

Quantitative Methods in Geography

Geo 340

William D. McCoy

Department of Geosciences
University of Massachusetts, Amherst

Autumn, 2009

Exploring bivariate data

- ▶ parallel boxplots
- ▶ superposed density plots
- ▶ quantile-quantile plots
- ▶ scatterplots
- ▶ correlation
- ▶ regression

The sleep data

The sleep data provided in **R** are the results of an experiment with two soporific drugs taken by student volunteers. The students were divided into two groups of ten students each and each group was given a different drug. The variables named `extra` gives the increased hours of sleep for each individual in the experiment. The variable `group` indicates to which group each individual belongs and, therefore, which drug was taken.

Type the name of the data object to have a look at the data:

```
> sleep
```

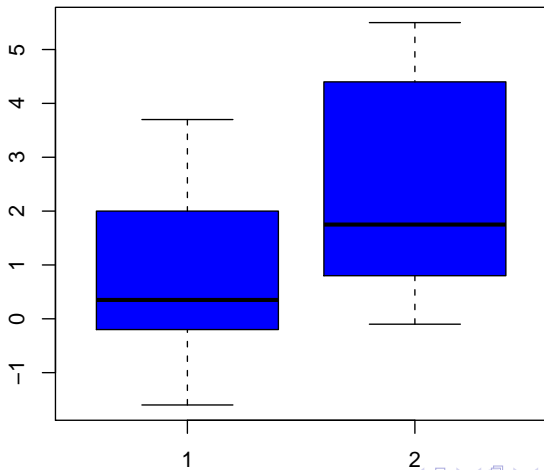
	extra	group
1	0.7	1
2	-1.6	1
3	-0.2	1
4	-1.2	1
5	-0.1	1
6	3.4	1
7	3.7	1
8	0.8	1
9	0.0	1
10	2.0	1
11	1.9	2
12	0.8	2
13	1.1	2
14	0.1	2
15	-0.1	2
16	4.4	2
17	5.5	2
18	1.6	2
19	4.6	2
20	3.4	2

Parallel boxplots

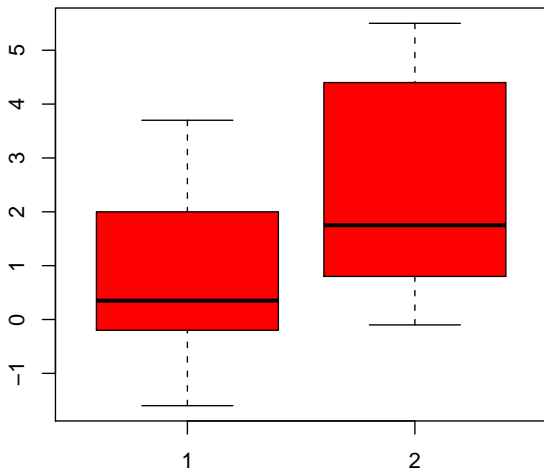
In order to compare the effects of the two drugs on the students, we can use parallel boxplots to compare summaries of the extra hours of sleep for the two groups. It appears that the drug taken by the second group is more effective.

There are several ways to produce parallel boxplots in **R**. Here are two different ways. Both of the methods shown here rely on using `split()` to break the vector `extra` into two vectors – one for each group.

```
> attach(sleep)
> boxplot(split(extra, group), col = "blue")
> detach(sleep)
```



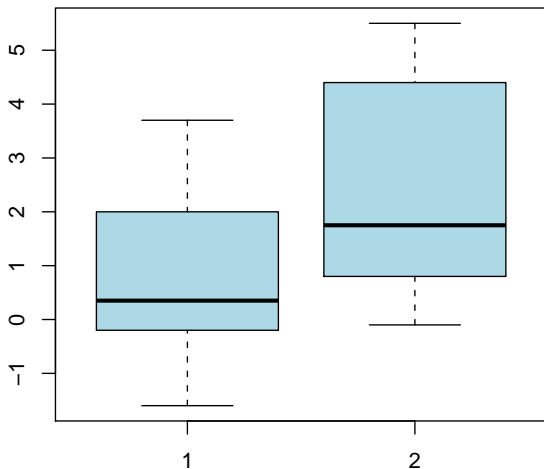
```
> with(sleep, boxplot(split(extra, group), col = "red"))
```



Using a formula argument to `boxplot()`

But perhaps the easiest way to use `boxplot()`, and many other **R** plotting functions, is to provide a formula as the first argument. This avoids the need for `attach()` or `with()` or complex indexing. You can build a formula argument by placing the name of the dependent variable on the left-hand side, then a tilde, then the name(s) of the independent variable(s) on the right-hand side. We will also use formula arguments when we build linear models in **R**, such as regression models and analysis of variance models.

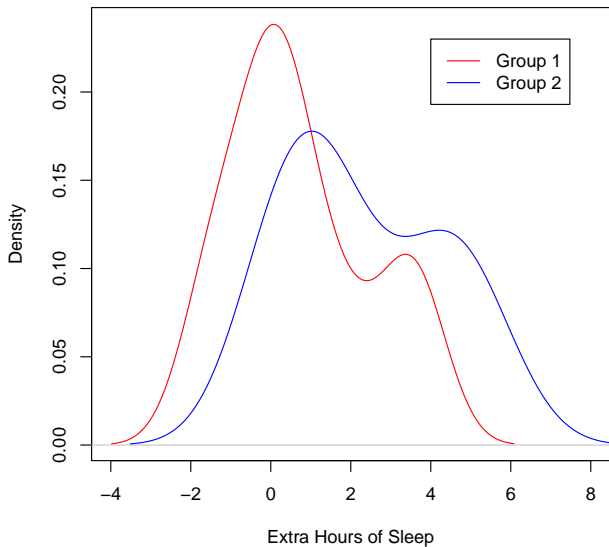
```
> boxplot(extra ~ group, data = sleep, col = "light blue")
```



Superposed density plots

We can also compare the empirical distributions of the two groups by making density plots of each on a single set of axes.

```
> with(sleep, plot(density(extra[group == 1]),  
+   xlim = c(-4, 8),  
+   xlab = "Extra Hours of Sleep", main = "",  
+   col = "red"))  
> with(sleep, lines(density(extra[group == 2]),  
+   col = "blue"))  
> legend(4, .23, c("Group 1", "Group 2"),  
+   col = c("red", "blue"), lty = 1)
```

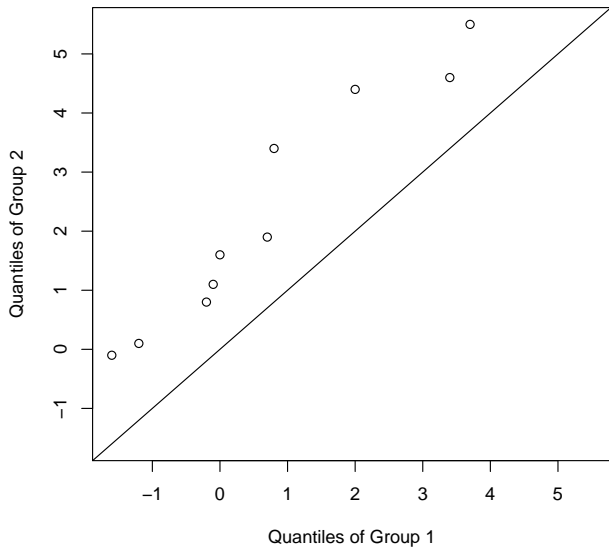


Quantile-quantile plots

To compare two distributions directly, we can also plot the quantiles of one empirical distribution against the quantiles of another. Here we can compare the quantiles of extra sleep of Group 1 against those of Group 2. We can add a line that shows where the points would plot if the distributions were identical.

```
> with(sleep, qqplot(extra[group == 1], extra[group == 2],  
+   xlim = range(extra), ylim = range(extra),  
+   xlab = "Quantiles of Group 1",  
+   ylab = "Quantiles of Group 2",  
+   main = "Q-Q Plot of Extra Sleep"))  
> abline(c(0, 1))
```

Q-Q Plot of Extra Sleep



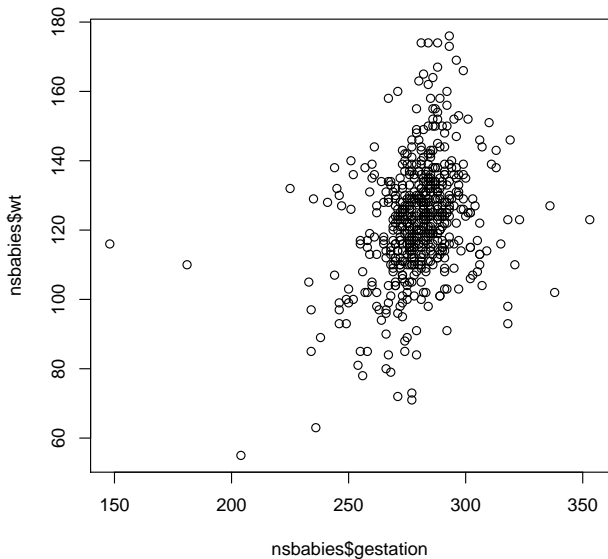
Scatterplots

To compare two variables of naturally paired data, we can use a scatterplot to see how the variable might be related. From the babies dataset in `UsingR`, we can make a subset of the data containing just the gestation time (in days) and the birth weight of babies (in ounces) of non-smoking mothers:

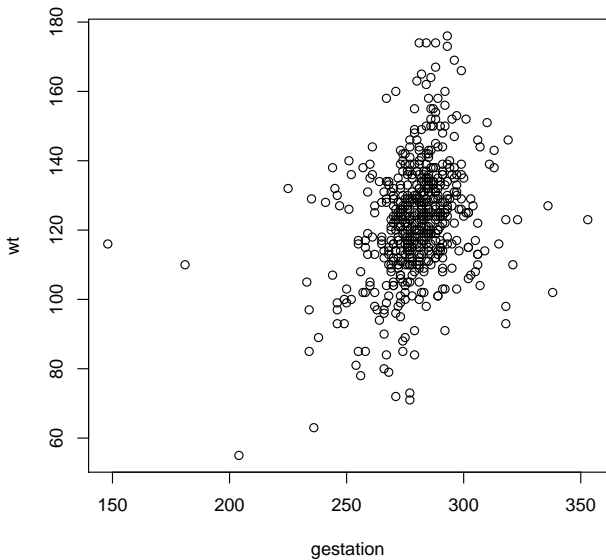
```
> library(UsingR)
> nsbabies <- subset(babies,
+   smoke == 0 & gestation != 999 & wt != 999,
+   select = c(gestation, wt))
```

Here are two different ways to use `plot()` to make a scatterplot of these data. The second method is easier and will cause less confusion when you try to add a regression line to a scatterplot.

```
> plot(nsbabies$gestation, nsbabies$wt)
```



```
> plot(wt ~ gestation, data = nsbabies)
```



Correlation

We can test the extent to which the variation in one variable is related to the variation in another associated variable by using correlation. The function `cor()` in **R** calculates the correlation coefficient between two related variables. We can examine the association between birth weight and gestation period in non-smoking mothers:

```
> cor(nsbabies$wt, nsbabies$gestation)
[1] 0.2974903
```

The correlation coefficient can vary between -1 and $+1$. A correlation coefficient of 0 means there is no correlation between the two variables. A negative coefficient suggests that one variable increases as the other decreases. A positive coefficient indicates that the two variables tend to increase or decrease together.

Regression

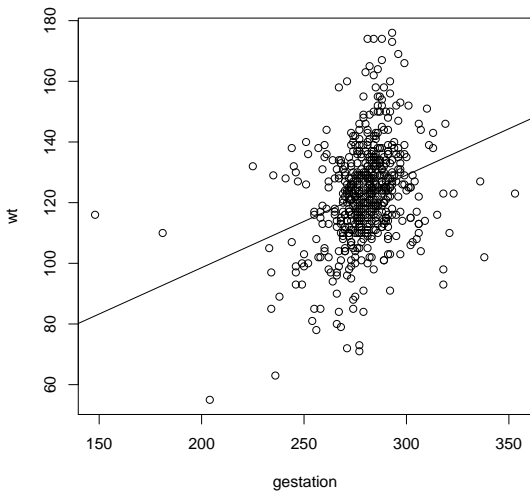
In regression we try to express the relationship between a dependent variable (or response variable) and one or more independent variables (or predictor variables). In simple linear regression there is just one predictor variable and we can express the general form of the equation we are trying to find as:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

This is a linear equation in which β_0 is the y -intercept and β_1 is the slope of the “best-fit” line to the data. We can use the `lm()` function in **R** to construct such a linear model:

```
> nsWtGest.lm <- lm(wt ~ gestation, data = nsbabies)
```

```
> plot(wt ~ gestation, data = nsbabies)
> abline(nsWtGest.lm)
```



The “best-fit” line in simple linear (least-squares) regression is the line that minimizes the sum of the squares of the residuals. The residuals (e_i) are the differences between the actual and predicted values of the response variable:

$$e_i = y_i - \hat{y}_i$$

Values of the response variable that are far from the mean (\bar{y}) have a strong influence on the slope of the regression line. Several types of resistant regression have been devised in order to produce regression lines that are less influenced by such points.

Resistant regression

The package MASS in **R** has several functions for carrying out resistant regression. The function `lqs()` implements several methods of which “lts” (“least trimmed squares”) is the default and is generally a good choice. This method minimizes the sum of the squares of the m smallest residuals (where, by default, m is roughly $n/2$). We can compare a resistant regression line from method “lts” with our previous least-squares line.

```
> library(MASS)
> nsWtGest.lqs <- lqs(wt ~ gestation, data = nsbabies)
```

```
> plot(wt ~ gestation, data = nsbabies)
> abline(nsWtGest.lm); abline(nsWtGest.lqs, lty = 2)
```

