

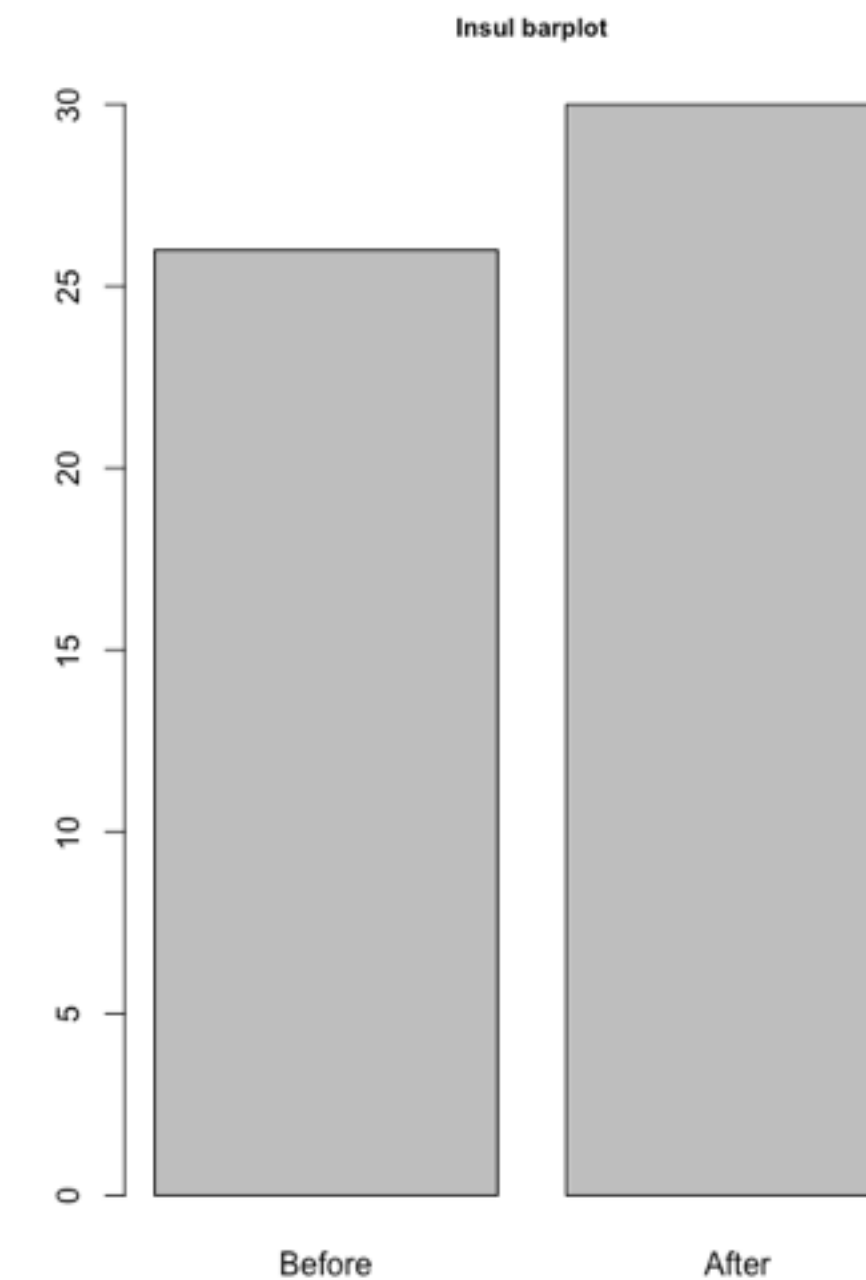
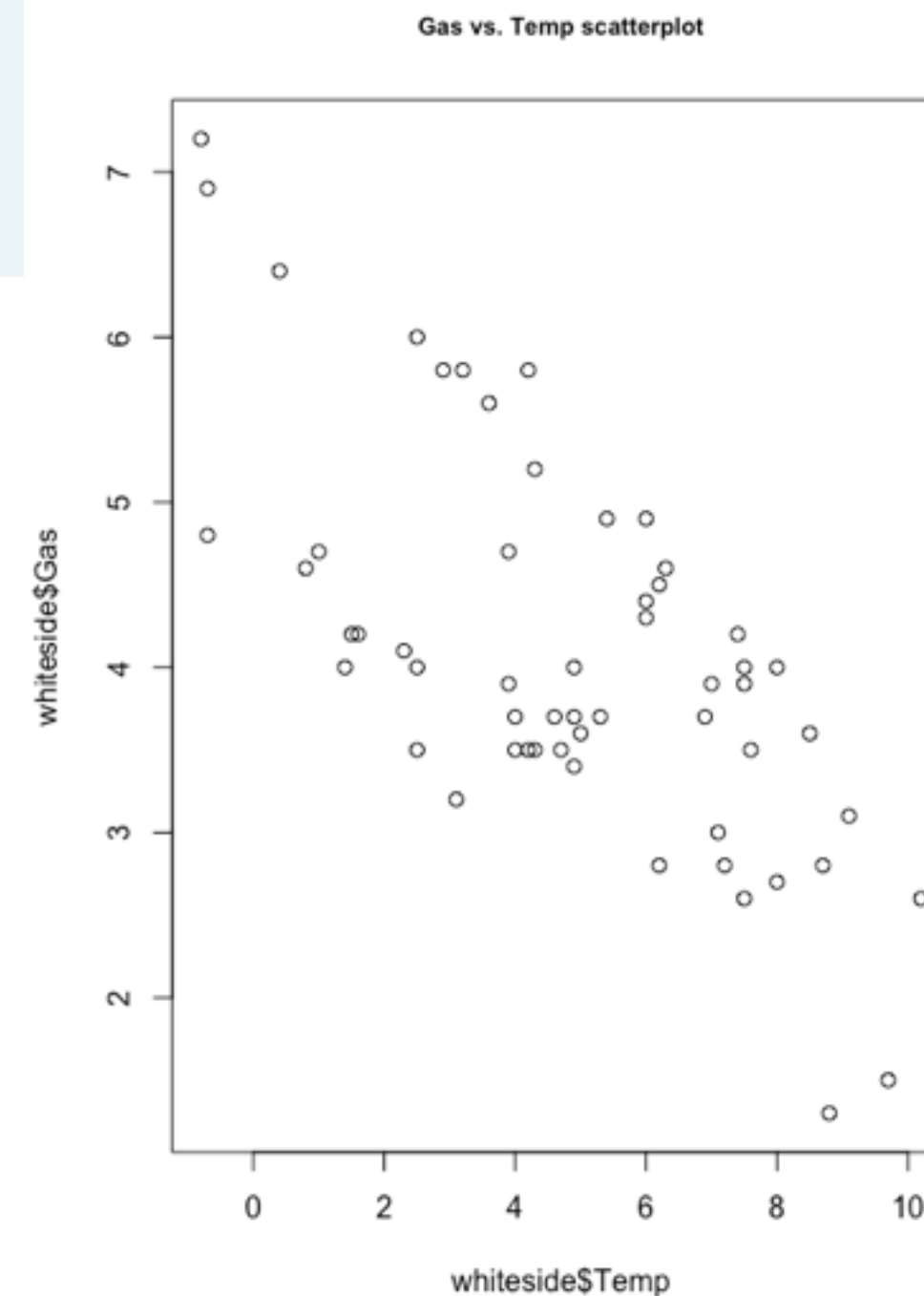


DATA VISUALIZATION IN R

The `plot()` function and its options

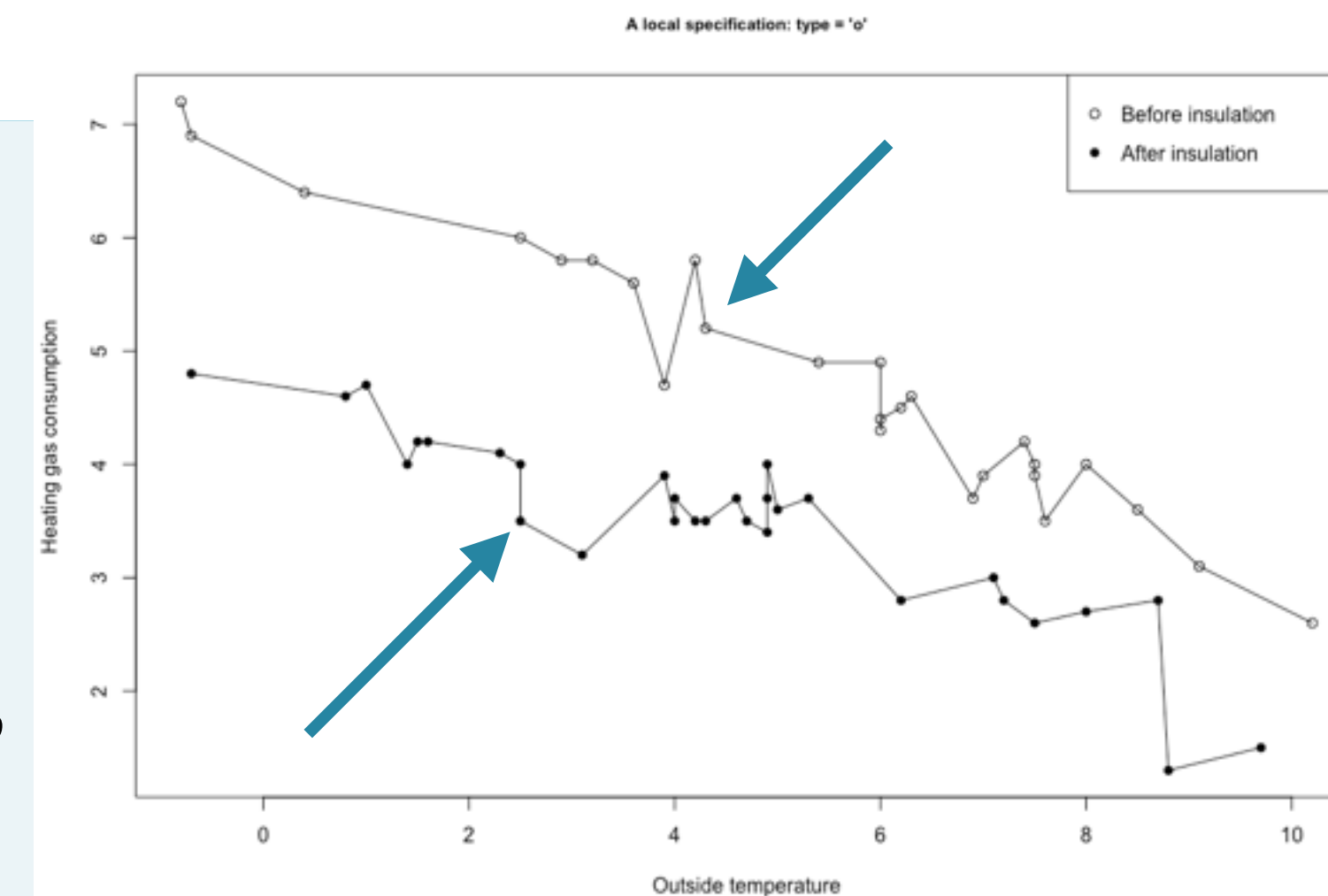
Some options can or must be specified globally

```
> library(MASS)
> par(mfrow = c(1, 2)) # mfrow specified globally
> par(cex.main = 0.8) # cex.main specified either locally or globally
> plot(whiteside$Temp, whiteside$Gas)
> title("Gas vs. Temp scatterplot")
> plot(whiteside$Insul)
> title("Insul barplot")
```



Other options can only be specified locally

```
> library(MASS)
> indexA <- which(whiteside$Insul == "After")
> indexB <- which(whiteside$Insul == "Before")
> x <- whiteside$Temp
> y <- whiteside$Gas
> plot(x[indexA], y[indexA], type = "o", pch = 16,
       xlim = range(x), ylim = range(y),
       xlab = "Outside temperature",
       ylab = "Heating gas consumption") # high-level function
> lines(x[indexB], y[indexB], type = "o", pch = 1) # low-level function
> legend("topright", pch = c(1, 16),
       legend = c("Before insulation", "After insulation"))
> title("A local specification: type = 'o'")
```



type = "n"



DATA VISUALIZATION IN R

Let's practice!

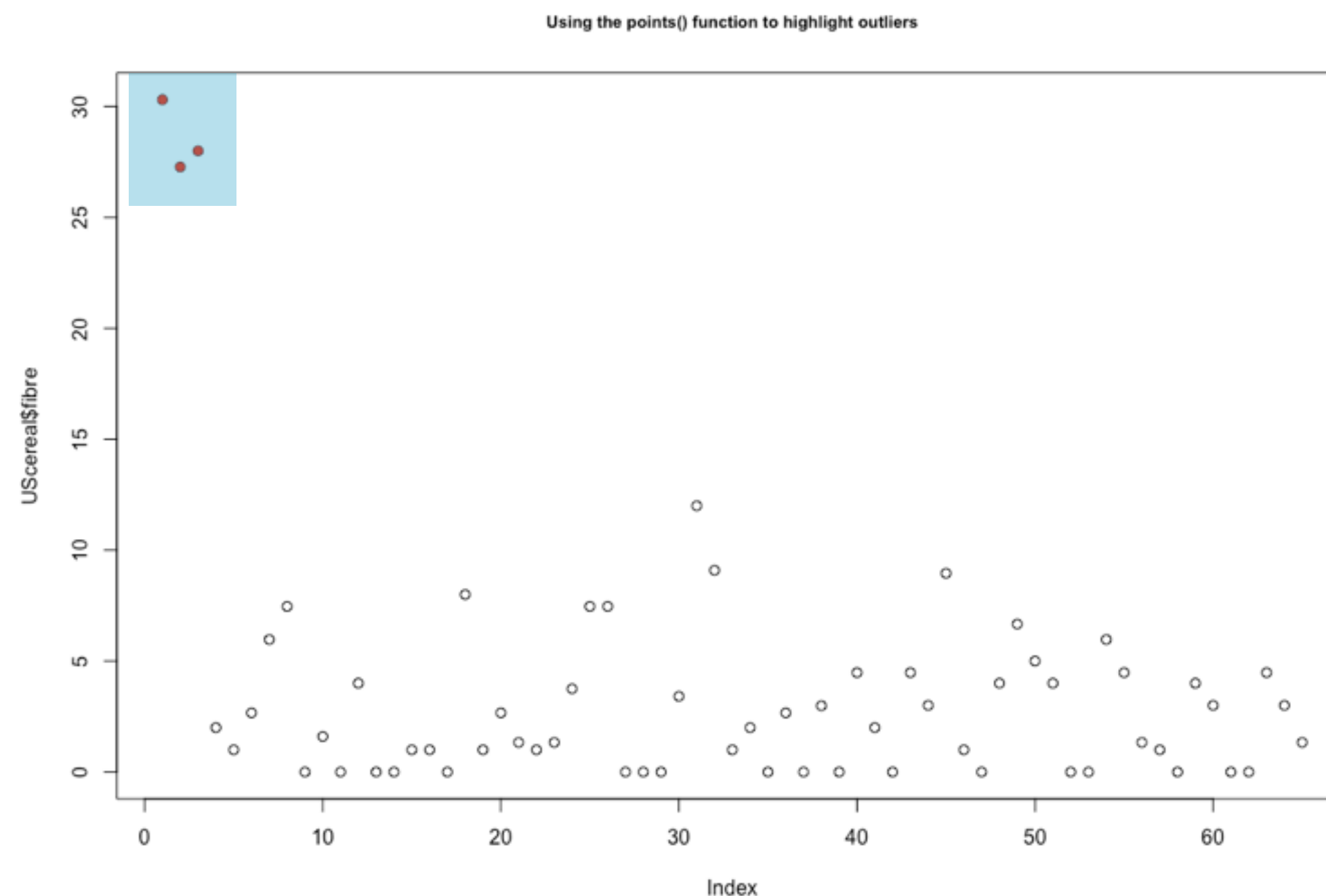


DATA VISUALIZATION IN R

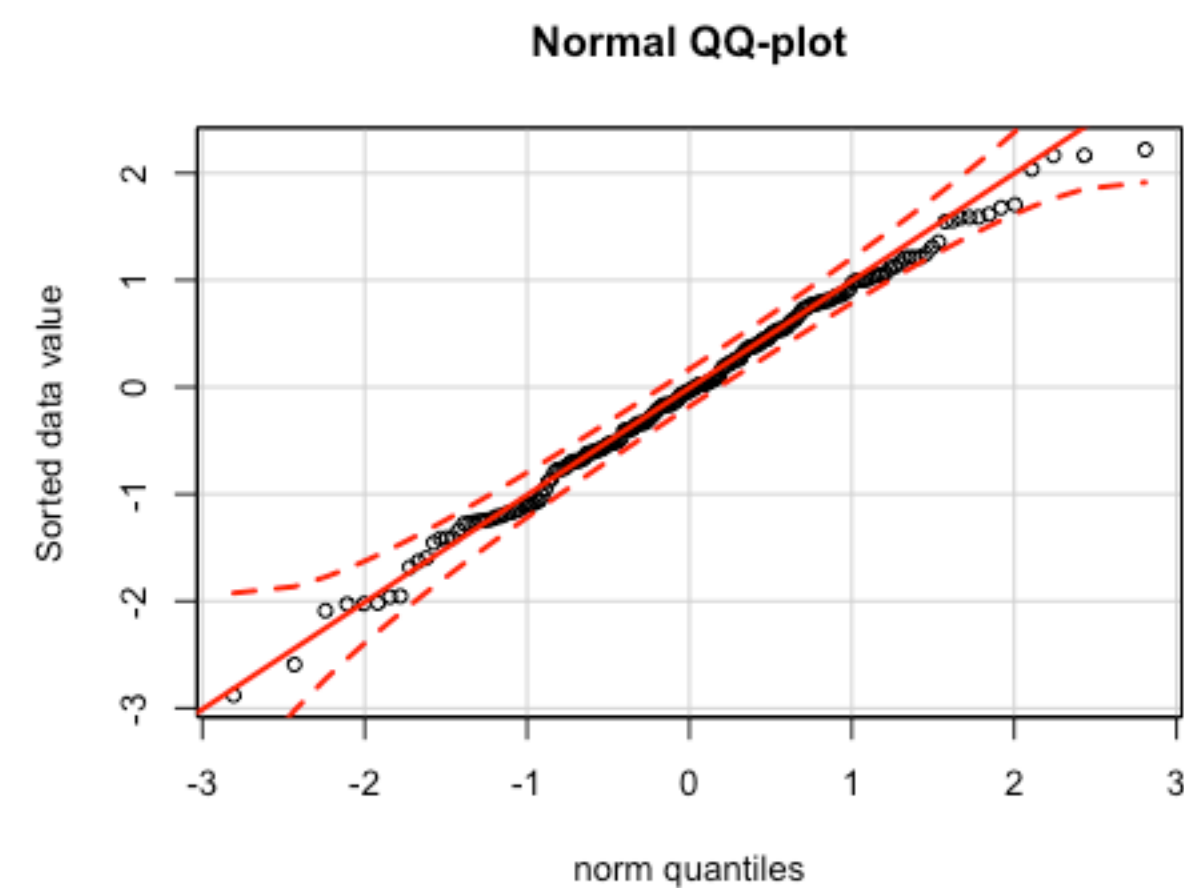
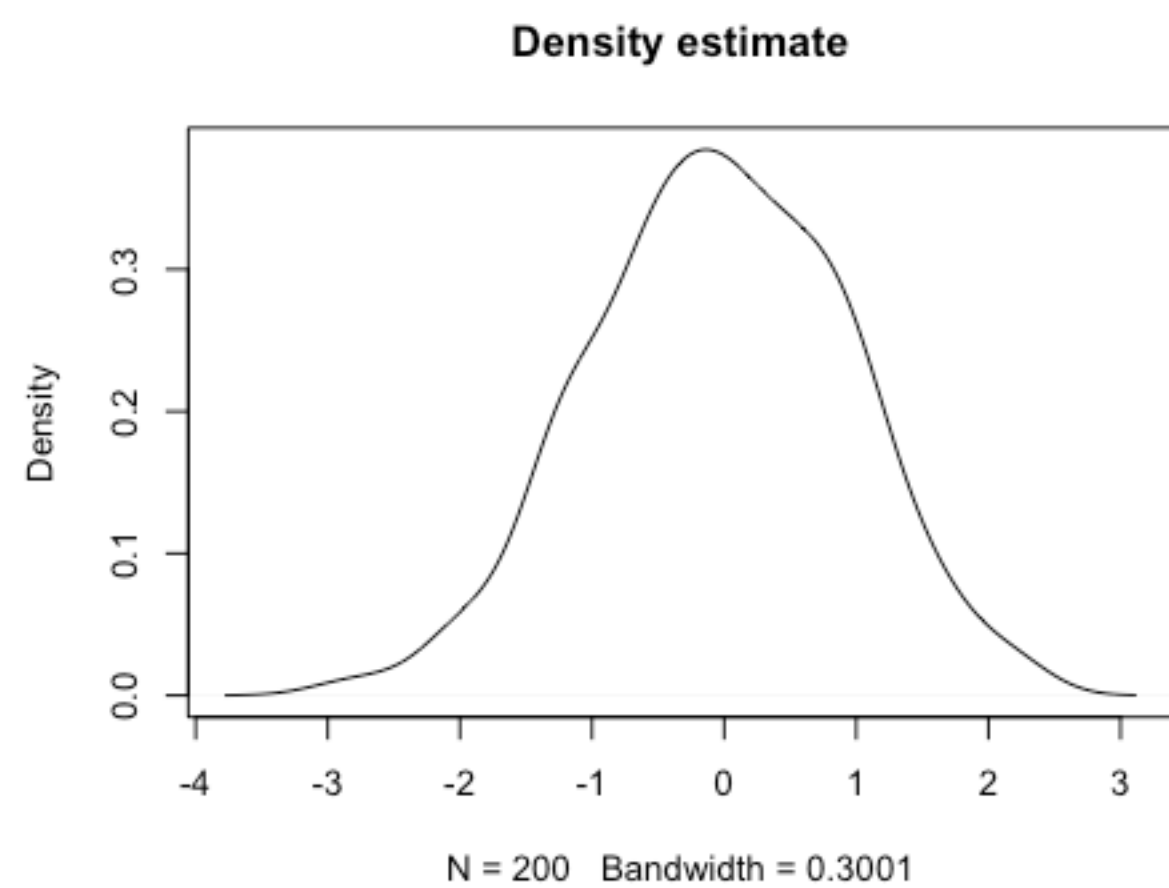
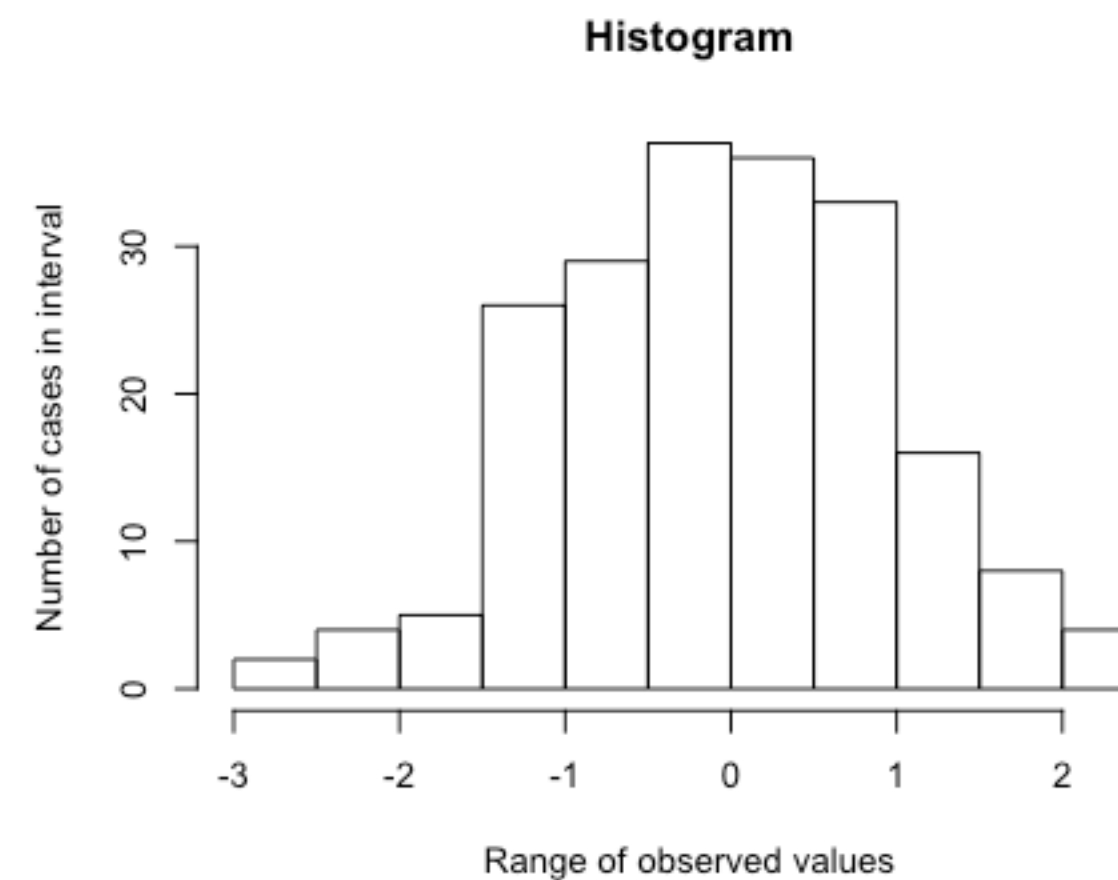
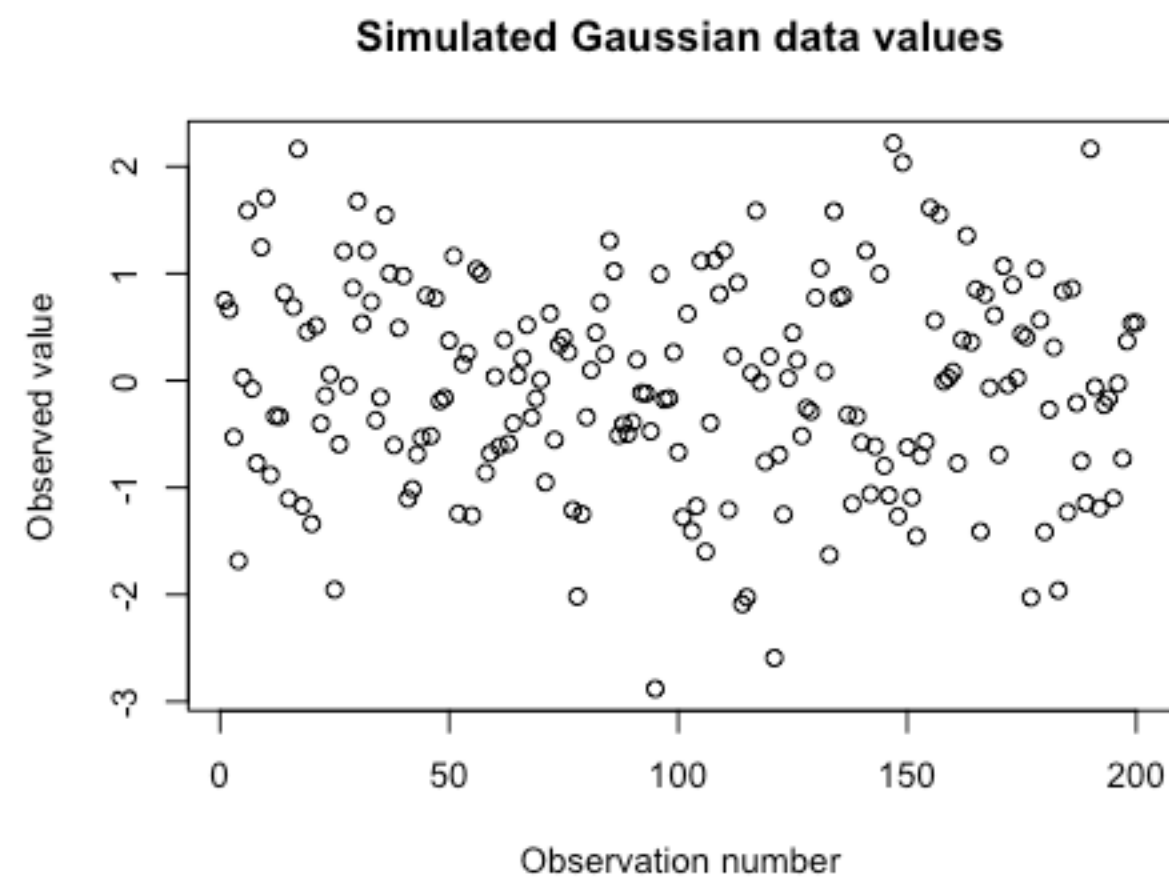
Adding lines and points to plots

Using the `points()` function to add points to a plot

```
> library(MASS)
> plot(UScereal$fibre)
> index <- which(UScereal$fibre > 20)
> points(index, UScereal$fibre[index], pch = 16, col = "red")
> title("Using the points() function to highlight outliers")
```

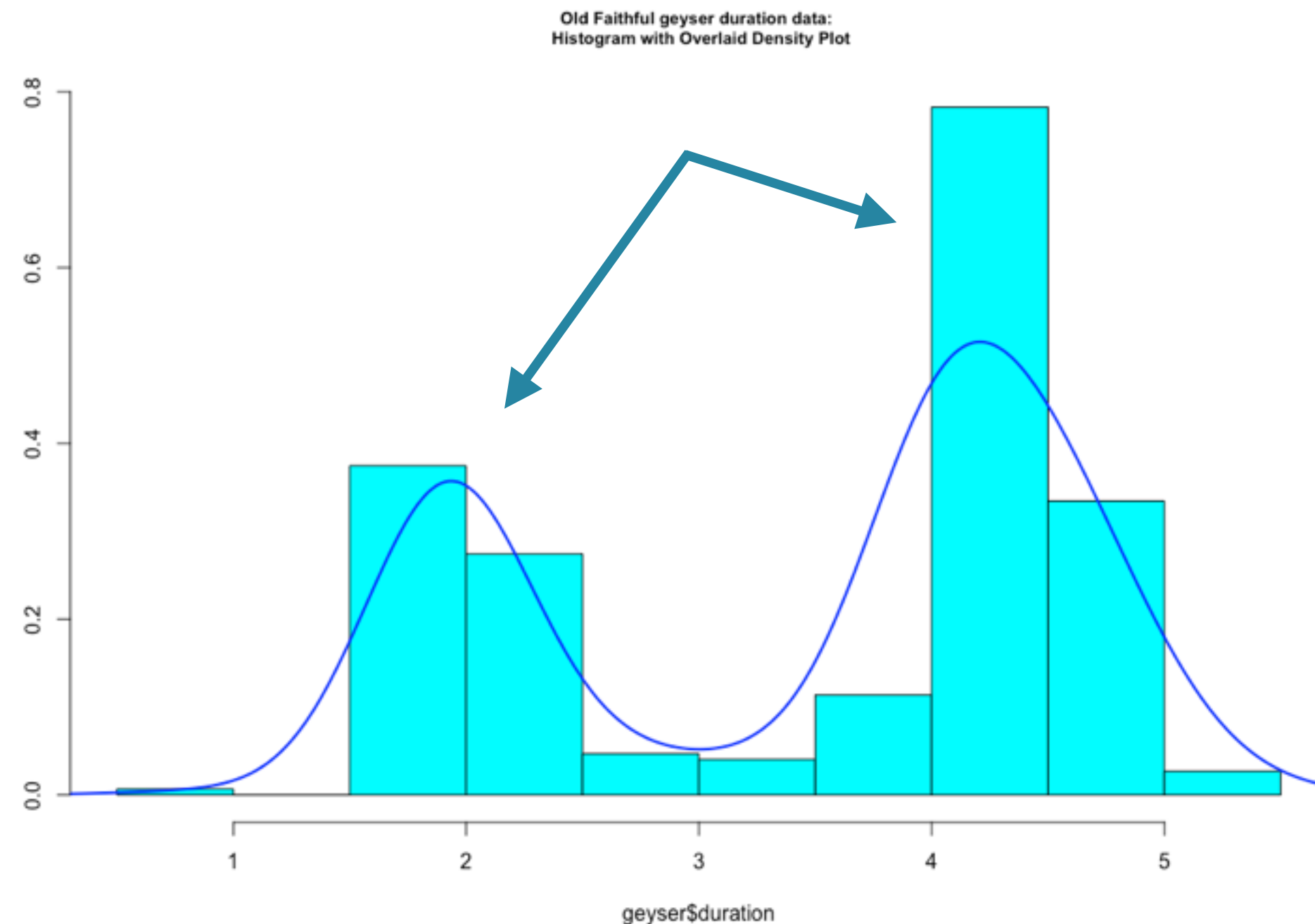


A refresher



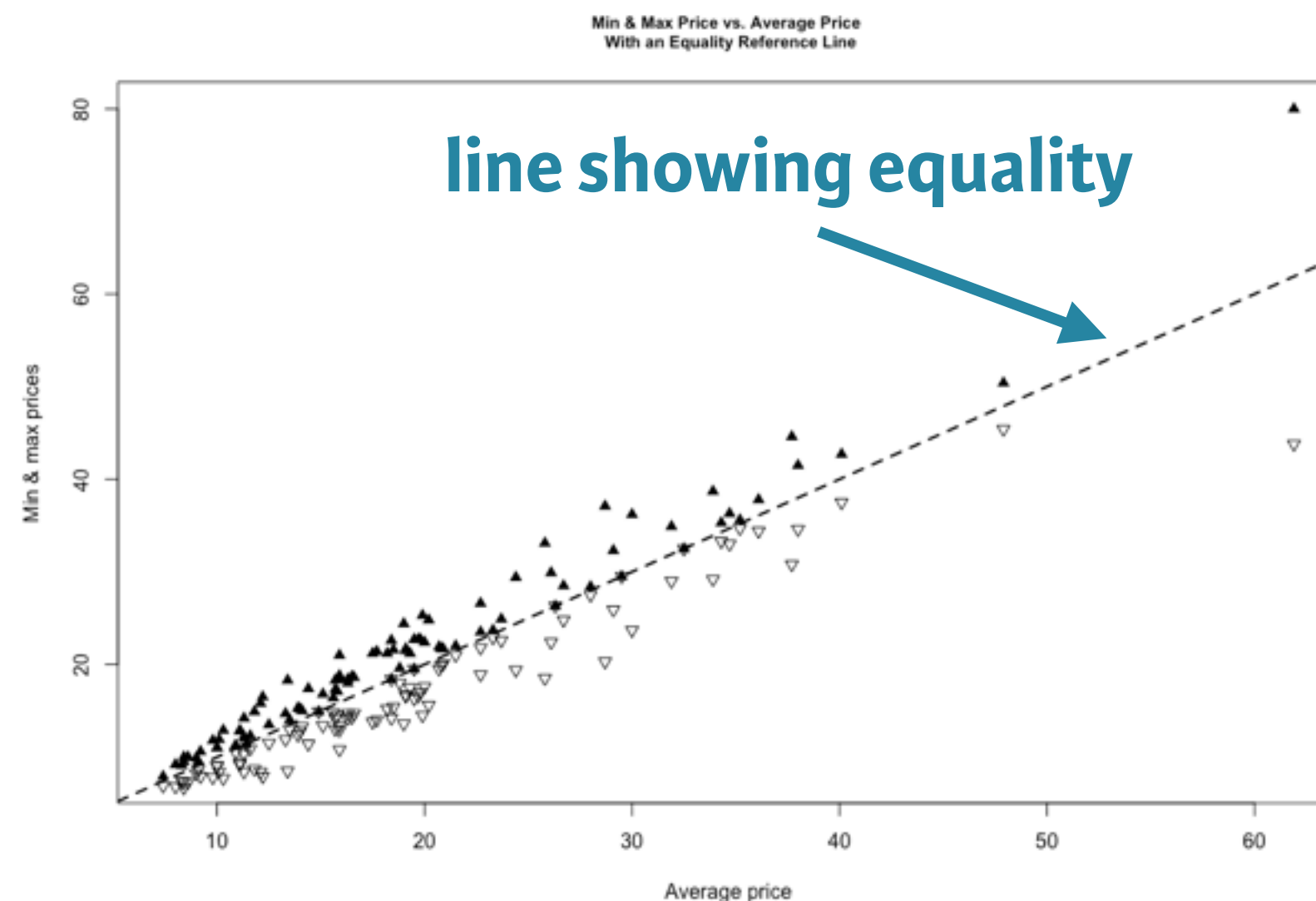
Using the `lines()` function to add lines to a plot

```
> library(MASS)
> truehist(geyser$duration)
> lines(density(geyser$duration), lwd = 2, col = "blue")
> title("Old Faithful geyser duration data:
  \n Histogram with Overlaid Density Plot")
```



Using the `abline()` function to add lines to a plot

```
> library(MASS)
> plot(Cars93$Price, Cars93$Max.Price, pch = 17,
       xlab = "Average price", ylab = "Min & max prices")
> points(Cars93$Price, Cars93$Min.Price, pch = 6)
> abline(a = 0, b = 1, lty = 2, lwd = 2)
> title("Min & Max Price vs. Average Price \n With an Equality
       Reference Line")
```





DATA VISUALIZATION IN R


Let's practice!



DATA VISUALIZATION IN R

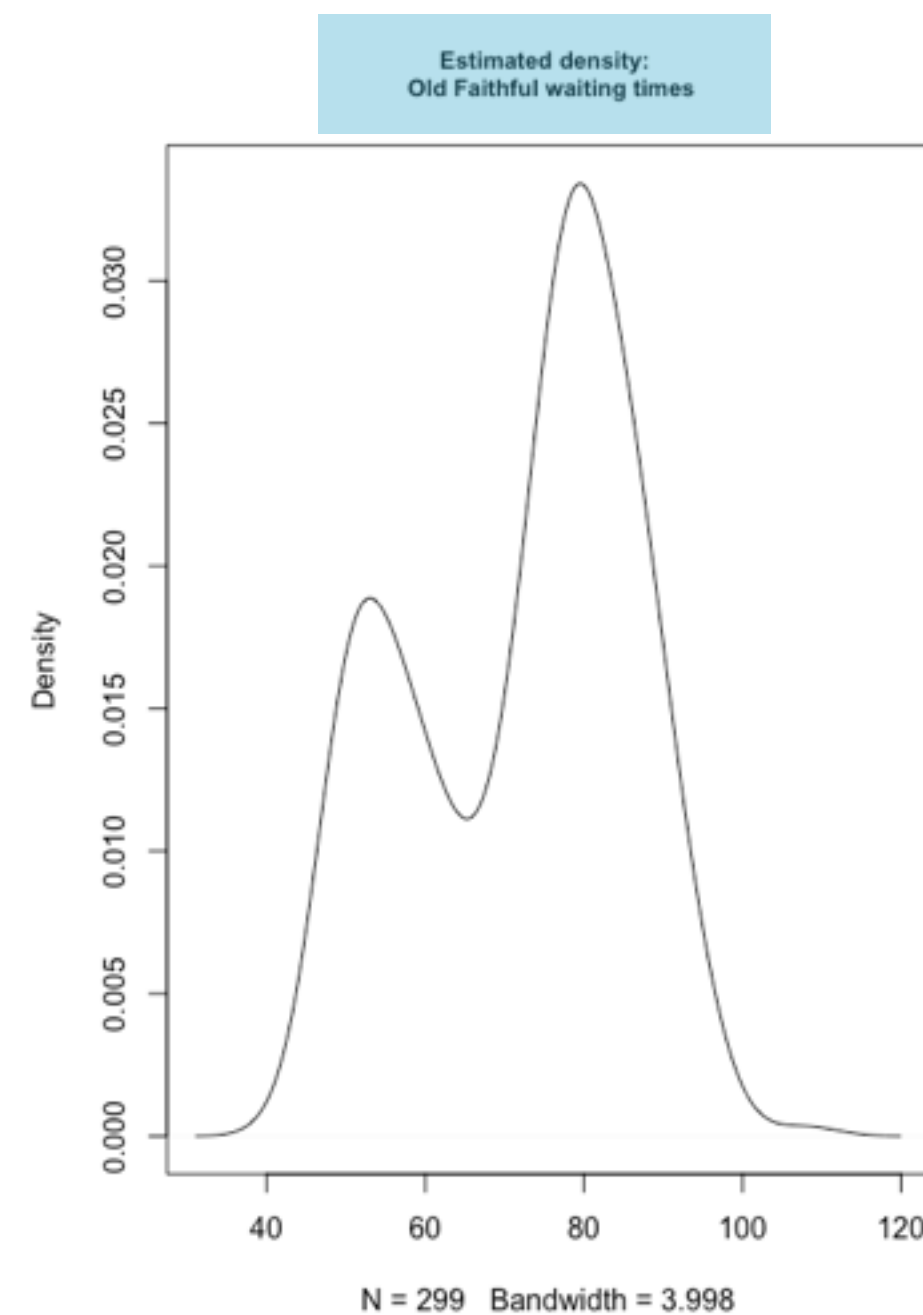
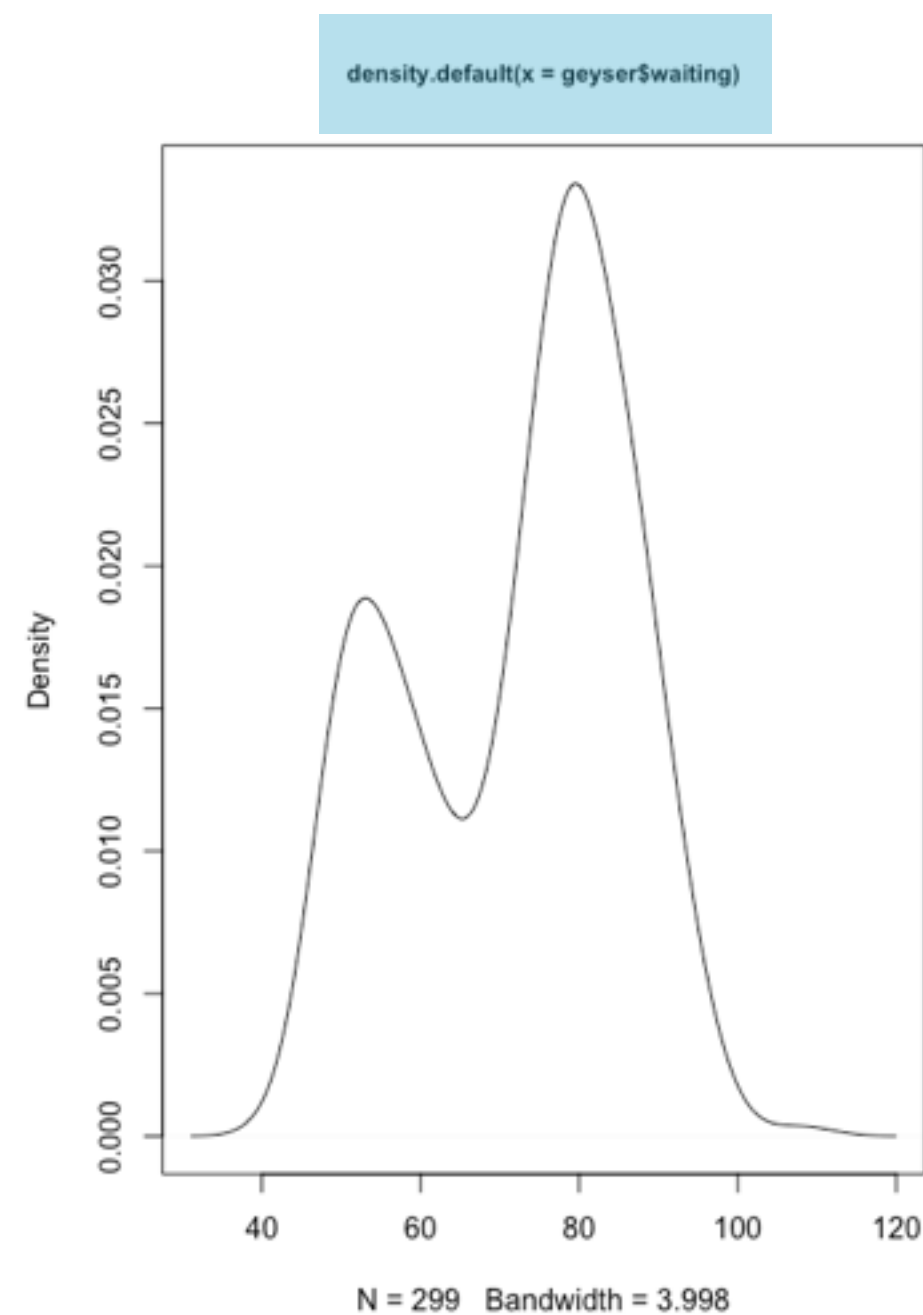
Adding text to plots

Explanatory text

- Axis labels  `xlab()`
`ylab()`
- Titles
- Legends
- Text in plot itself

Overriding default titles

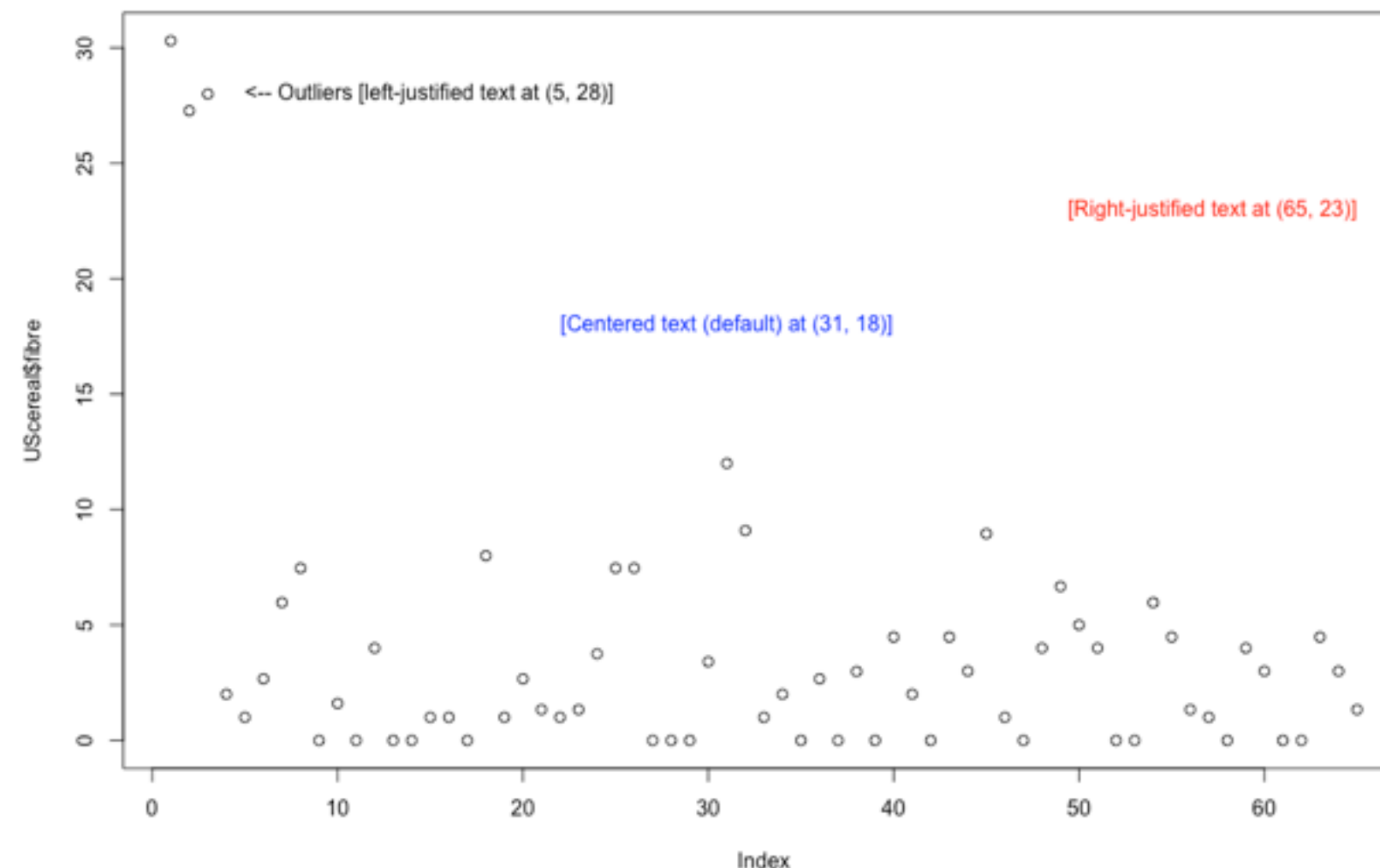
```
> library(MASS)
> par(mfrow = c(1, 2))
> plot(density(geyser$waiting))
> plot(density(geyser$waiting),
      main = "Estimated density: \n Old Faithful waiting times")
```



```
text(x, y, labels, adj)
```

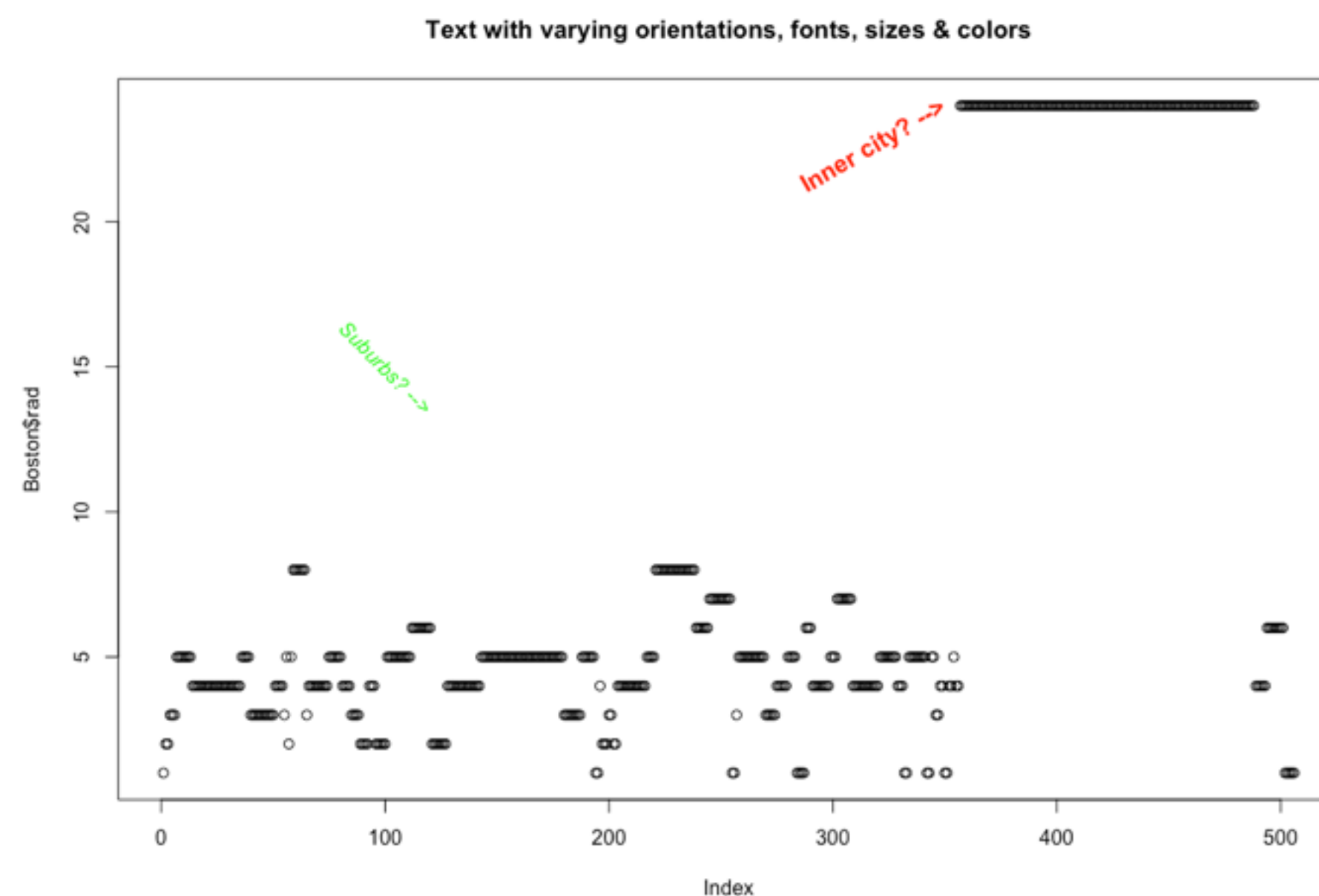
Adding explanatory text to a plot

```
> library(MASS)
> plot(UScereal$fibre)
> text(5, 28, "<-- Outliers [left-justified text at (5, 28)]", adj = 0)
> text(65, 23, "[Right-justified text at (65, 23)]", adj = 1,
      col = "red")
> text(31, 18, "[Centered text (default) at (31, 18)]", col = "blue")
```



Varying fonts, orientations, and other text features

```
> library(MASS)
> plot(Boston$rad)
> text(350, 24, adj = 1, "Inner city? -->", srt = 30, font = 2,
      cex = 1.2, col = "red")
> text(100, 15, "Suburbs? -->", srt = -45, font = 3, col = "green")
> title("Text with varying orientations, fonts, sizes & colors")
```





DATA VISUALIZATION IN R

Let's practice!

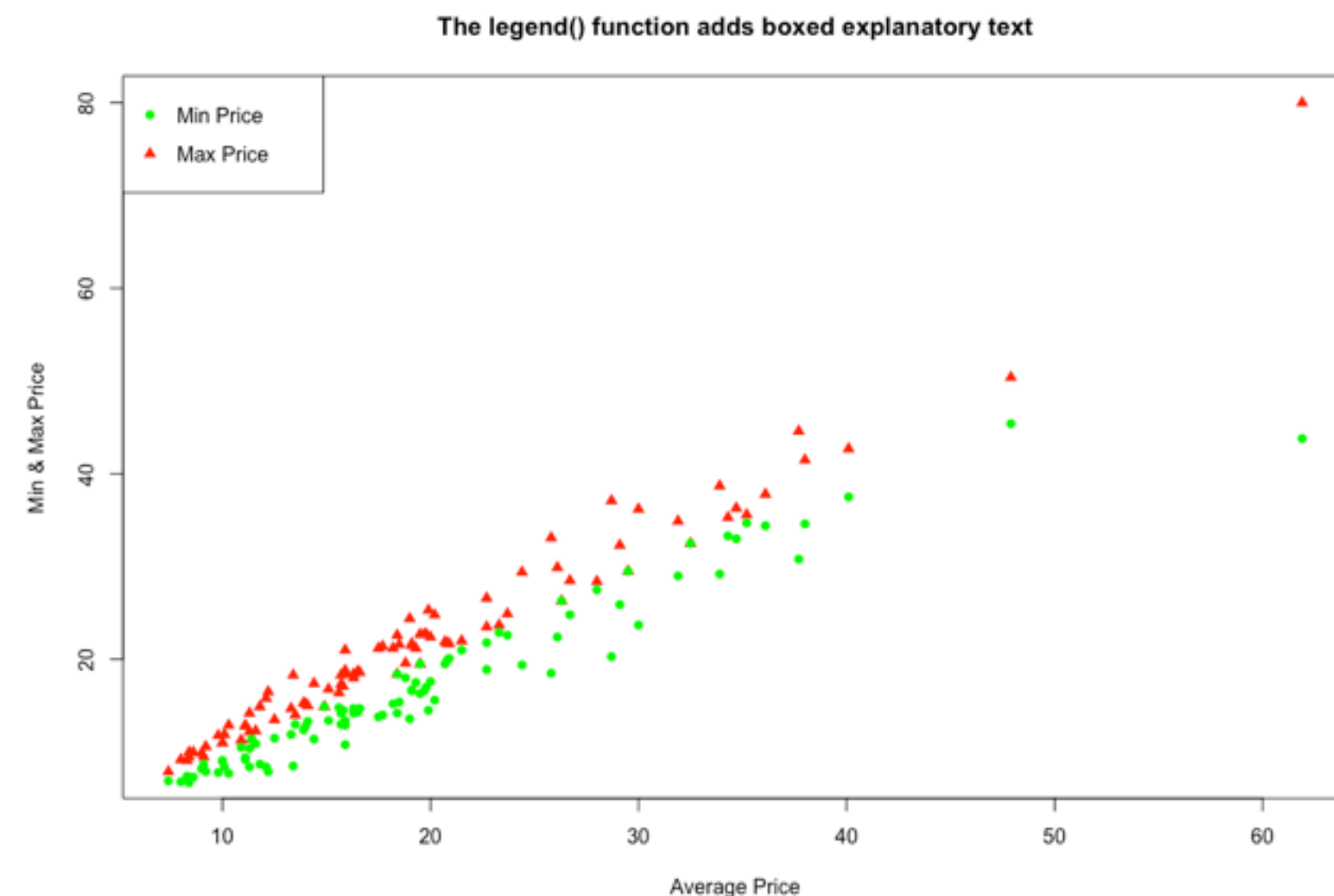


DATA VISUALIZATION IN R

Adding or modifying other plot details

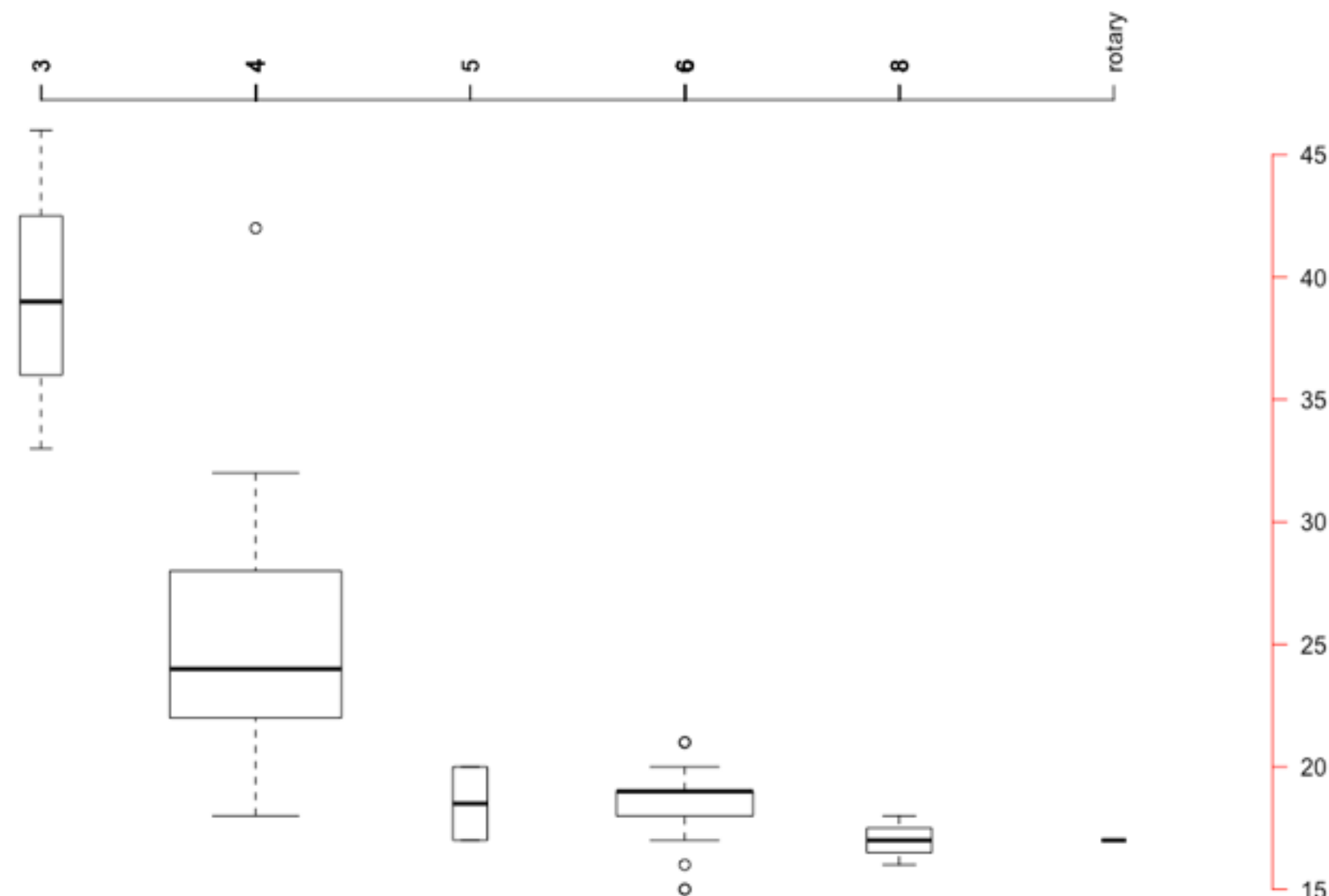
Adding legends to a plot

```
> library(MASS)
> plot(Cars93$Price, Cars93$Max.Price, pch = 17, col = "red",
       xlab = "Average Price", ylab = "Min & Max Price")
> points(Cars93$Price, Cars93$Min.Price, pch = 16, col = "green")
> legend(x = "topleft", pch = c(16, 17), col = c("green", "red"),
       legend = c("Min Price", "Max Price"))
> title("The legend() function adds boxed explanatory text")
```



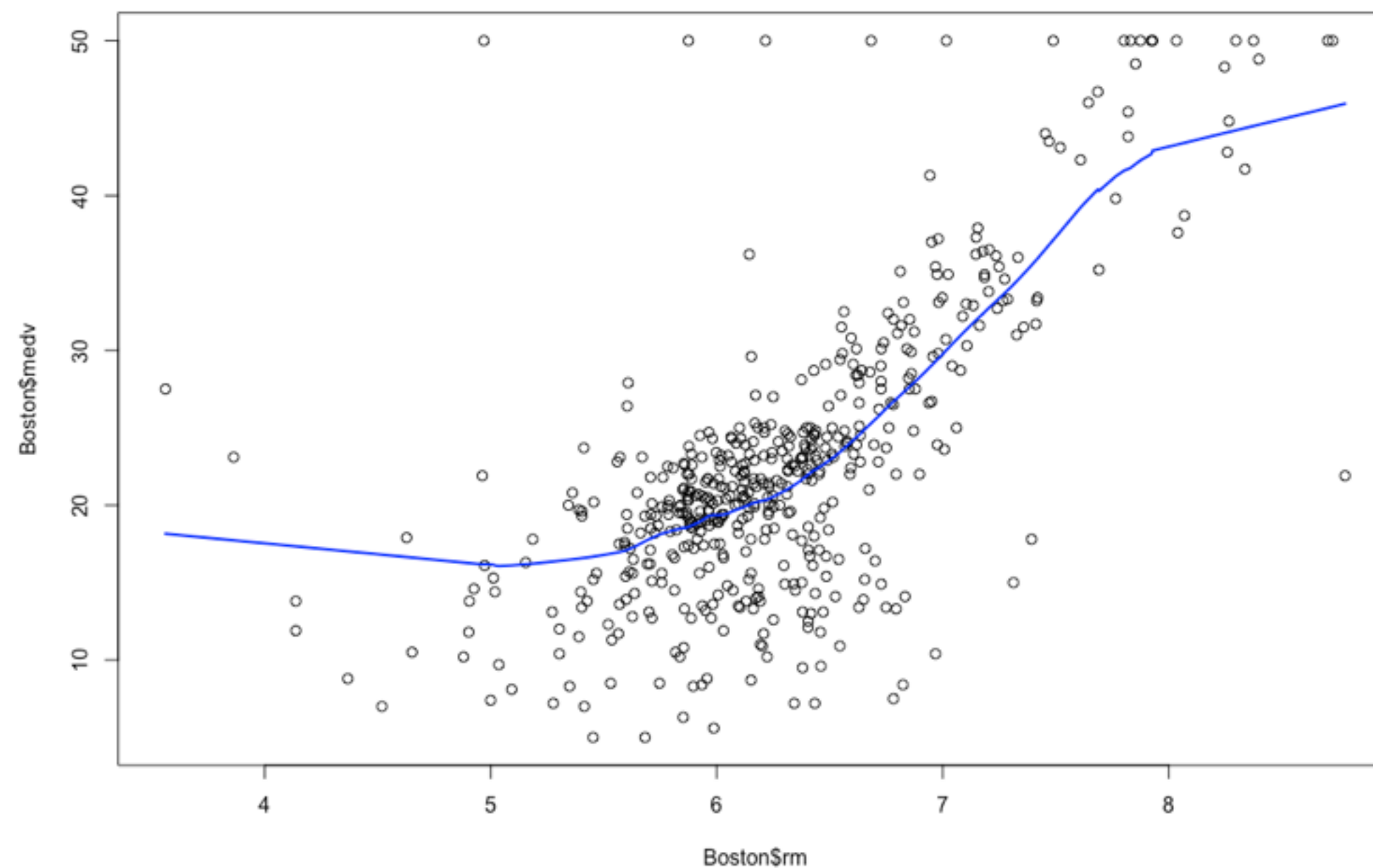
Adding custom axes to a plot

```
> library(MASS)
> boxplot(MPG.city ~ Cylinders, data = Cars93,
          varwidth = TRUE, axes = FALSE)
> axis(side = 3, at = Cars93$Cylinders,
       labels = as.character(Cars93$Cylinders), las = 2)
> axis(side = 4, col = "red", las = 1)
```



Adding a smooth trend line

```
> library(MASS)
> plot(Boston$rm, Boston$medv)
> trend <- supsmu(Boston$rm, Boston$medv)
> lines(trend, lwd = 2, col = "blue")
```





DATA VISUALIZATION IN R

Let's practice!