

# Other Graphs

David Asare Kumi

4/3/2020

## Correlation Matrix

- Here we are not going to focus on data that is categorical.

```
library(datasets)
data(mtcars)
head(mtcars)
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360	175	3.15	3.440	17.02	0	0	3	2
Valiant	18.1	6	225	105	2.76	3.460	20.22	1	0	3	1

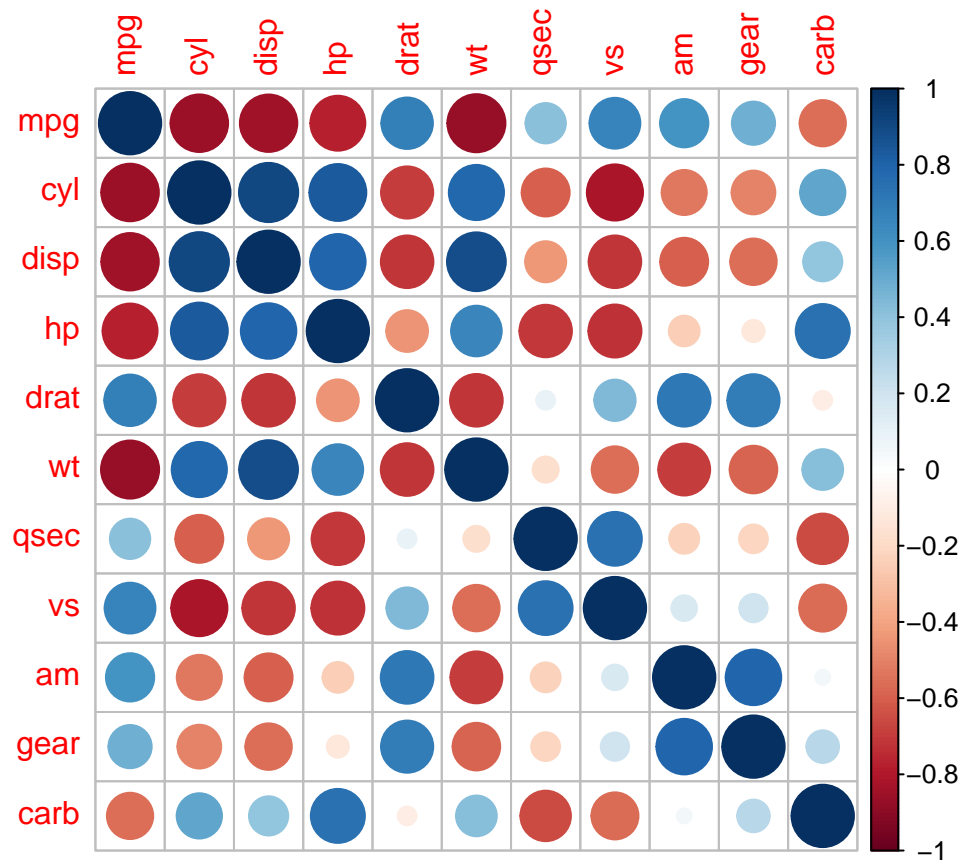
```
mt_cor<-cor(mtcars)
round(mt_cor,2)
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
mpg	1.00	-0.85	-0.85	-0.78	0.68	-0.87	0.42	0.66	0.60	0.48	-0.55
cyl	-0.85	1.00	0.90	0.83	-0.70	0.78	-0.59	-0.81	-0.52	-0.49	0.53
disp	-0.85	0.90	1.00	0.79	-0.71	0.89	-0.43	-0.71	-0.59	-0.56	0.39
hp	-0.78	0.83	0.79	1.00	-0.45	0.66	-0.71	-0.72	-0.24	-0.13	0.75
drat	0.68	-0.70	-0.71	-0.45	1.00	-0.71	0.09	0.44	0.71	0.70	-0.09
wt	-0.87	0.78	0.89	0.66	-0.71	1.00	-0.17	-0.55	-0.69	-0.58	0.43
qsec	0.42	-0.59	-0.43	-0.71	0.09	-0.17	1.00	0.74	-0.23	-0.21	-0.66
vs	0.66	-0.81	-0.71	-0.72	0.44	-0.55	0.74	1.00	0.17	0.21	-0.57
am	0.60	-0.52	-0.59	-0.24	0.71	-0.69	-0.23	0.17	1.00	0.79	0.06
gear	0.48	-0.49	-0.56	-0.13	0.70	-0.58	-0.21	0.21	0.79	1.00	0.27
carb	-0.55	0.53	0.39	0.75	-0.09	0.43	-0.66	-0.57	0.06	0.27	1.00

## Graph the correlation matrix

```
library(datasets)
library(corrplot)
data(mtcars)

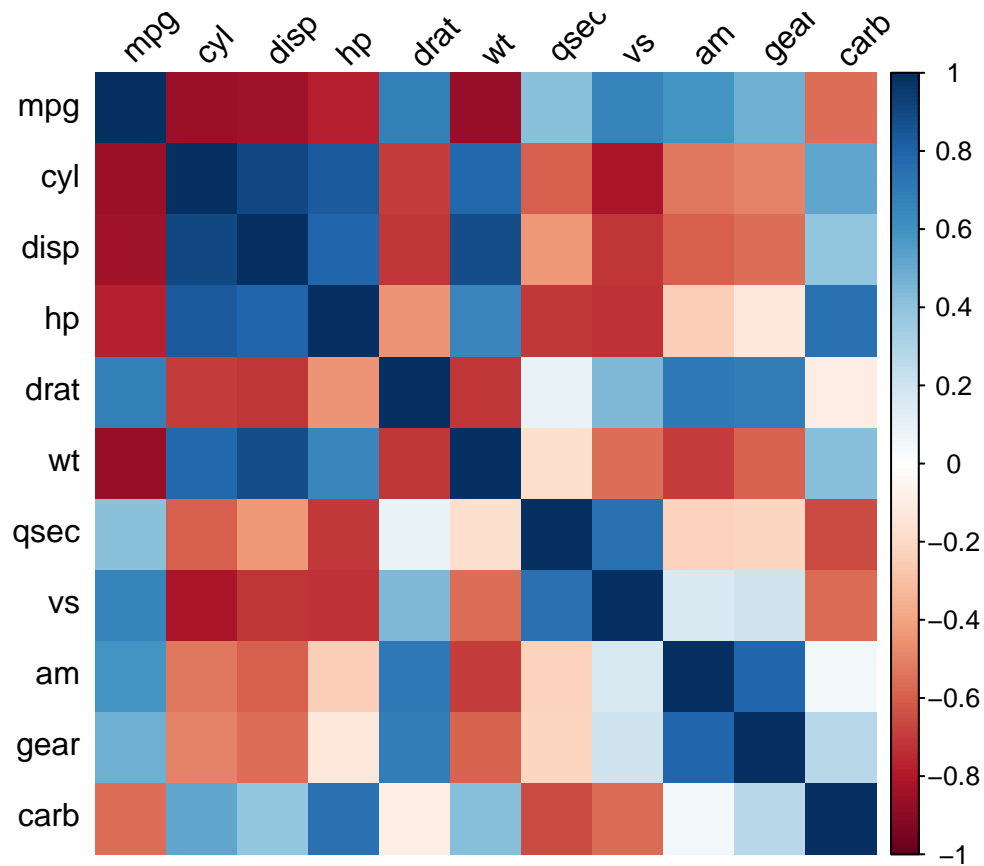
corrplot(mt_cor)
```



Some options of the `corrplot()` function

```
library(datasets)
library(corrplot)
data(mtcars)

corrplot(mt_cor,method="shade",shade.col = NA,tl.col = "black",tl.srt = 45)
```

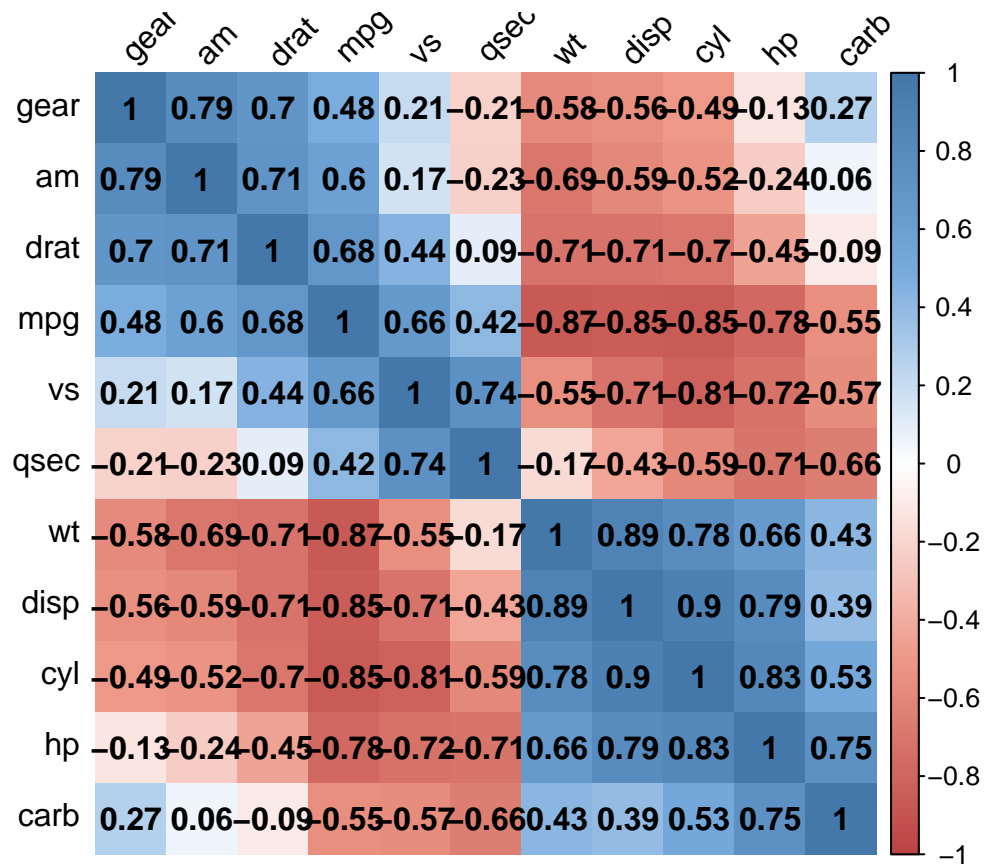


## Display tables representing the correlation

```
library(datasets)
library(corrplot)
data(mtcars)

# Generate a lighter palette
col <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD", "#4477AA"))

corrplot(mt_cor, method="shade", shade.col=NA, tl.col="black", tl.srt=45,
col=col(200), addCoef.col="black", addcolorlabel="no", order="AOE")
```

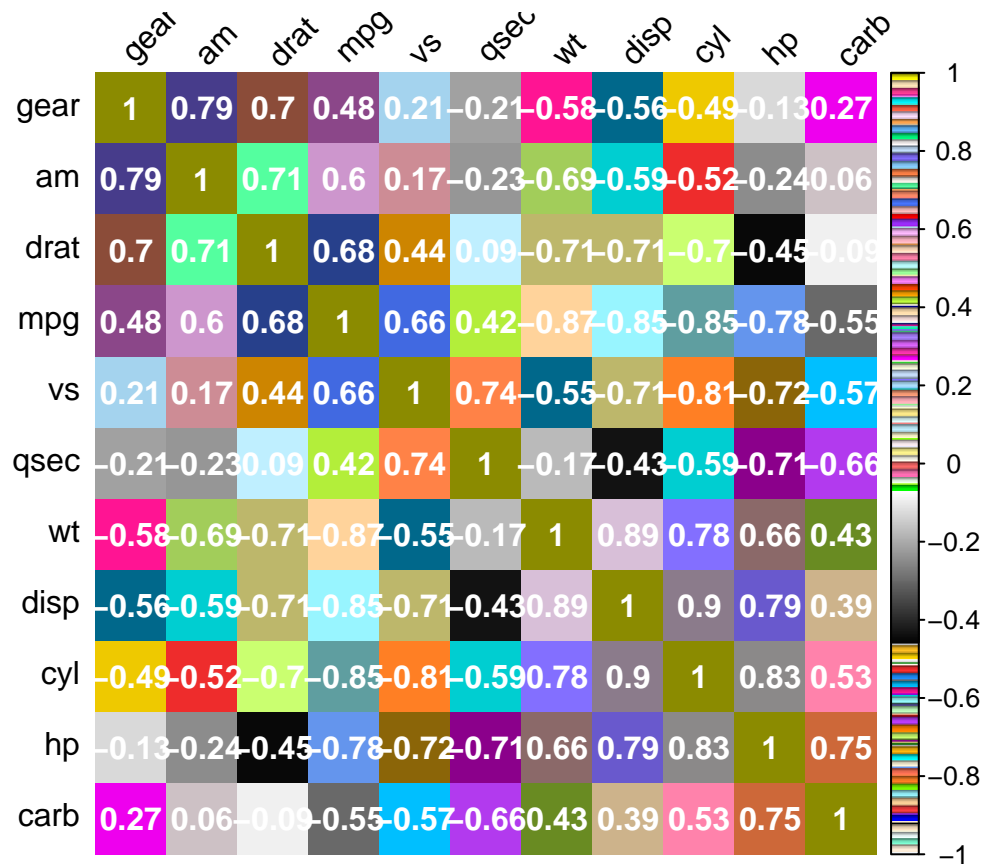


Display tables representing the correlation(alternative)

```
library(datasets)
library(corrplot)
data(mtcars)

# Generate a lighter palette
colors <- c("lightblue", "orange", "pink", "yellow", "yellowgreen", "white", "red", "maroon", "fuchsia",
            "aqua")

corrplot(mt_cor, method="shade", shade.col=NA, tl.col="black", tl.srt=45,
         col=colors(200), addCoef.col="white", addcolorlabel="no", order="AOE")
```

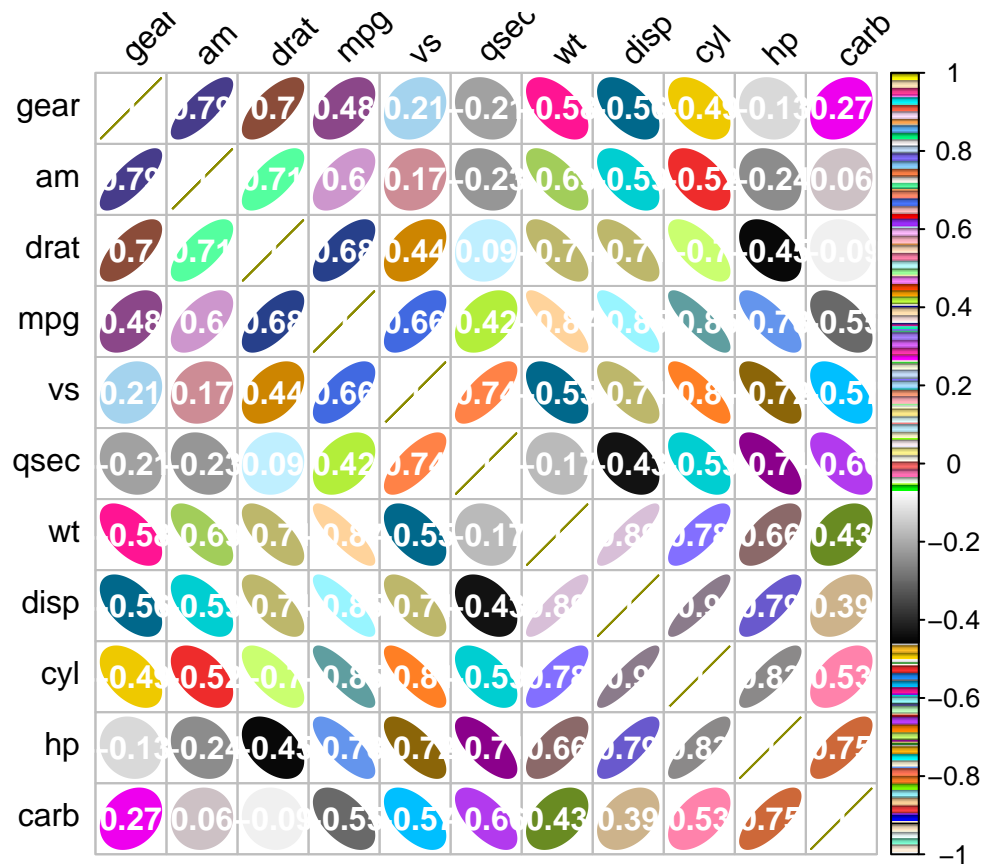


Display tables representing the correlation(alternative with method=ellipse)

```
library(datasets)
library(corrplot)
data(mtcars)

# Generate a lighter palette
colors <- c("lightblue", "orange", "pink", "yellow", "yellowgreen", "white", "red", "maroon", "fuchsia",
            "aqua")

corrplot(mt_cor, method="ellipse", shade.col=NA, tl.col="black", tl.srt=45,
         col=colors(200), addCoef.col="white", addcolorlabel="no", order="AOE")
```

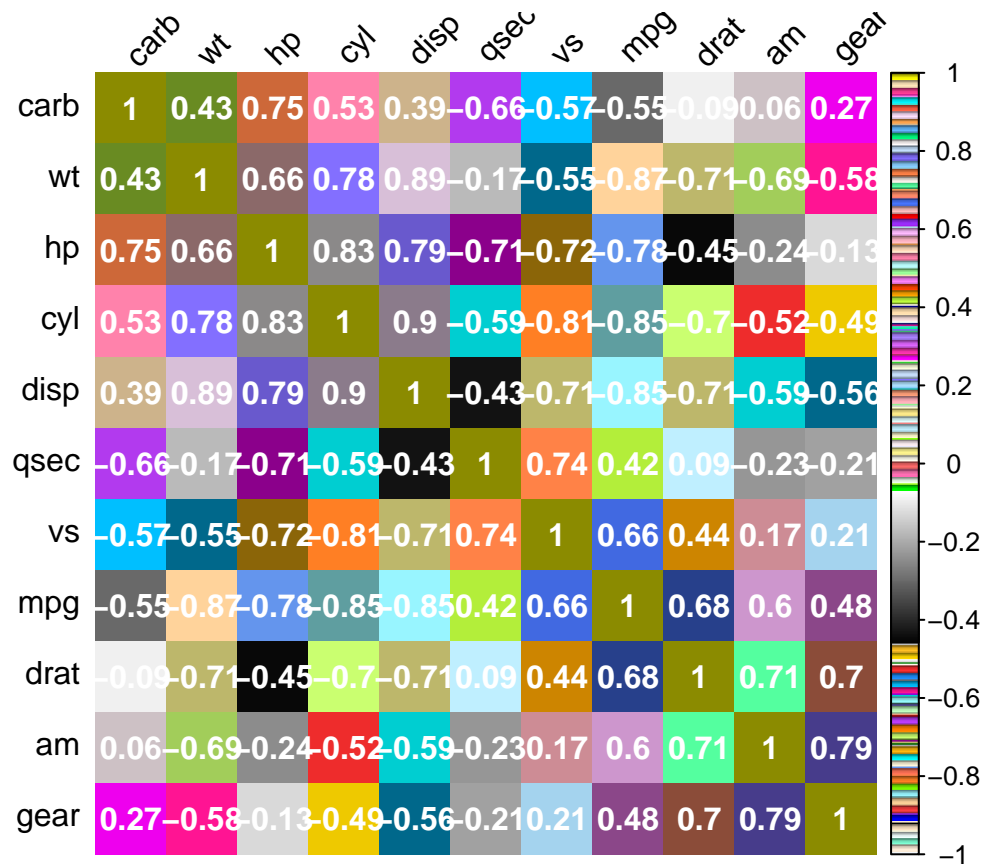


Display tables representing the correlation(alternative with order=hclust)

```
library(datasets)
library(corrplot)
data(mtcars)

# Generate a lighter palette
colors <- c("lightblue", "orange", "pink", "yellow", "yellowgreen", "white", "red", "maroon", "fuchsia",
            "aqua")

corrplot(mt_cor, method="shade", shade.col=NA, tl.col="black", tl.srt=45,
         col=colors(200), addCoef.col="white", addcolorlabel="no", order="hclust")
```



## Options for corrplot()

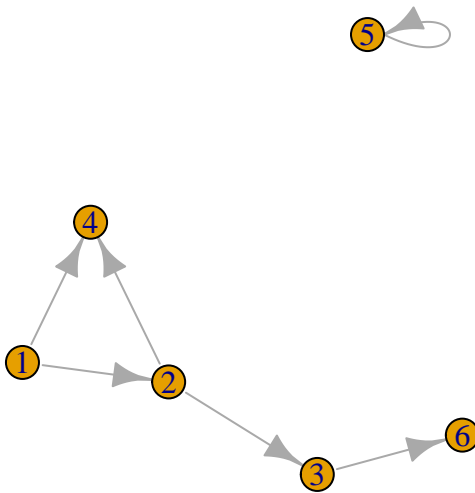
- type={"lower" | "upper"} Only use the lower or upper triangle.
- diag=FALSE Don't show values on the diagonal.
- addshade="all" Add lines indicating the direction of the correlation.
- shade.col=NA Hide correlation direction lines.
- method="shade" Use colored squares.
- method="ellipse" Use ellipses.
- addCoef.col="color" Add correlation coefficients, in color.
- tl.srt="number" Specify the rotation angle for top labels.
- tl.col="color" Specify the label color.
- order={"AOE" | "FPC" | "hclust"} Sort labels using angular order of eigenvectors, first principle component, or hierarchical clustering

## Creating a Network Graph

- Use the igraph package. To create a graph, pass a vector containing pairs of items to graph(), then plot the resulting object

```
library(igraph)
library(ggplot2)
library(plotly)
```

```
# Specify edges for a directed graph  
gd <- graph(c(1,2, 2,3, 2,4, 1,4, 5,5, 3,6))  
plot(gd)
```



```
# For an undirected graph  
gu <- graph(c(1,2, 2,3, 2,4, 1,4, 5,5, 3,6), directed=FALSE)  
# No labels  
plot(gu, vertex.label=NA)
```



