Intro to R - 4. Base R Plots

OIT/SMU Libraries Data Science Workshop Series

Michael Hahsler

OIT, SMU

World Changers Shaped Here SMU®

1. Simple Plots
2. High-level Graphics Functions
3. Low-level Graphics Functions
4. Exercises

Section 1

Simple Plots

Introduction

- Ploting is an integral part of R.
- R plots on devices (e.g., `X11()`, `quarz()`, `windows()`, `pdf()`, `png()`)
- Plotting commands are divided into three basic groups:
  1. **High-level plotting functions** create a new plot on the graphics device, possibly with axes, labels, titles and so on.
  2. **Low-level plotting functions** add more information to an existing plot, such as extra points, lines and labels.
  3. **Interactive graphics functions** allow you interactively add information to, or extract information from, an existing plot, using a pointing device such as a mouse.

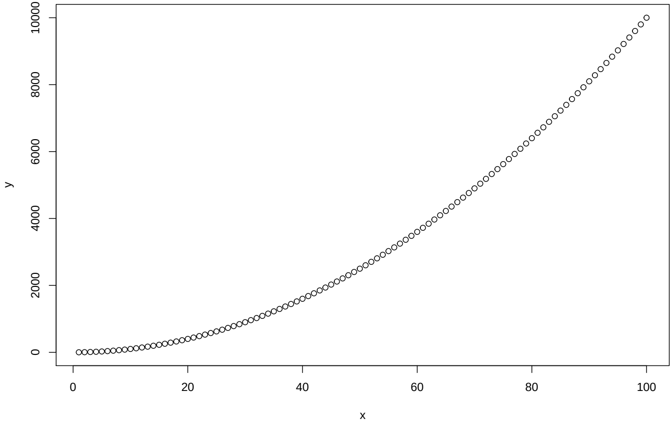
We will only discuss "base" graphics here. A very popular alternative is `ggplot`.

Section 2

High-level Graphics Functions

plot

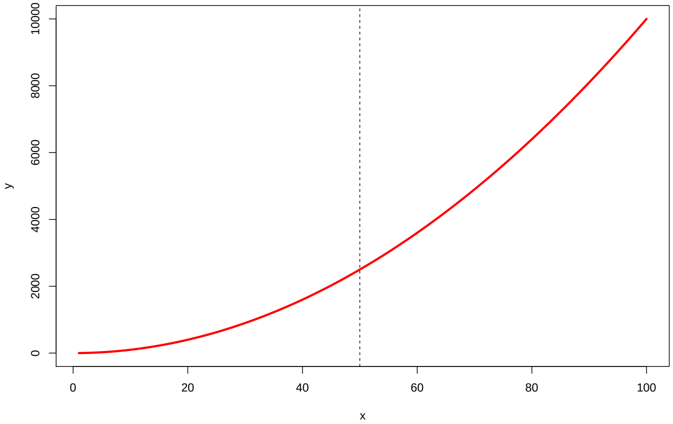
```
x <- 1:100
y <- x^2
plot(x, y)
```



plot

```
# use type l = line, line width 3 and red
plot(x, y, type = "l", lwd = 3, col = "red")

# add a vertical dashed (line type 2) line
abline(v=50, lty=2)       # add a vertical line
```



Getting help for plot

```
? plot
```

Shows that plot is a so called generic function. Generic functions have implementations dor different data types which get “dispatched” at call-time.

```
? plot.default
```

This is the default function for plot.

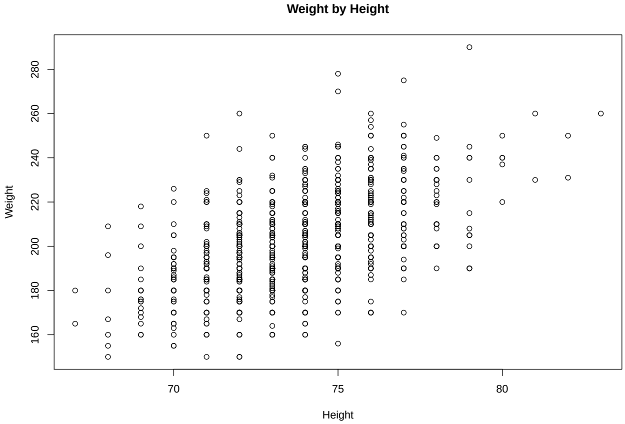
```
? par
```

Graphical parameters which typically can be passed on as ... to plot.

Scatterplot

```
mlb <- read.csv(paste0("https://michael.hahsler.net/SMU/",
  "DS_Workshop_Intro_R/examples/MLB_cleaned.csv"))

plot(x = mlb$Height.inches., y = mlb$Weight.pounds.,
  xlab = "Height", ylab = "Weight",
  main = "Weight by Height")
```



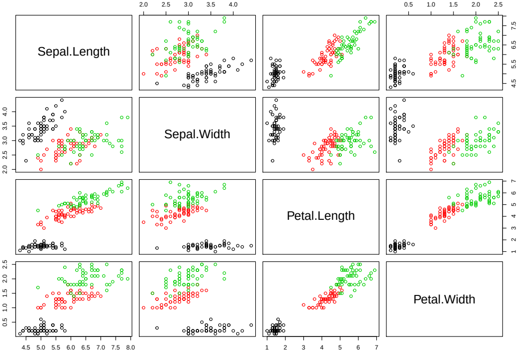
Scatterplot matrix (pairs plot)

plot() is generic which means that it produces different results depending on the type of data. For example, data frames result in a pairwise scatterplot matrix.

```
data(iris)
head(iris, n=1)      # iris is a data.frame
```

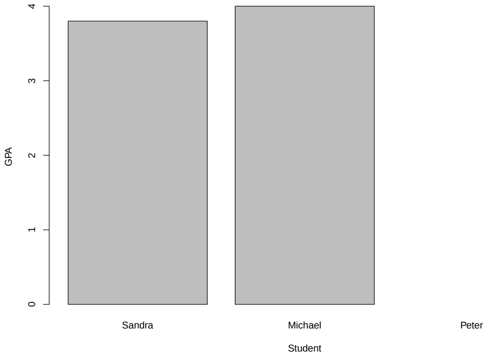
```
##   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
## 1          5.1         3.5          1.4         0.2  setosa
```

```
plot(iris[,-5], col= iris[,5]) # use Species (col 5) for color
```



barplot - Bar Charts

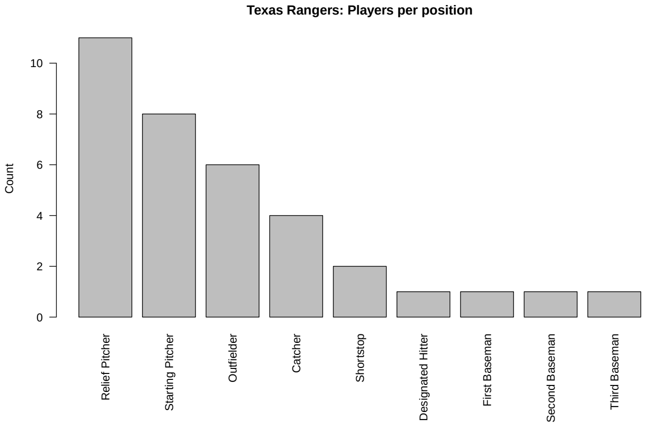
```
gpa <- c(Sandra = 3.8, Michael = 4.0, Peter = NA)
gpa.sort <- sort(gpa, na.last = TRUE)
barplot(gpa.sort, ylab = "GPA", xlab = "Student")
```



barplot - Bar Charts

How many players do the Texas Rangers have for each position?

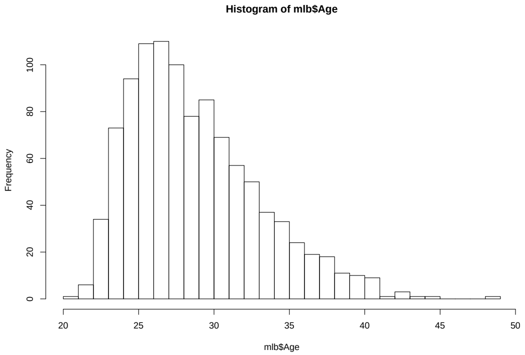
```
mlb_tex <- mlb[mlb$Team == "TEX", ]
# use par to make space for labels see ? par and look for mar (plot margin)
oldpar <- par(mar = c(10, 4, 4, 1) + .1)
barplot(sort(table(mlb_tex$Position), decreasing = TRUE),
  las = 2, ylab = "Count", main = "Texas Rangers: Players per position")
```



```
par(oldpar)
```

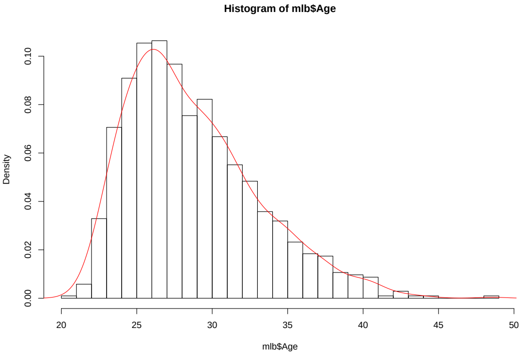
hist - Histogram

```
hist(mlb$Age, breaks=20)
```



hist - Histogram with estimated density

```
hist(mlb$Age, breaks=20, prob=TRUE)
lines(density(mlb$Age), col="red")
```



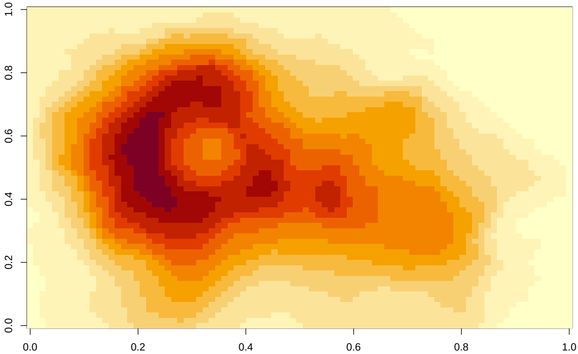
image

volcano is a R data set with elevation measurements of Maunga Whau on a 10m by 10m grid.

```
data(volcano)
dim(volcano)
```

```
## [1] 87 61
```

```
image(volcano)
```



Commmon arguments for plot functions

- `add=TRUE`: add to an existing plot?
- `axes=FALSE`: plot axes?
- `log="x"`, `log="y"` or `log="xy"`: Use a logarithmic scale?
- `type="l"`: plot lines instead of points
- `xlab, ylab`: axis labels
- `main`: figure title
- `sub`: sub-title

Section 3

Low-level Graphics Functions

Some low-level functions

These functions can be used to add elements to a plot.

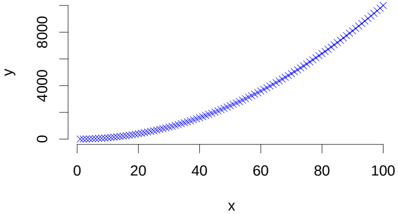
- `points(x, y)`
- `lines(x, y)`
- `text(x, y, labels, ...)`
- `abline(a, b)` or `abline(h=y)` or `abline(v=x)`
- `polygon(x, y, ...)`
- `legend(x, y, legend, ...)`
- `title(main, sub)`
- `axis(side, ...)`

Graphical parameter list: par

R maintains a list of graphics parameters to control line style, colors, figure arrangement and text justification.

A separate list of graphics parameters is maintained for each active device.

```
oldpar <- par(col = "blue", pch=4, cex = 2)
plot(x, y, axes = FALSE)
axis(1); axis(2)
```



```
par(oldpar) # restore the original parameters
```

Many parameters from `par()` can also be passed to `plot()`. Try `par()` and `?par`

Danger

Do not forget to reset par to the original settings!
You can also reset par by closing the plotting device or with the little broom symbol in the Plots tab of RStudio.

Important parameters in par

- `pch=4`: Plotting symbol (0-25)
- `lty=2`: Line type
- `lwd=2`: Line width
- `col=2`: Color for points, lines, etc.
- `cex=1.5`: Character expansion (e.g., 50% larger than default text size)
- `mai=c(1, 0.5, 0.5, 0)`: Widths of the bottom, left, top and right margins, respectively, measured in inches. `mar` is the same, just measured in rows.
- `mfcol, mfrow`: Put multiple plots next to each other.

Saving a plot as an image

```
png(file="plot.png")    # open device
plot(x, y)
dev.off()        # close device
```

```
## pdf
##   2
```

Other devices are ‘jpeg(), tiff(),‘ ‘pdf(), postscript(),‘ ‘win.metafile()‘ (Windows). Use ‘?Devices‘ for a complete list.

Advanced and InteractiveGraphics

The most populare advanced graphics packages are:

- `ggplot2`: Grammar of graphics. Produces elegant visualizations (see http://ggplot2.org/).
- `grid`: Advanced graphics can be programmed using flexible low level ploting functions (viewports, different coordinate systems and units, lines, points, text, etc.) See also package `lattice`.

Interactive Graphics are available via several extension packages. Here are some examples:

https://www.r-graph-gallery.com/interactive-charts.html

Section 4

Exercises

Exercises

1. Plot a ‘sin(x)/x‘. Hint: Trigonometric functions in R use angles in radians (see ‘sin‘)
2. The “cars” data set gives the speed of cars and the distances taken to stop. Note that the data were recorded in the 1920s. Plot the “cars” data set as a scatter plot. Plot all data points with distances taken to stop greater than 80 in red.
3. Plot histograms for speed and dist in “cars”.