Scatter Diagrams

Intro to Stats, Spring 2017

Prof. Gaston Sanchez

Learning Objectives

- How to use `plot()` to create scatter diagrams
- Adding points with `points()`
- Adding lines with `abline()`
- How to use `ggplot()` to create scatter diagrams

Introduction

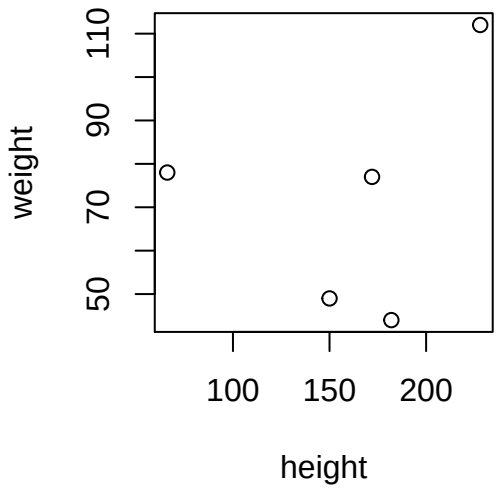
The easiest way to plot scatter diagrams in R is with the `plot()` function. I should say that `plot()` produces different kinds of plots depending on the type of input(s) that you pass to it.

If you pass two numeric variables (i.e. two R vectors) `x` and `y`, `plot()` will produce a scatter diagram. For example, consider the `height` and `weight` variables of the following toy data table:

name	sex	height	weight
Luke	male	172.00	77.00
Leia	female	150.00	49.00
Obi-Wan	male	182.00	44.00
Yoda	male	66.00	78.00
Chebacca	male	228.00	112.00

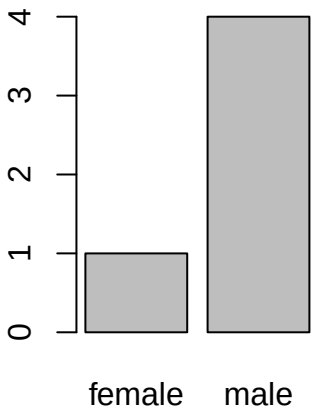
To make a scatter diagram with `height` and `weight`, you can create two vectors and pass them to `plot()`:

```
height = c(172, 150, 182, 66, 228)  # in centimeters
weight = c(77, 49, 44, 78, 112)     # in kilograms

# default scatter diagram
plot(height, weight)
```



If you pass a factor to `plot()` it will produce a bar-chart:

```
# qualitative variable (as an R factor)
sex = factor(c('male', 'female', 'male', 'male', 'male'))

# default scatter diagram
plot(sex)
```

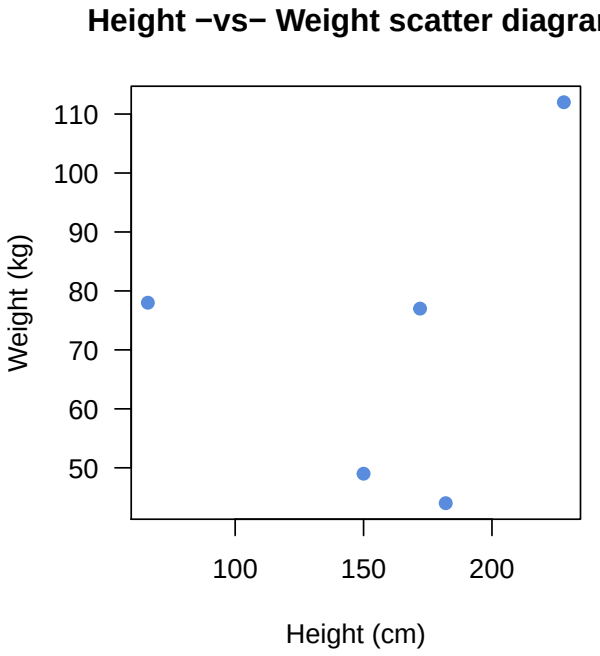


Note that `plot()` displays a very simple, and kind of ugly, scatter diagram. This not an accident. In fact, the basic plots in R follow a "quick and dirty" approach. They are not publication quality, but that is OK. The default display of `plot()` was not designed to produce pretty graphics, but rather to produce visualizations that quickly allow you to explore the data, identify patterns, help you ask new research questions, and then move on with more visualizations or to the next analytical stages.

Although `plot()` produces a basic graph, you can use several arguments, or graphical parameters, to obtain a nicer chart. To find more information about the available graphical parameters for `plot()`, take a look at the documentation provided by `help(plot)`.

The following code uses various graphical parameters to display a more visually appealing scatter diagram:

```
# nicer scatter diagram
plot(height, weight,
     las = 1,          # orientation of y-axis tick marks
     pch = 19,         # filled dots
     col = '#598CDD', # color of dots
     xlab = 'Height (cm)',    # x-axis label
     ylab = 'Weight (kg)',    # y-axis label
     main = 'Height -vs- Weight scatter diagram')
```



Adding points and lines

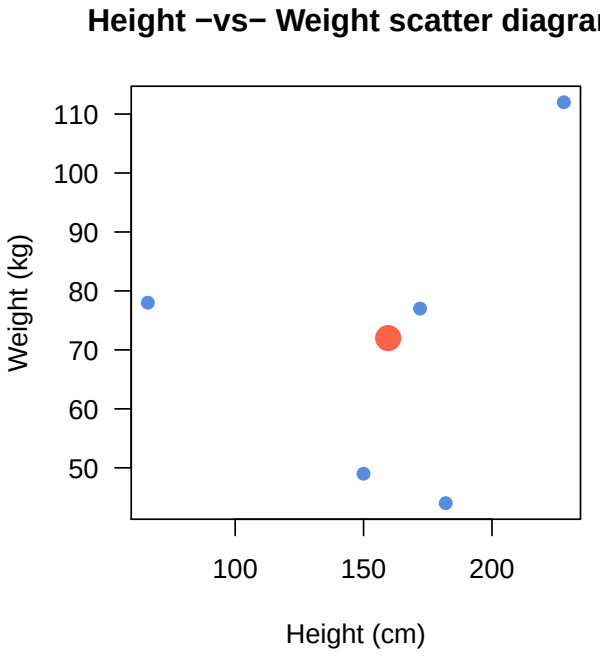
Often, you may want to add more points and/or line(s) to a given plot. When you use `plot()`, you add points with `points()`, and lines with `abline()`.

For example, say you want to add the point of averages. First, get the averages:

```
avg_height = mean(height)
avg_weight = mean(weight)
```

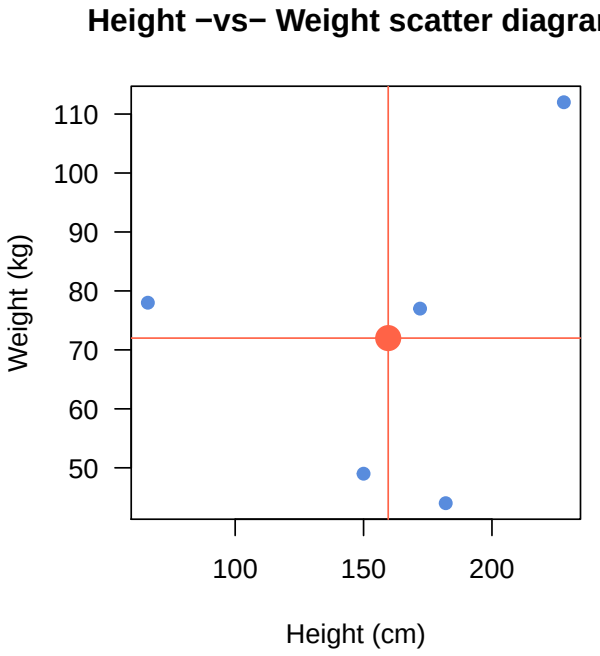
Once you have the coordinates of the point of averages, you can `plot()` again the scatter diagram, adding the point of averages with `points()`:

```
# scatter diagram
plot(height, weight,
     las = 1,          # orientation of y-axis tick marks
     pch = 19,         # filled dots
     col = '#598CDD', # color of dots
     xlab = 'Height (cm)',    # x-axis label
     ylab = 'Weight (kg)',    # y-axis label
     main = 'Height -vs- Weight scatter diagram')
# point of averages
points(avg_height, avg_weight, pch = 19, cex = 2, col = "tomato")
```



Another common task involves adding one or more lines to a scatter diagram produced by `plot()`. One option to achieve this task is via the `abline()` function. Here's an example showing the previous scatter diagram, with two guide lines corresponding to the point of averages

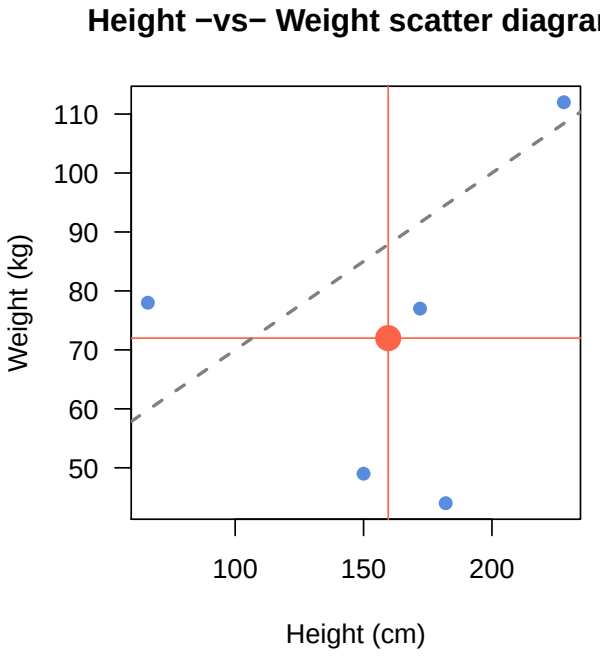
```
# scatter diagram
plot(height, weight,
     las = 1,          # orientation of y-axis tick marks
     pch = 19,         # filled dots
     col = '#598CDD', # color of dots
     xlab = 'Height (cm)',    # x-axis label
     ylab = 'Weight (kg)',    # y-axis label
     main = 'Height -vs- Weight scatter diagram')
# guide lines for point of avgs
abline(h = avg_weight, v = avg_height, col = "tomato")
# point of averages
points(avg_height, avg_weight, pch = 19, cex = 2, col = "tomato")
```



The argument `h` is used to specify the y-value for *horizontal* lines; the argument `v` is used to specify the x-value for *vertical* lines.

If what you want is to specify a line with intercept `a` and slope `b`, then specify these arguments inside `abline()`:

```
# scatter diagram
plot(height, weight,
     las = 1,          # orientation of y-axis tick marks
     pch = 19,         # filled dots
     col = '#598CDD', # color of dots
     xlab = 'Height (cm)',    # x-axis label
     ylab = 'Weight (kg)',    # y-axis label
     main = 'Height -vs- Weight scatter diagram')
# guide lines for point of avgs
abline(h = avg_weight, v = avg_height, col = "tomato")
# line with intercep and slope
abline(a = 40, b = 0.3, col = "gray50", lty = 2, lwd = 2)
# point of averages
points(avg_height, avg_weight, pch = 19, cex = 2, col = "tomato")
```



Scatter diagrams with `ggplot2`

Another approach to create scatter diagrams in R is to use functions from the package `"ggplot2"`. This package provides a different philosophy to define graphs, and it also produces plots with visual attributes carefully chosen to provide prettier plots.

You should have the package `"ggplot2"` already installed, since you were supposed to use it for HW02. Assuming that this is the case, you need to load `"ggplot2"` with the function `library()` in order to start using its functions:

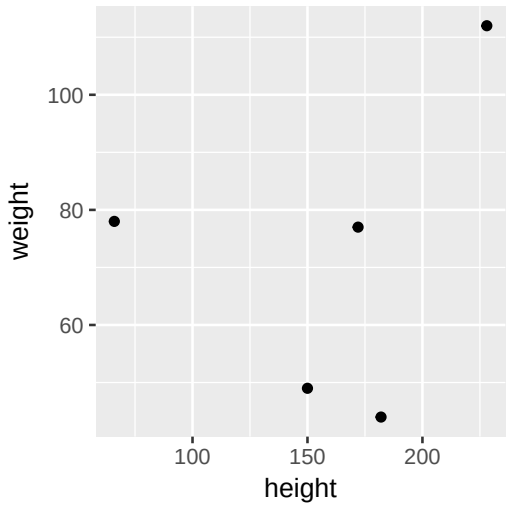
```
# load ggplot2
library(ggplot2)
```

One of the major differences between basic plots—like those produced by `plot()`—and graphics with `ggplot()`, is that the latter requires the data to be in the form of a data frame:

```
dat = data.frame(
  name = c('Luke', 'Leia', 'Obi-Wan', 'Yoda', 'Chewbacca'),
  sex = c('male', 'female', 'male', 'male', 'male'),
  height = c(172, 150, 182, 66, 228),
  weight = c(77, 49, 44, 78, 112)
)
```

To create a scatter diagram with `"ggplot2"`, type the following commands:

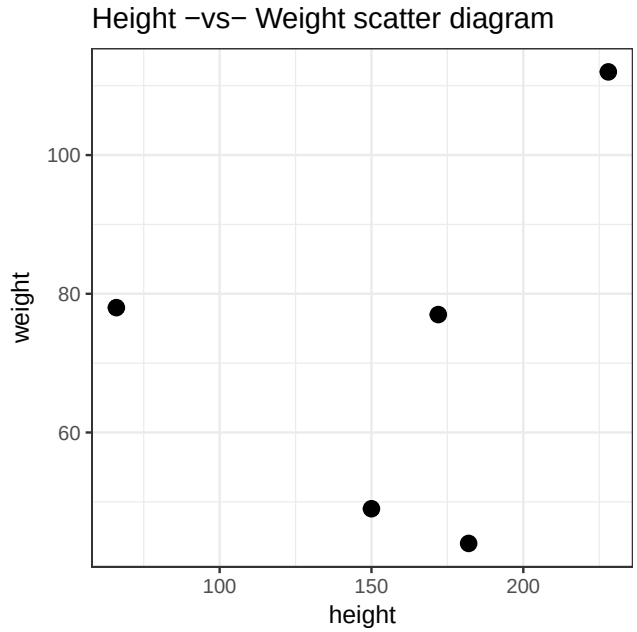
```
ggplot(data = dat, aes(x = height, y = weight)) +
  geom_point()
```



- The main input of `ggplot()` is `data` which takes the name of the data frame containing the variables.
- The `aes()` function—inside `ggplot()`—allows you to specify which variables will be used for the `x` and `y` positions.
- The `+` operator is used to add a *layer*, in this case, the layer corresponds to `geom_point()`
- The function `geom_point()` specifies the type of geometric object to be displayed: points (since we want a scatter diagram with dots).

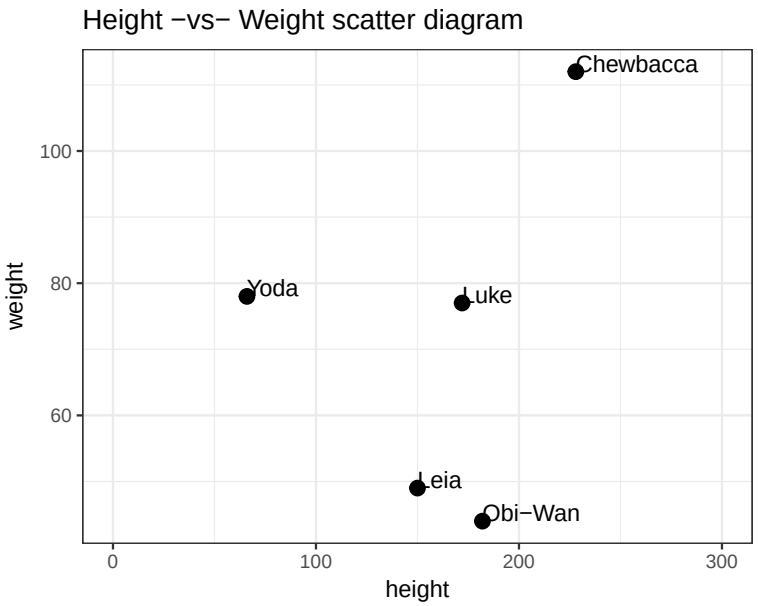
As you can tell, the default chart produced by `ggplot()` is nicer than the one produced with `plot()`. You can customize the previous graph to add more details:

```
ggplot(data = dat, aes(x = height, y = weight)) +
  geom_point(size = 3) +
  theme_bw() +
  ggtitle("Height -vs- Weight scatter diagram")
```



Here's another example of a scatter diagram that includes labels for each dot:

```
ggplot(data = dat, aes(x = height, y = weight)) +
  geom_point(size = 3) +
  geom_text(aes(label = name), hjust=0, vjust=0) +
  xlim(0, 300) +
  theme_bw() +
  ggtitle("Height -vs- Weight scatter diagram")
```



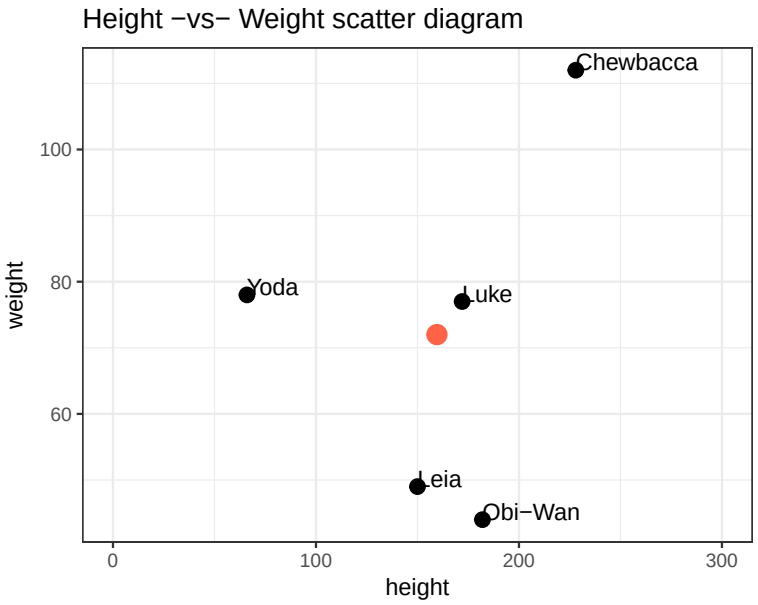
Adding specific points with `ggplot()` is a bit trickier. This is because you need to provide data to `ggplot()` in the form of a data.frame. In order to plot the point of averages with `ggpot()`, we need to create a data frame for such a point:

```
# data frame for the point of averages
avgs = data.frame(height = avg_height, weight = avg_weight)
avgs
```

```
##   height weight
## 1  159.6     72
```

One way to add the point of averages is to use `geom_point()` twice: one for the heighths and weights of the individuals, and the second time for the point of averages:

```
ggplot(data = dat, aes(x = height, y = weight)) +
  geom_point(size = 3) +
  geom_point(data = avgs, aes(x = height, y = weight),
             col = "tomato", size = 4) +
  geom_text(aes(label = name), hjust=0, vjust=0) +
  xlim(0, 300) +
  theme_bw() +
  ggtitle("Height -vs- Weight scatter diagram")
```



Finally, here's how to add guide lines for the point of averages:

```
ggplot(data = dat, aes(x = height, y = weight)) +
  geom_point(size = 3) +
  geom_point(data = avgs, aes(x = height, y = weight),
             col = "tomato", size = 4) +
  geom_vline(xintercept = avg_height, col = 'tomato') +
  geom_hline(yintercept = avg_weight, col = 'tomato') +
  geom_text(aes(label = name), hjust=0, vjust=0) +
  xlim(0, 300) +
  theme_bw() +
  ggtitle("Height -vs- Weight scatter diagram")
```