

```

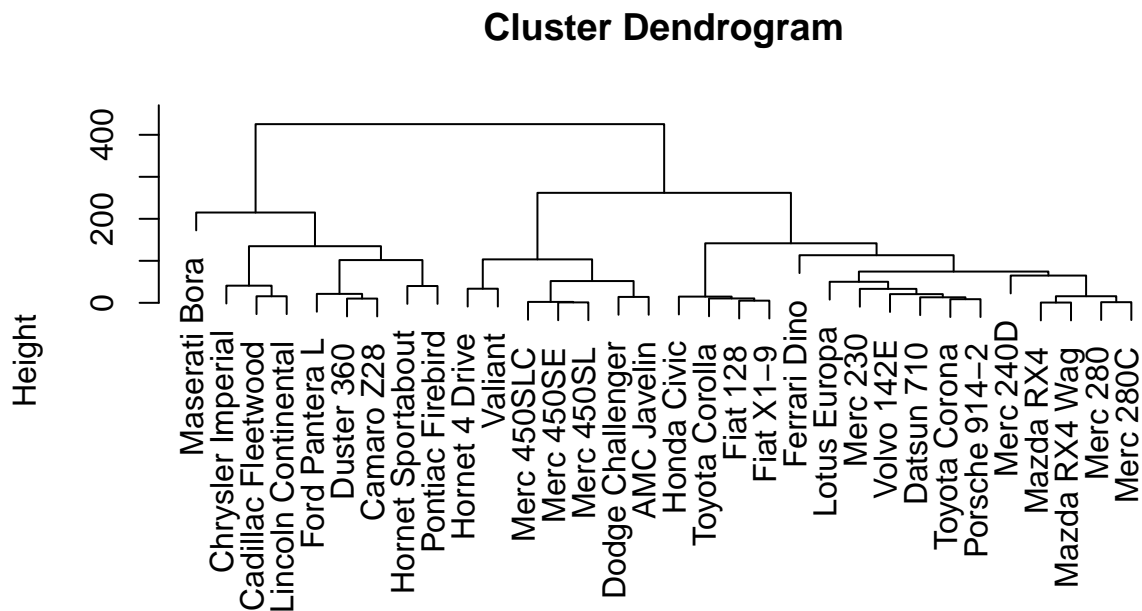
# Lab 14_Q3
# Load the mtcars dataset
data(mtcars)

# Lab 14_Q3A
# Compute the distance matrix
dist_matrix <- dist(mtcars)

# Perform hierarchical clustering
hclust_res <- hclust(dist_matrix, method = "complete")

# Plot the cluster dendrogram
plot(hclust_res, main = "Cluster Dendrogram", xlab = "", sub = "")

```



```
install.packages("ellipse") library (ellipse)
```

```

# Lab 14_Q3C
library (ellipse)

##
## Attaching package: 'ellipse'
## The following object is masked from 'package:graphics':
##
##      pairs
# Calculate correlation matrix
cor_matrix <- cor(mtcars)

# Create a color palette function from red to white to blue
col_palette <- colorRampPalette(c("red", "white", "blue"))

# Set up the plotting area
plotcorr(cor_matrix,
  type = "full", #Show only lower triangle
  col = col_palette(100), # Use 100 color levels
  mar = c(1,1,1,1), # Adjust margins

```

```

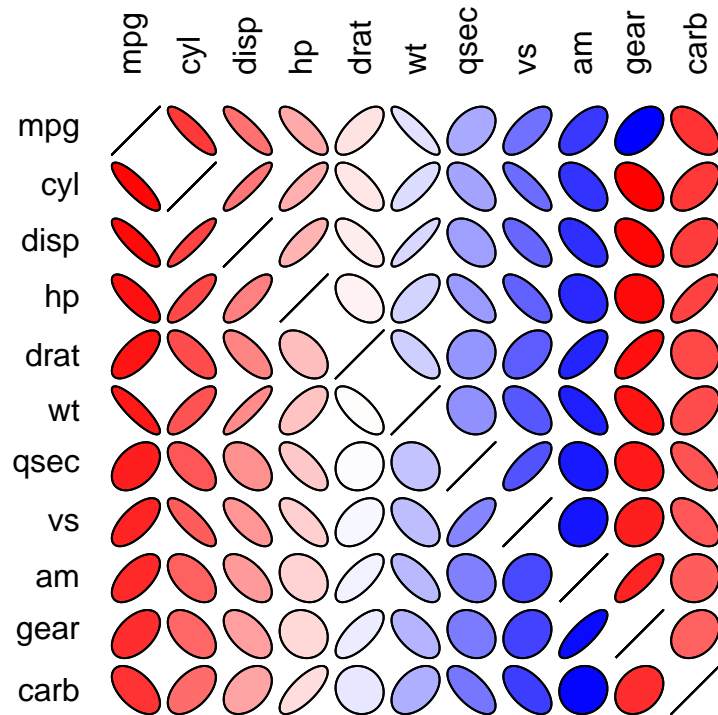
order = TRUE, # Order variables by correlation
diag = TRUE) # Show diagonal

```

```

## Warning in plot.window(...): "order" is not a graphical parameter
## Warning in plot.xy(xy, type, ...): "order" is not a graphical parameter
## Warning in title(...): "order" is not a graphical parameter

```



```

# Add variable names
colnames(mtcars) -> dimnames(cor_matrix)[[1]]

# Lab 14_Q3C
# Load required libraries
library(ellipse) # For plotcorr function

# Load the mtcars dataset
data(mtcars)

# Calculate correlation matrix
cor_matrix <- cor(mtcars)

# Make sure the correlation matrix has the proper dimnames
colnames(mtcars) -> dimnames(cor_matrix)[[1]]
colnames(mtcars) -> dimnames(cor_matrix)[[2]]

# Create a custom color matrix with diagonal color pattern
n <- ncol(cor_matrix)
custom_colors <- matrix(NA, n, n)

# Create diagonal gradient
for (i in 1:n) {
  for (j in 1:n) {

```

```

# Calculate position on diagonal (ranges from 0 to 1)
diag_position <- (i + j) / (2 * n)

# Map position to color (red->white->blue)
if (diag_position < 0.5) {
  # Red to white gradient for first half of diagonal
  intensity <- 2 * diag_position
  custom_colors[i, j] <- rgb(1, intensity, intensity)
} else {
  # White to blue gradient for second half of diagonal
  intensity <- 2 * (1 - diag_position)
  custom_colors[i, j] <- rgb(intensity, intensity, 1)
}
}
}

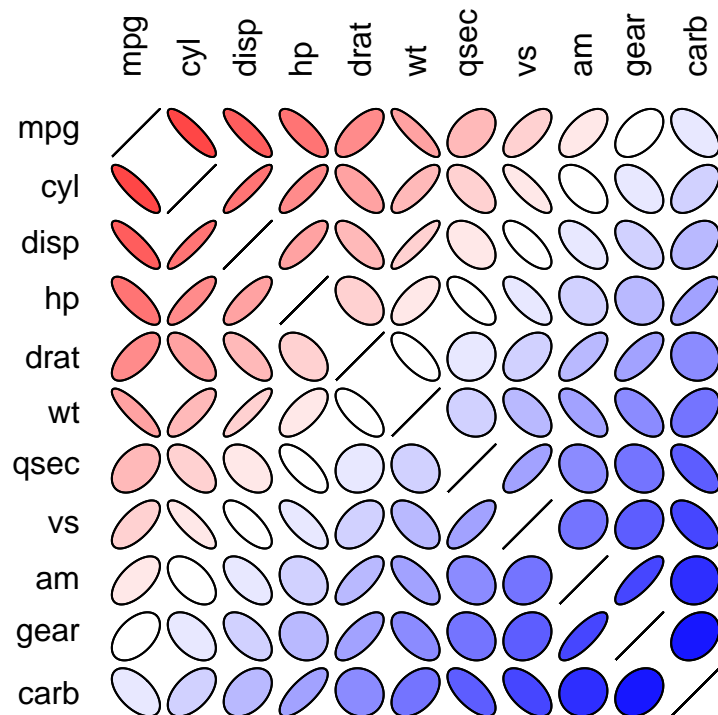
# Plot with custom colors
plotcorr(cor_matrix,
         col = custom_colors,
         mar = c(1, 1, 1, 1),
         diagx = TRUE)

```

```

## Warning in plot.window(...): "diagx" is not a graphical parameter
## Warning in plot.xy(xy, type, ...): "diagx" is not a graphical parameter
## Warning in title(...): "diagx" is not a graphical parameter

```



```

# Lab 14_Q3B
if (!require(corrgram)) {
  install.packages("corrgram")
  library(corrgram)
}

```

```

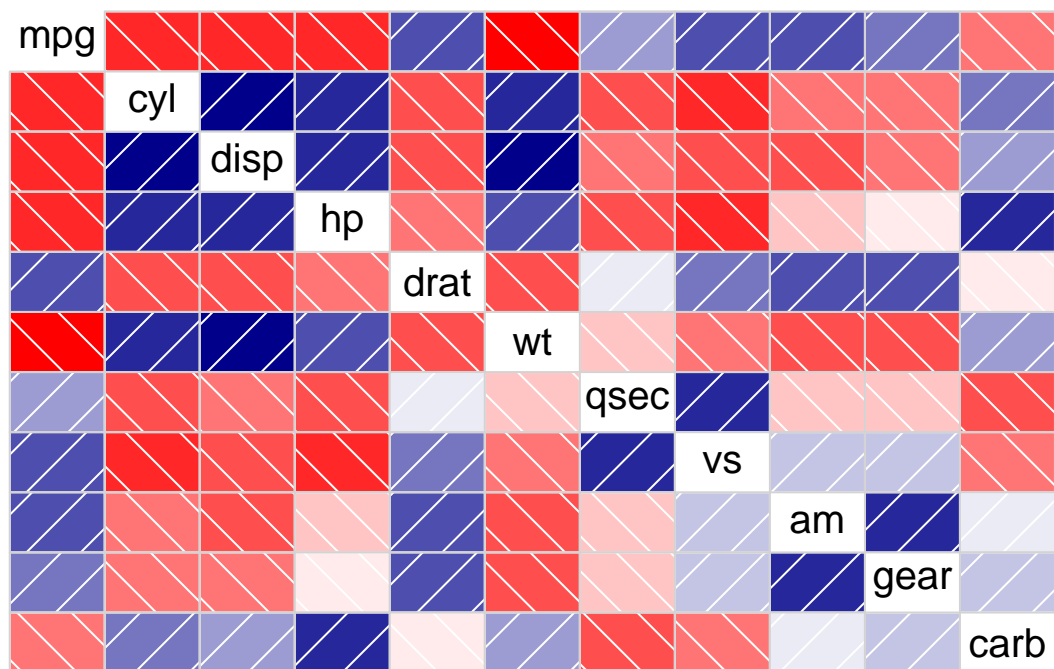
}

## Loading required package: corrgram

# Load the mtcars dataset (already available in R)
data(mtcars)

# Create the correlation plot with color mapping reversed to match reference image
corrgram(mtcars,
  order = FALSE,          # Don't reorder automatically
  lower.panel = panel.shade, # Lower panel uses colored squares
  upper.panel = panel.shade, # Upper panel also uses colored squares with lines
  text.panel = panel.txt,   # Diagonal panel shows variable names
  main = NULL,             # No title to match reference image
  col.regions = colorRampPalette(c("red", "white", "darkblue")) # Reversed colors

```



```

# Lab 14_Q3C
# Load required libraries
library(ellipse) # For plotcorr function

# Load the mtcars dataset
data(mtcars)

# Calculate correlation matrix
cor_matrix <- cor(mtcars)

# Define a color gradient function (red -> white -> blue)
color_gradient <- function(correlation) {
  if (correlation < 0) {
    rgb(1, 1 + correlation, 1 + correlation) # Red for negative correlations
  } else {
    rgb(1 - correlation, 1 - correlation, 1) # Blue for positive correlations
  }
}

```

```

}

# Create a color matrix based on the correlation values
color_matrix <- apply(cor_matrix, c(1, 2), color_gradient)

# Plot the correlation matrix with custom colors
plotcorr(cor_matrix,
         col = color_matrix,
         mar = c(1, 1, 1, 1),
         diag = TRUE)

```

