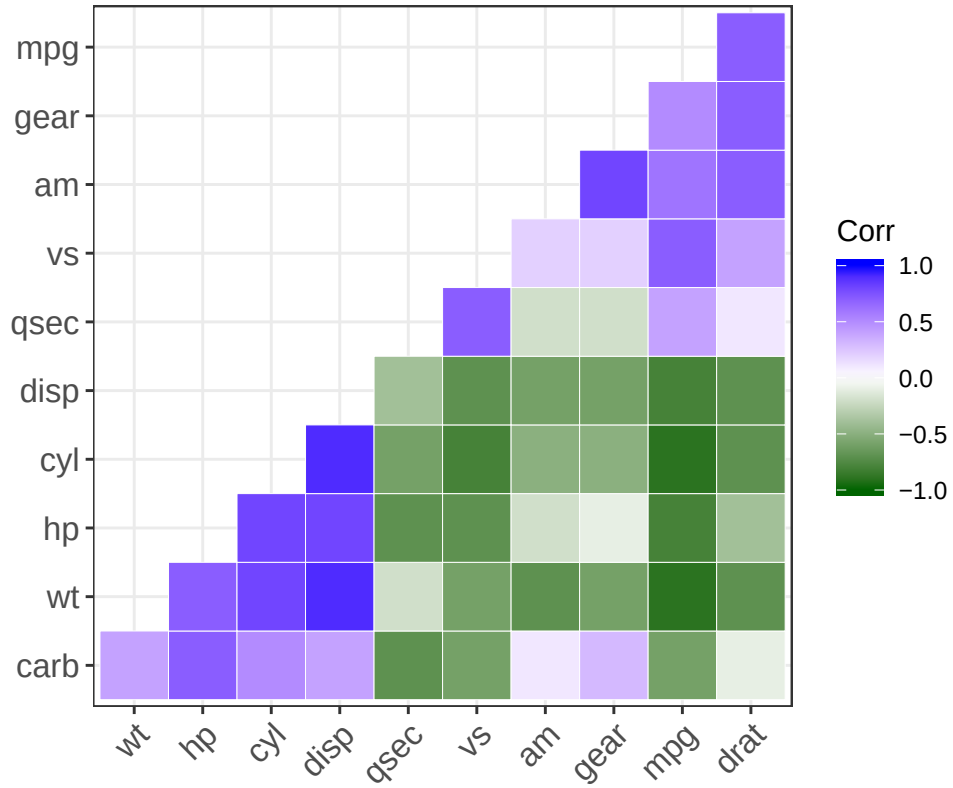


Correlation matrix: Change theme and save plot

```
> G7 = ggcorrplot(corr.mat, hc.order = TRUE, type = "lower",
outline.col = "white", ggtheme = ggplot2::theme_bw(), colors =
c("darkgreen", "white", "blue"))
```

Interactive form

```
> G7 = ggplotly(G7)
```



Saving interactive plots

Save the previous plot (Interactive form)

```
> htmlwidgets::saveWidget(widget = G7, "corr.html")
```

Thank You