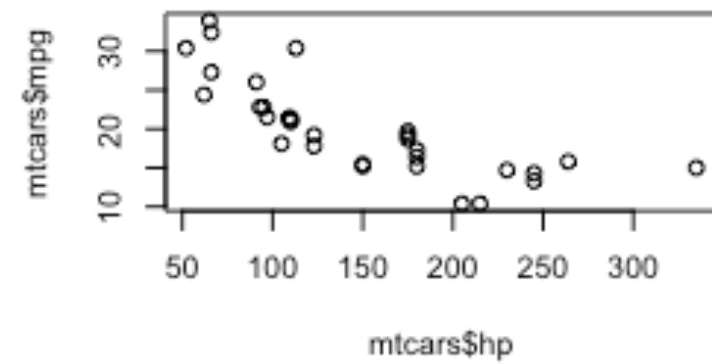




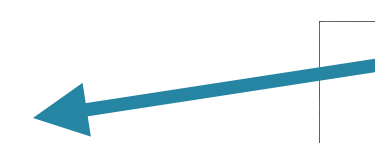
DATA VISUALIZATION IN R

Managing visual complexity

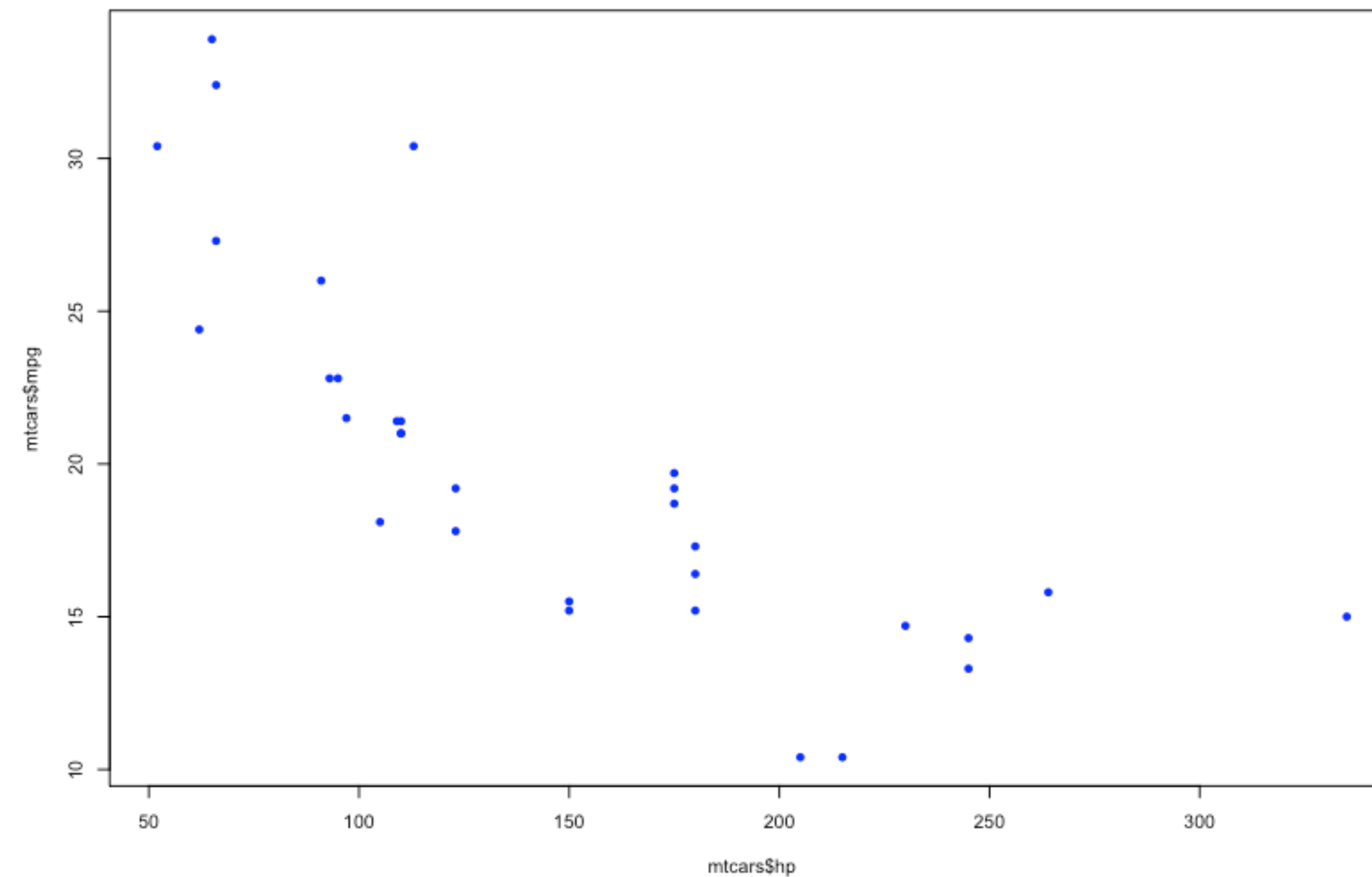
Useful visualizations show us details



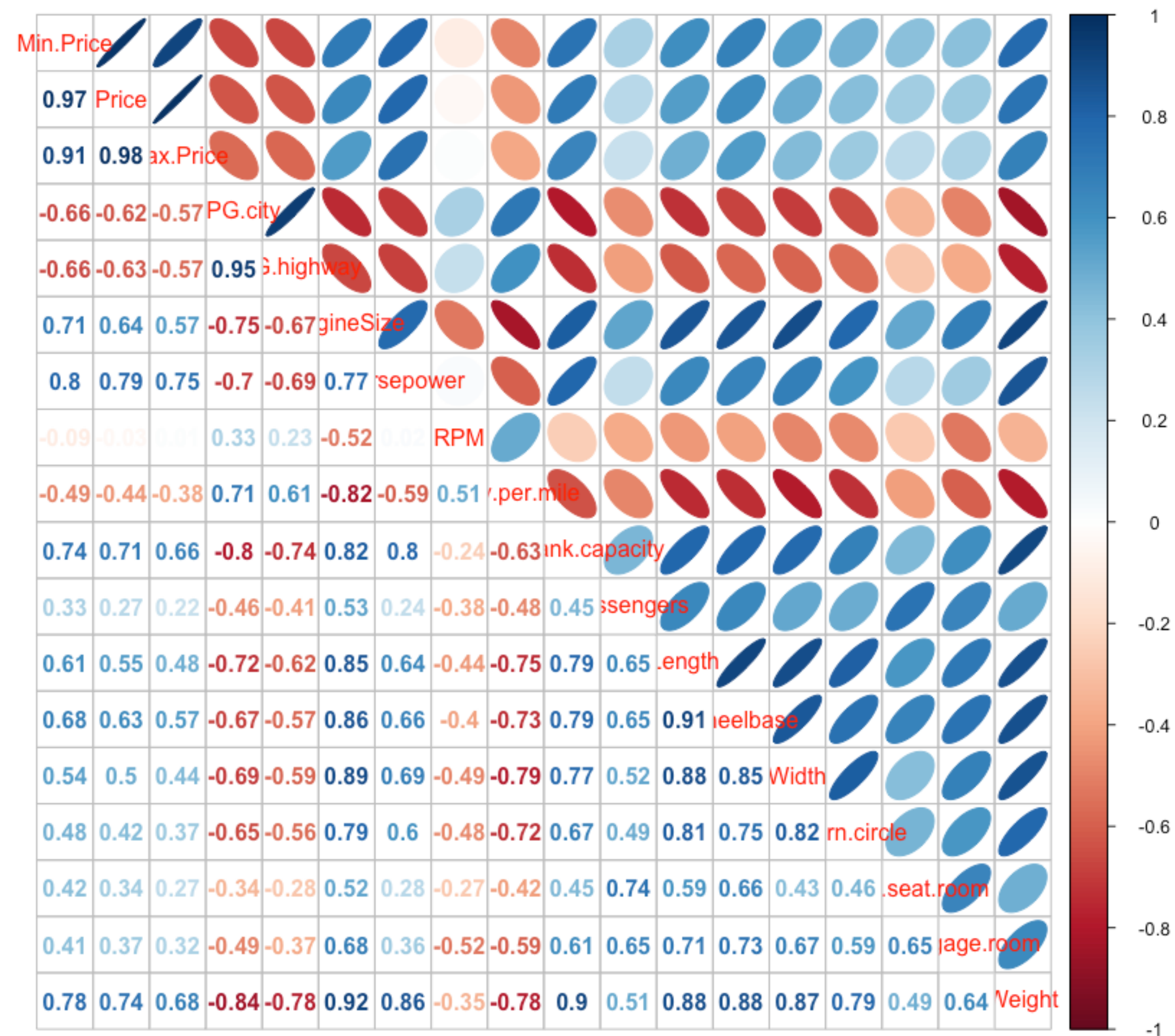
too small to be useful



Useful plot: details visible

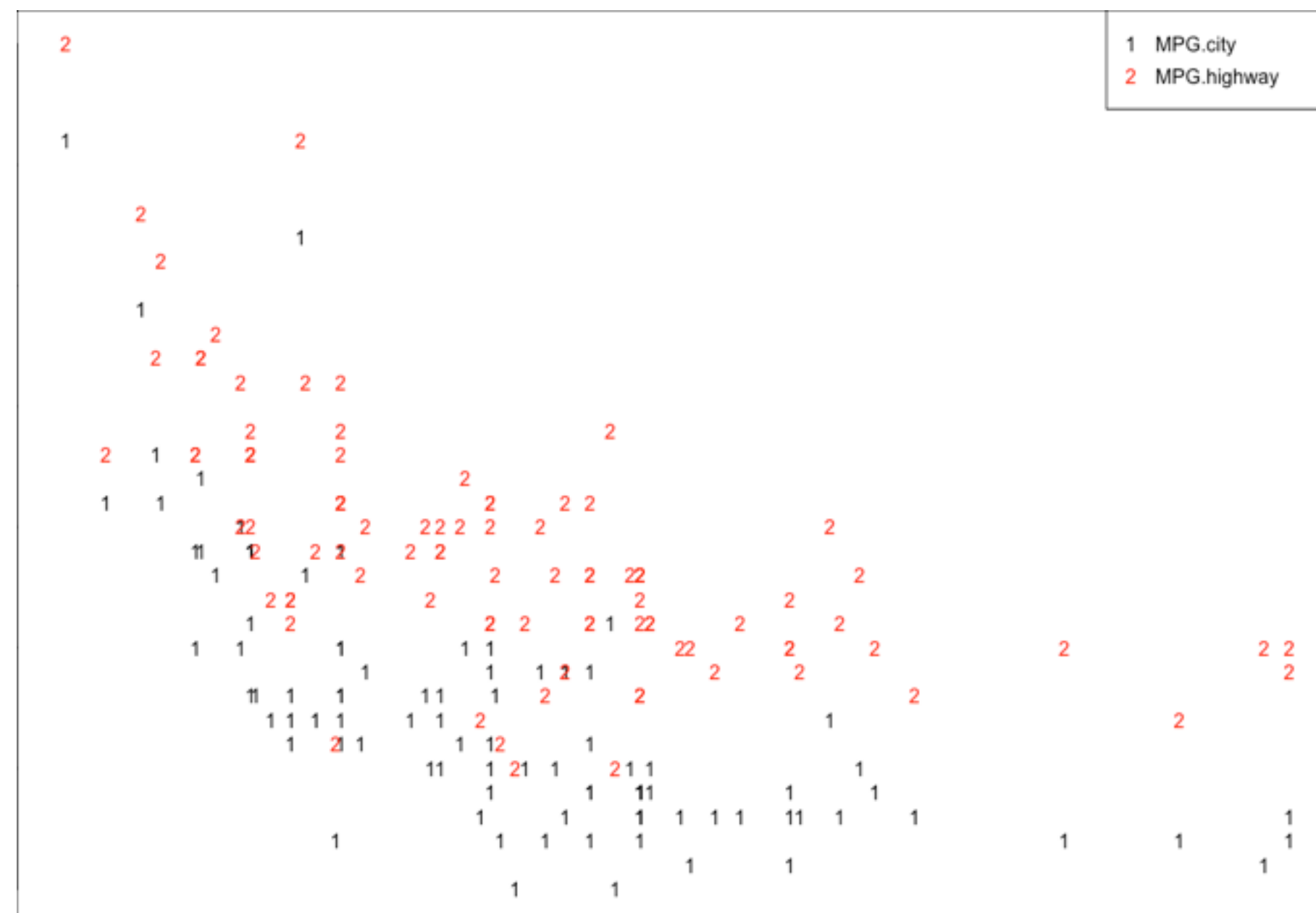


Useful visualizations don't attempt to show too many details



Multiple scatterplots on one set of axes: `matplot()`

```
> library(MASS)
> matplot(Cars93$Horsepower, Cars93[, c("MPG.city", "MPG.highway")],
          xlab = "Horsepower", ylab = "Gas mileages")
> legend("topright", pch = c("1", "2"), col = c("black", "red"),
        legend = c("MPG.city", "MPG.highway"))
```





DATA VISUALIZATION IN R

Let's practice!

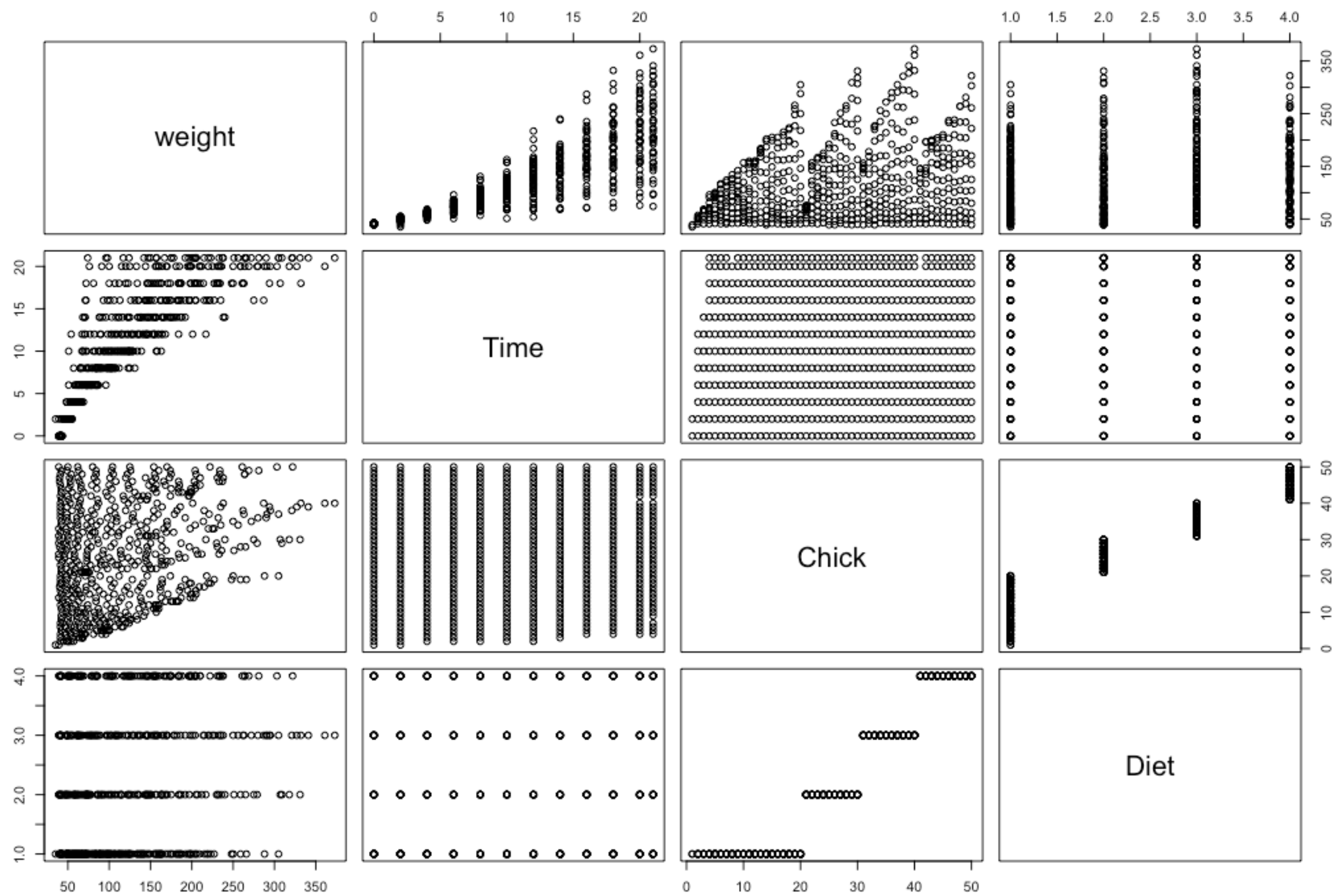


DATA VISUALIZATION IN R

Creating plot arrays with the `mflow` argument

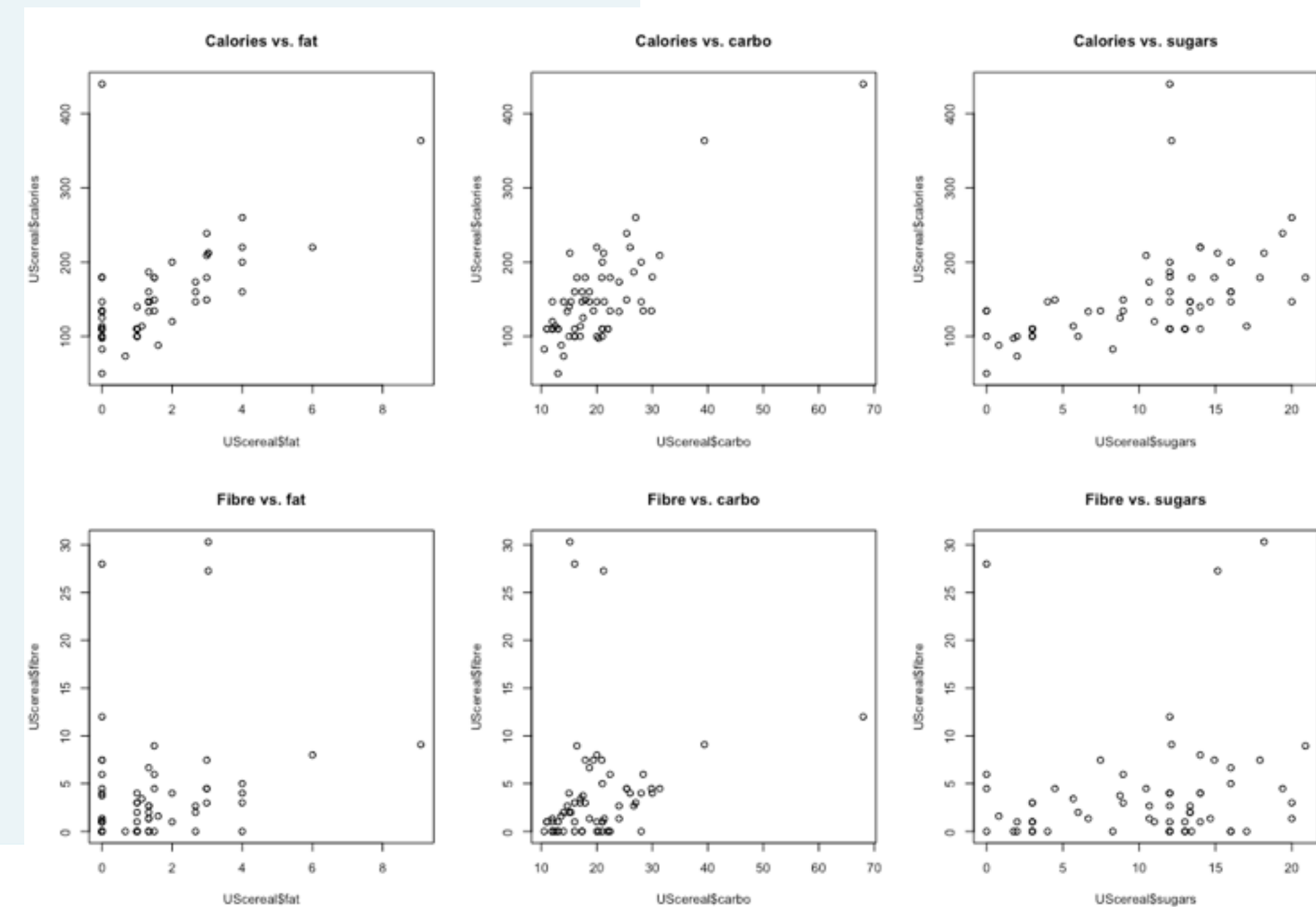
Multiple plot arrays

- Differences between datasets
- Different views of same dataset
- Similarities between datasets
- Related views of same dataset



Set up a plot array with R rows & C columns

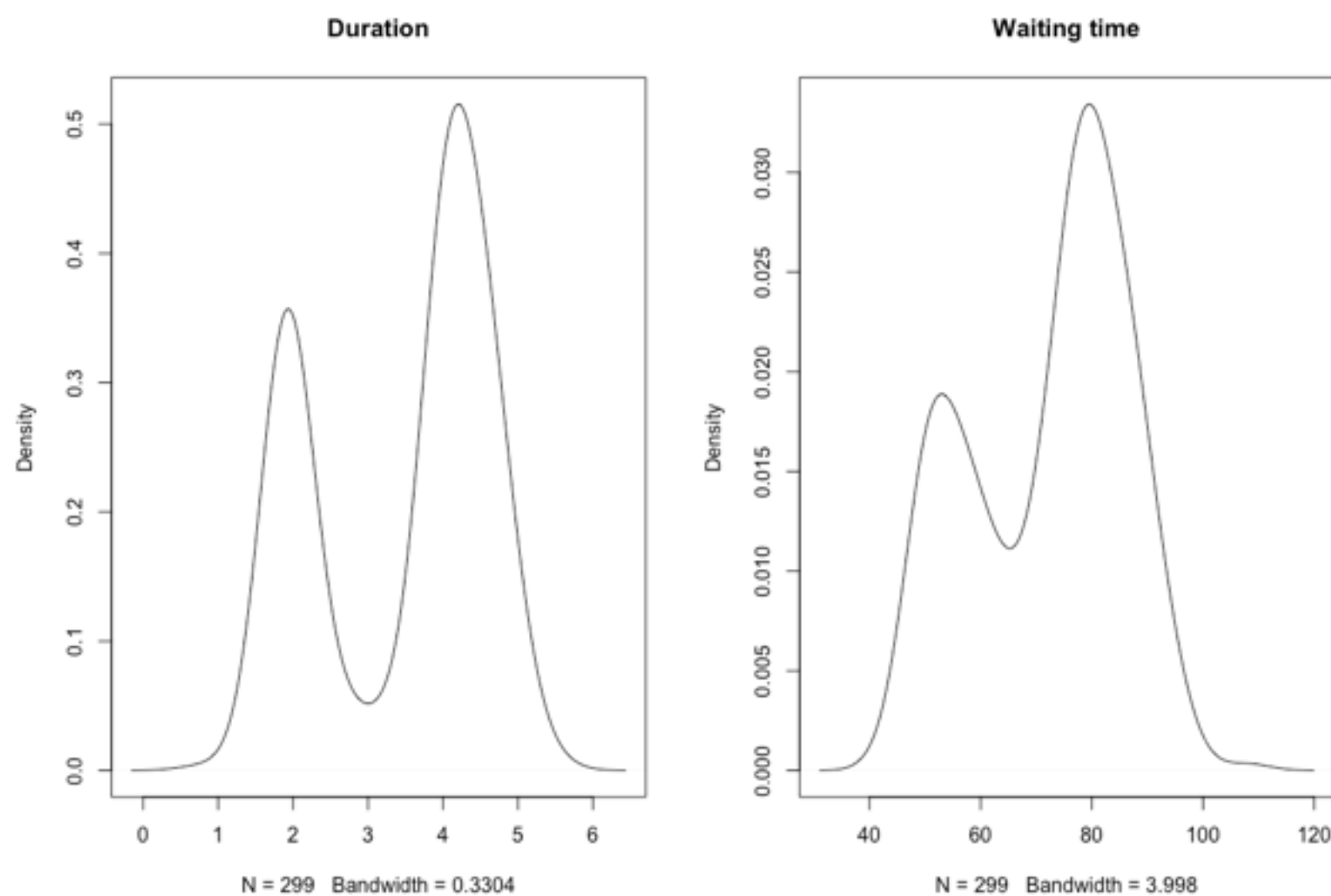
```
> library(MASS)
> par(mfrow = c(2, 3)) # Set up a 2-row, 3-column array
> plot(UScereal$fat, UScereal$calories)
> title("Calories vs. fat")
> plot(UScereal$carbo, UScereal$calories)
> title("Calories vs. carbo")
> plot(UScereal$sugars, UScereal$calories)
> title("Calories vs. sugars")
> plot(UScereal$fat, UScereal$fibre)
> title("Fibre vs. fat")
> plot(UScereal$carbo, UScereal$fibre)
> title("Fibre vs. carbo")
> plot(UScereal$sugars, UScereal$fibre)
> title("Fibre vs. sugars")
```



Creating side-by-side plot pairs

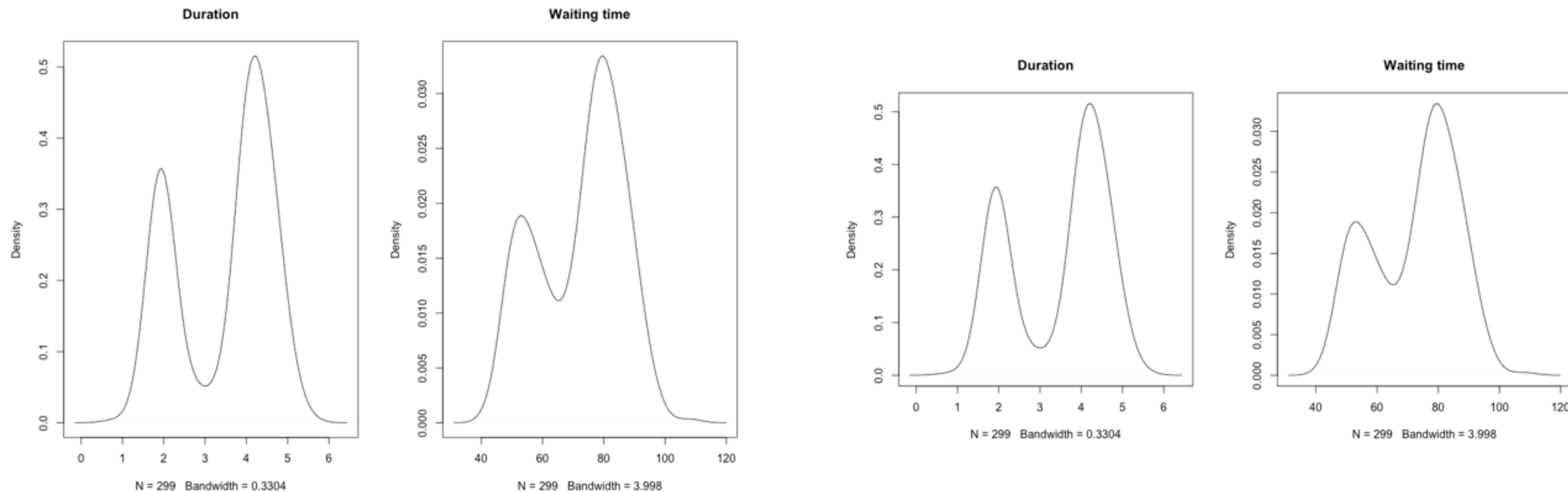
```
> library(MASS)
> par(mfrow = c(1, 2))

> plot(density(geyser$duration), main = "Duration")
> plot(density(geyser$waiting), main = "Waiting time")
```



Creating side-by-side plot pairs

```
> library(MASS)
> par(mfrow = c(1, 2))
> par(pty = "s")
> plot(density(geyser$duration), main = "Duration")
> plot(density(geyser$waiting), main = "Waiting time")
```





DATA VISUALIZATION IN R

Let's practice!



DATA VISUALIZATION IN R

Creating plot arrays with the `layout()` function

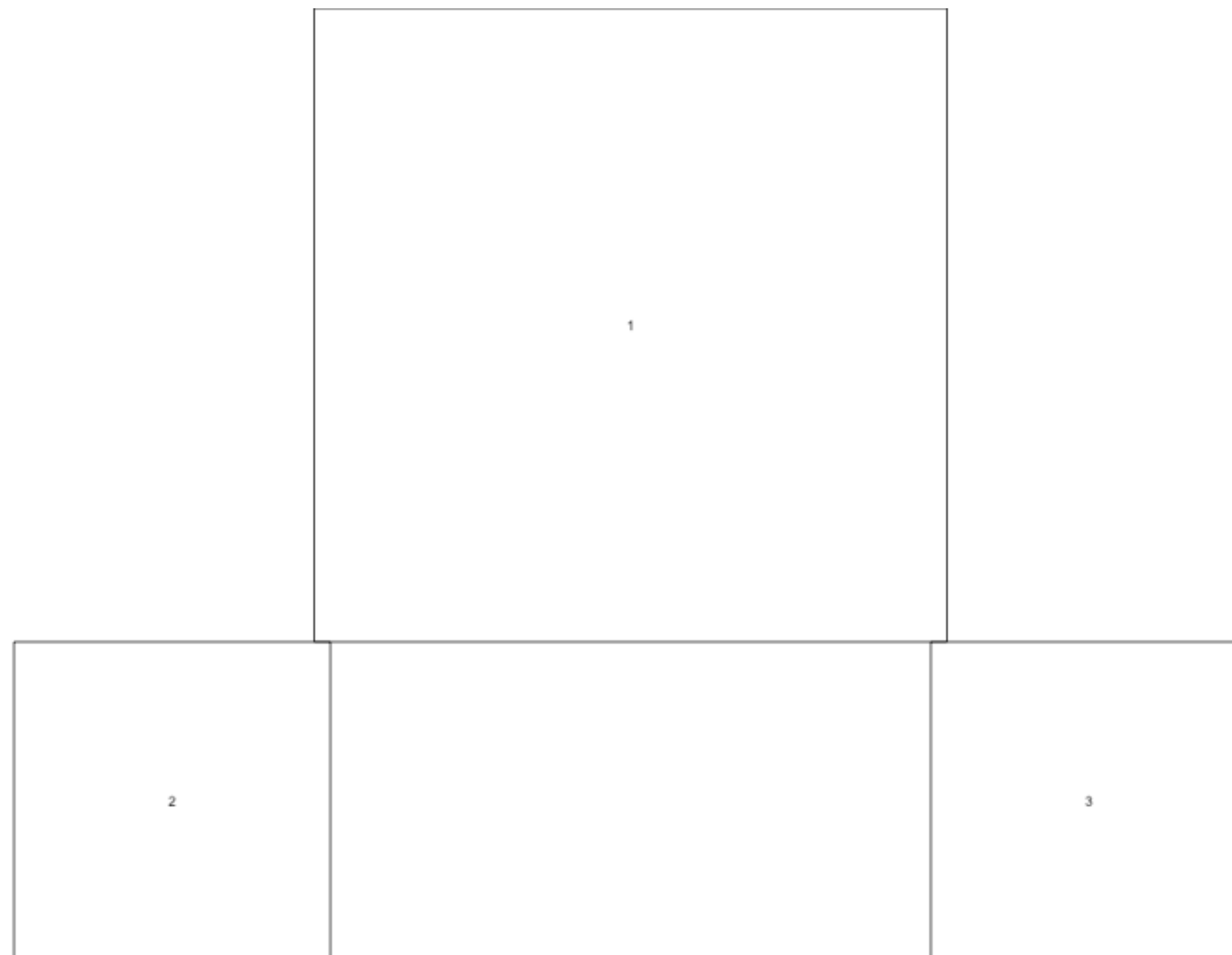
layout() uses a matrix to define the plot array

```
> rowA <- c(1, 1, 1)
> rowB <- c(2, 0, 3)
> layoutVector <- c(rowA, rowA, rowB)
> layoutMatrix <- matrix(layoutVector, byrow = TRUE, nrow = 3)
> layoutMatrix
```

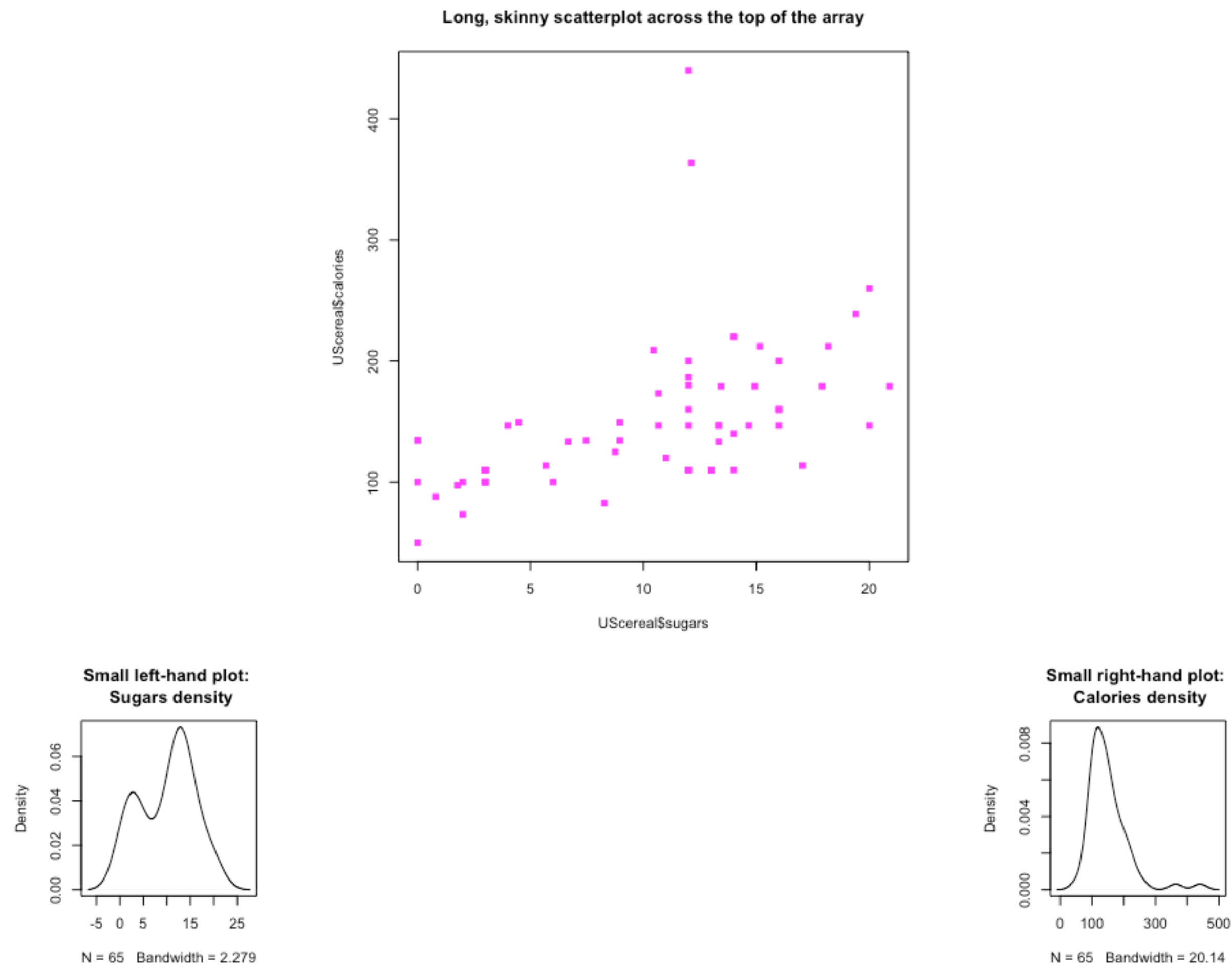
	[,1]	[,2]	[,3]
[1,]	1	1	1
[2,]	1	1	1
[3,]	2	0	3

`layout.show()` lets you see the structure of the array

```
> layout(layoutMatrix) # Use the matrix constructed previously  
> layout.show(n = 3) # See layout of all three plots
```



layout() allows you to create non-rectangular plot arrays



layout() allows you to create non-rectangular plot arrays

```
> rowA <- c(1, 1, 1)
> rowB <- c(2, 0, 3)
> layoutVector <- c(rowA, rowA, rowB)
> layoutMatrix <- matrix(layoutVector, byrow = TRUE, nrow = 3)
> layout(layoutMatrix)
> library(MASS)
> plot(UScereal$sugars, UScereal$calories, pch = 15,
      col = "magenta")
> title("Long, skinny scatterplot across the top of the array")
> plot(density(UScereal$sugars),
      main = "Small left-hand plot: \n Sugars density")
> plot(density(UScereal$calories),
      main = "Small right-hand plot: \n Calories density")
```



DATA VISUALIZATION IN R

Let's practice!