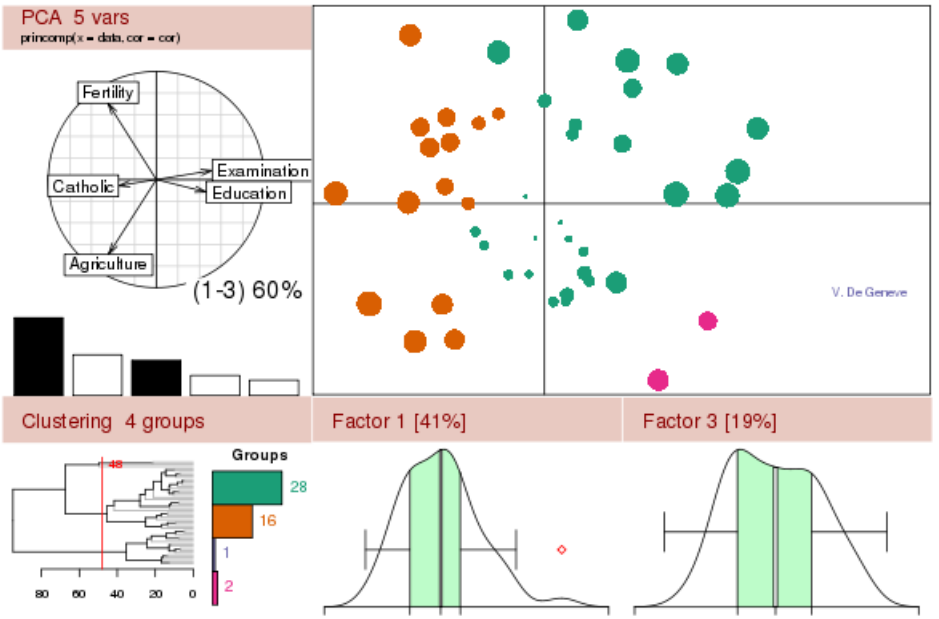
Imperial College London

diamond

A short introduction to the R statistical programming language



Diamond Light Source - April 2013 – PART 3

James Foadi

MPL, Imperial College London - Diamond Light Source Ltd, Oxfordshire

R for statistical modeling

In order to draw inferences about data we need, first to understand what it is meant by “data” in statistics. The preferred data structure used in R for this purpose is the so-called “data frame”. It is essentially a regular table with m rows and n columns (like a matrix). Columns are named; they represent statistical variables. Rows can be named and represent the value taken by all statistical variables involved in a specific instance. When a value is not known (missing data case), we find the
NA symbol in its place.

R for statistical modeling

There are many examples of data frames stored in R for the purpose of testing, comparison or teaching. They are all collected in the package "datasets". This is automatically loaded when we start R. Other packages, too, can come with their own datasets.

Let's practice with cars, a dataset showing speed and stopping distance of 50 cars (try "?datasets").

```
> data(cars)
> cars[1:3,]
  speed dist
1     4    2
2     4   10
3     7    4
> class(cars)
[1] "data.frame"
```

How to load built-in data

With "class" you can find out the nature of an R object

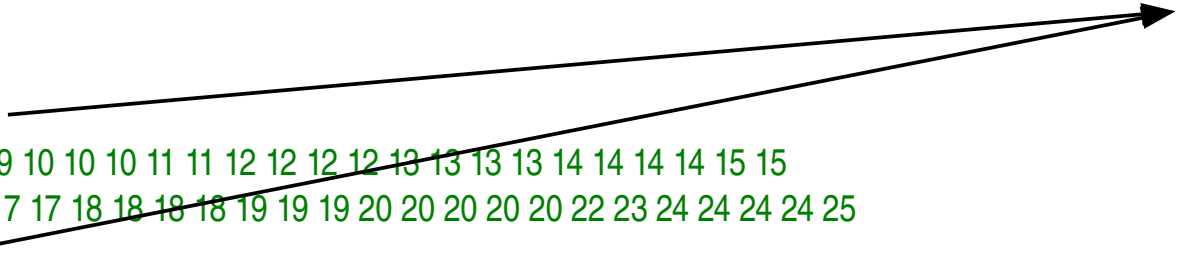
R for statistical modeling

Data frame can be explored with "str":

```
> str(cars)
'data.frame':    50 obs. of  2 variables:
 $ speed: num  4 4 7 7 8 9 10 10 10 11 ...
 $ dist : num  2 10 4 22 16 10 18 26 34 17 ...
```

Then we can play with it:

```
> dim(cars)
[1] 50  2
> cars[1,]
  speed dist
1     4    2
> cars[,1]
 [1]  4  4  7  7  8  9 10 10 10 11 11 12 12 12 12 13 13 13 13 14 14 14 14 15 15
[26] 15 16 16 17 17 17 18 18 18 18 19 19 19 20 20 20 20 20 22 23 24 24 24 24 25
> length(cars[,1])
[1] 50
> cars$speed
 [1]  4  4  7  7  8  9 10 10 10 11 11 12 12 12 12 13 13 13 13 14 14 14 14 15 15
[26] 15 16 16 17 17 17 18 18 18 18 19 19 19 20 20 20 20 20 22 23 24 24 24 24 25
> cars$dist
 [1]  2 10  4 22 16 10 18 26 34 17 28 14 20 24 28 26 34 34 46
[20] 26 36 60 80 20 26 54 32 40 32 40 50 42 56 76 84 36 46 68
[39] 32 48 52 56 64 66 54 70 92 93 120 85
```



Operator $ to select specific columns. Valid only for data frames.

R for statistical modeling

Very interesting and effective is the idea of *conditional selection.* Suppose we want to work only with cars whose speed is less than 10 miles per hour:

```
> cars[cars$speed < 10,]
  speed dist
1     4    2
2     4   10
3     7    4
4     7   22
5     8   16
6     9   10
> scars1 <- cars[cars$speed < 10,]
> class(scars1)
[1] "data.frame"
> dim(scars1)
[1] 6 2
```

R for statistical modeling

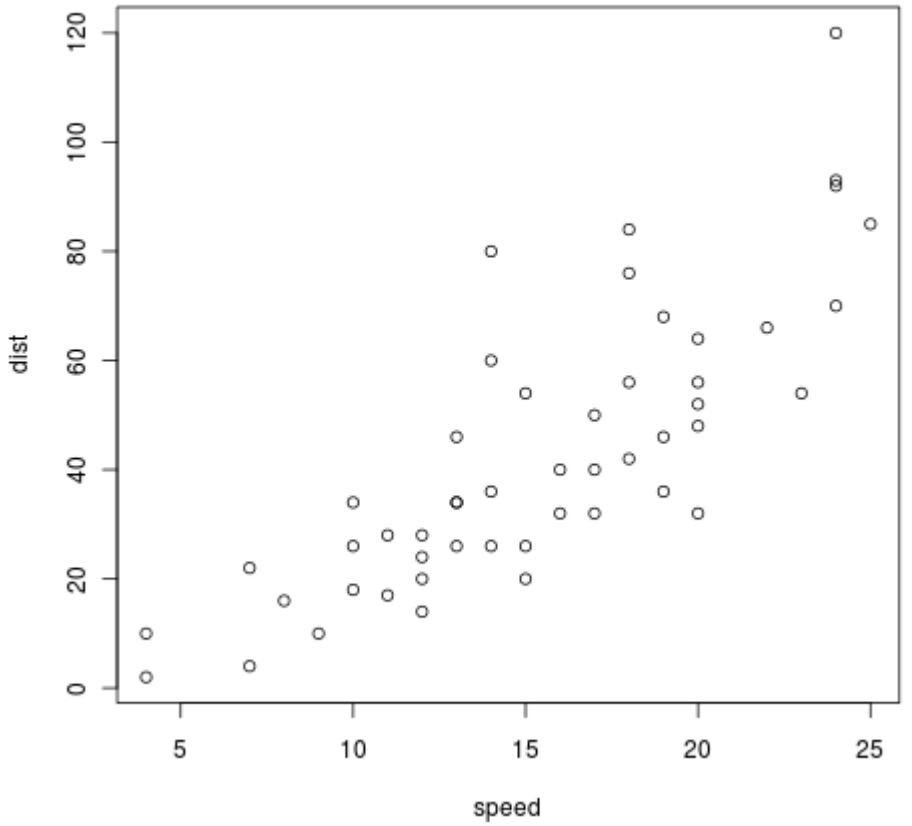
Even more complicated, speed between 5 and 20 miles per hour and breaking distance greater than 15 meters:

```
> scars2 <- cars[cars$speed >= 5 & cars$speed <= 20 & cars$dist > 15,]
> dim(scars2)
[1] 38  2
> scars2[1:5,]
  speed dist
4     7   22
5     8   16
7    10   18
8    10   26
9    10   34
> scars2$dist <= 15
 [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[13] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[25] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[37] FALSE FALSE
```

R for statistical modeling

Things become interesting when data visualization is considered:

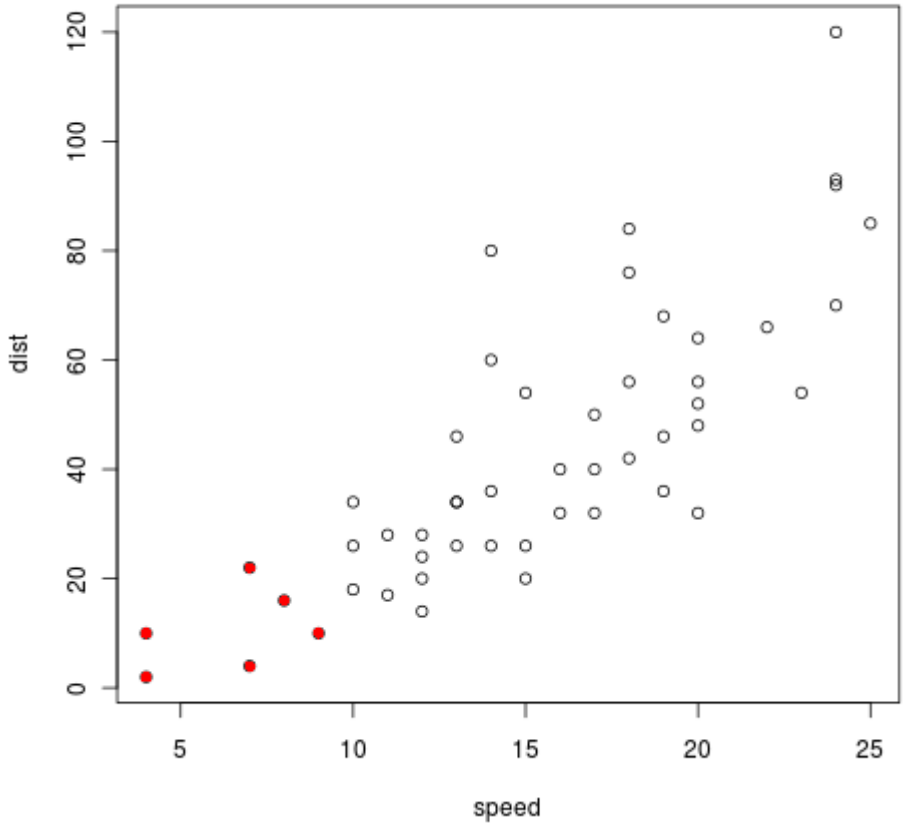
```
> plot(cars)
```



R for statistical modeling

Let's colour the selection represented by scars1 in red:

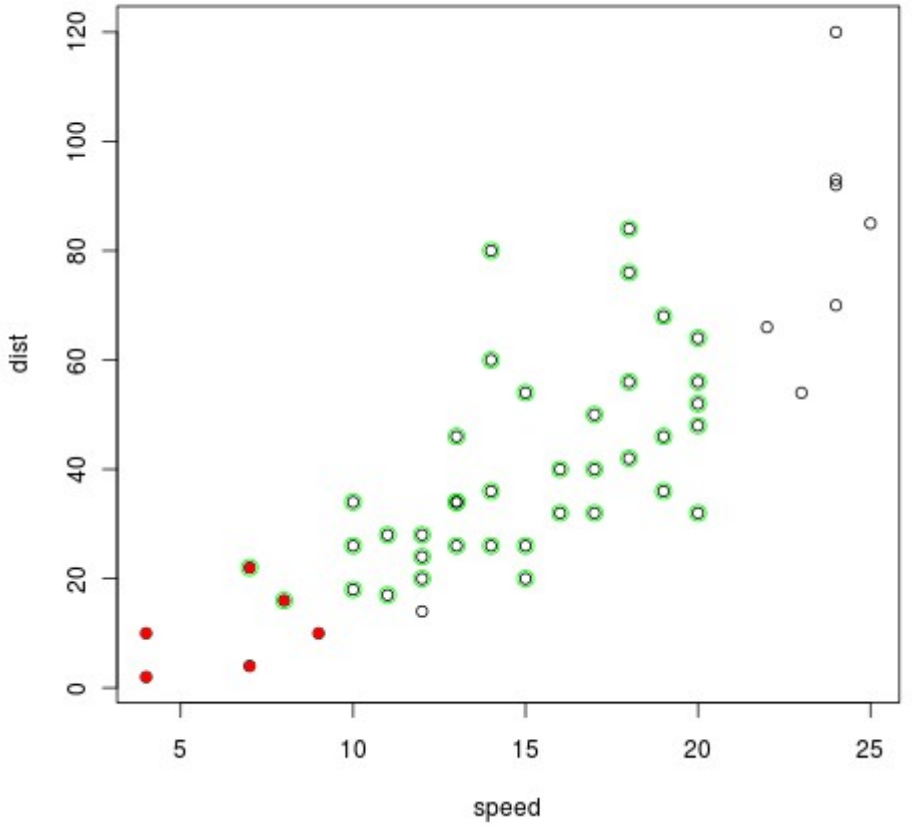
```
> points(scars1,pch=16,col=2)
```



R for statistical modeling

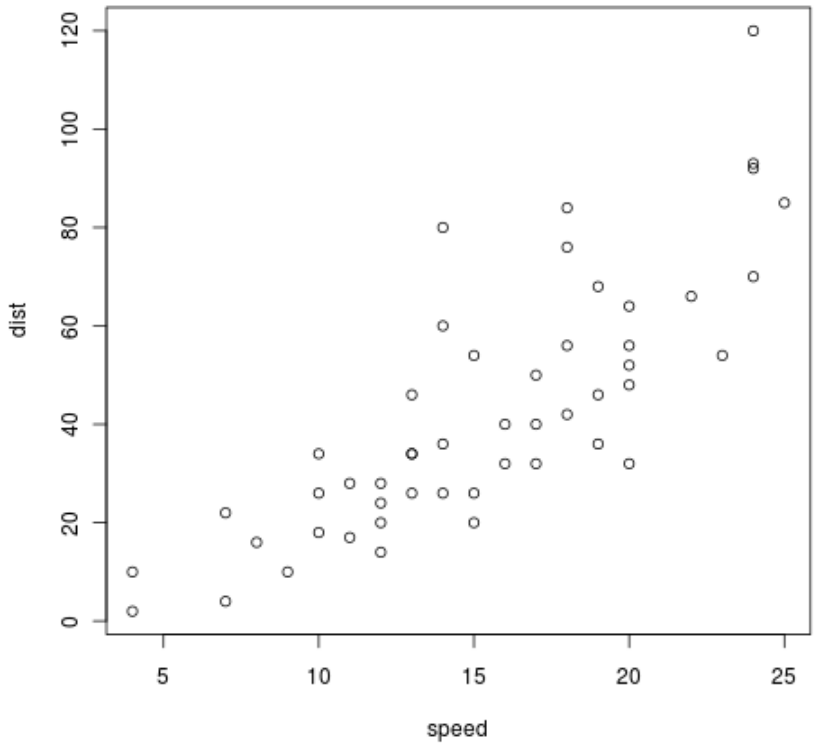
For the selection represented by scars2 let's use a bigger empty green circle:

```
> points(scars2,cex=1.5,col="green")
```



R for statistical modeling

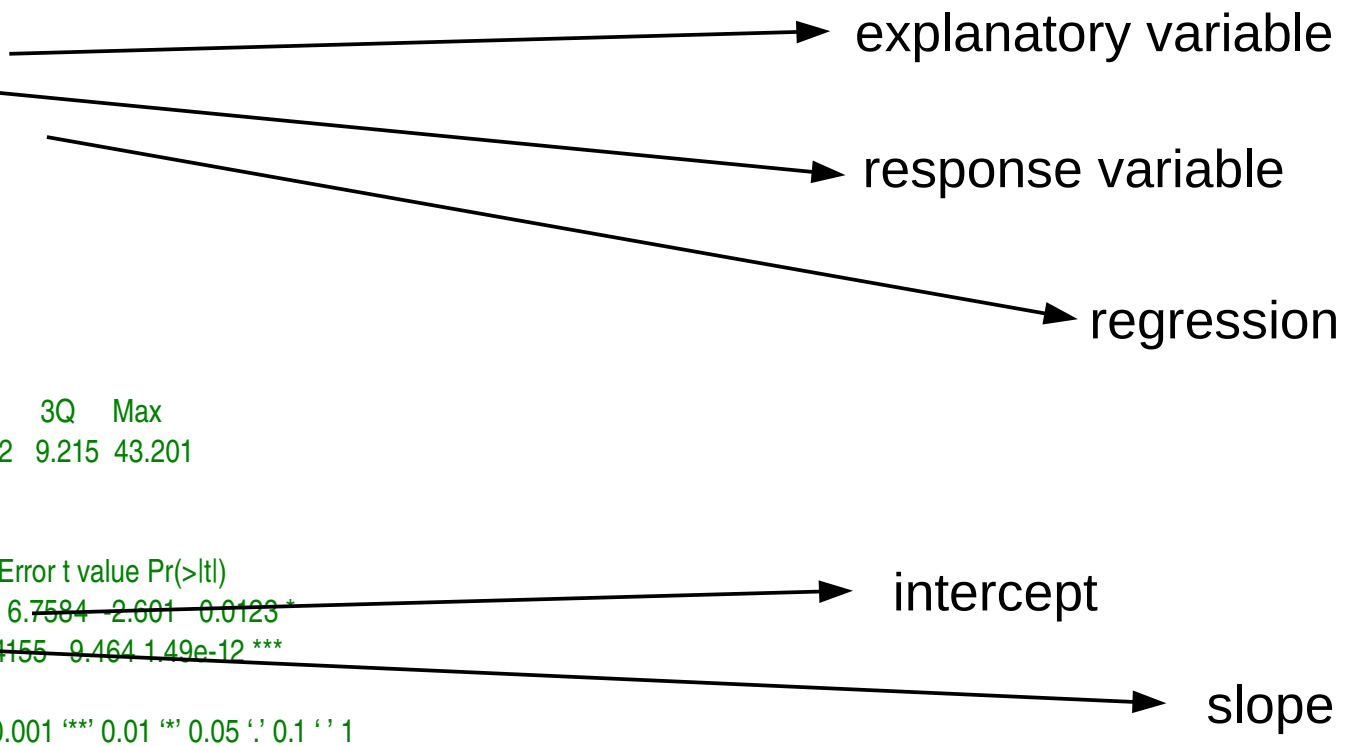
Now, just by looking at this plot it would appear that there's a correlation between speed and breaking distance. How do we handle this statistically?



R for statistical modeling

The answer is **regression**. In R this is carried out in a very easy way. There exist a sort of *grammar* for the regression and it is very easy to learn the basics of this grammar.

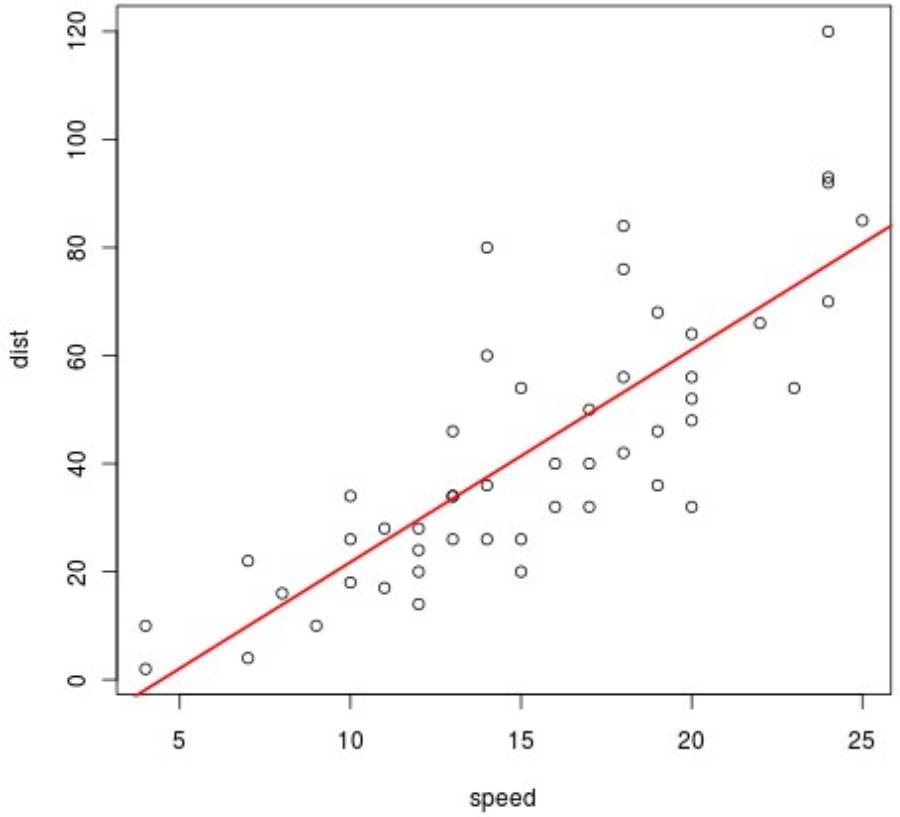
```
> x <- cars$speed
> y <- cars$dist
> model <- lm(y ~ x)
> summary(model)

Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max
-29.069  -9.525  -2.272   9.215  43.201

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -17.5791     6.7584  -2.601   0.0123 *
x             3.9324     0.4155   9.464 1.49e-12 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 15.38 on 48 degrees of freedom
Multiple R-squared: 0.6511,     Adjusted R-squared: 0.6438
F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.490e-12
```

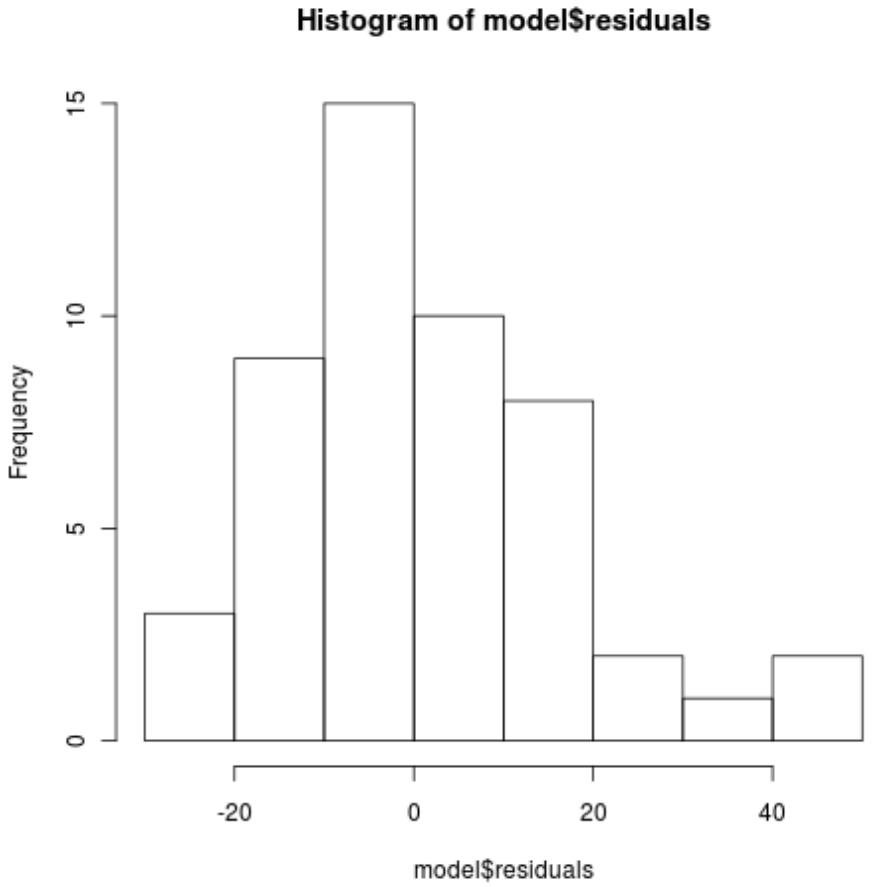


R for statistical modeling

Plotting stuff related to regression couldn't be easier:

```
> plot(cars)
> abline(model,lwd=2,col="red")
> hist(model$residuals)
```





R for statistical modeling

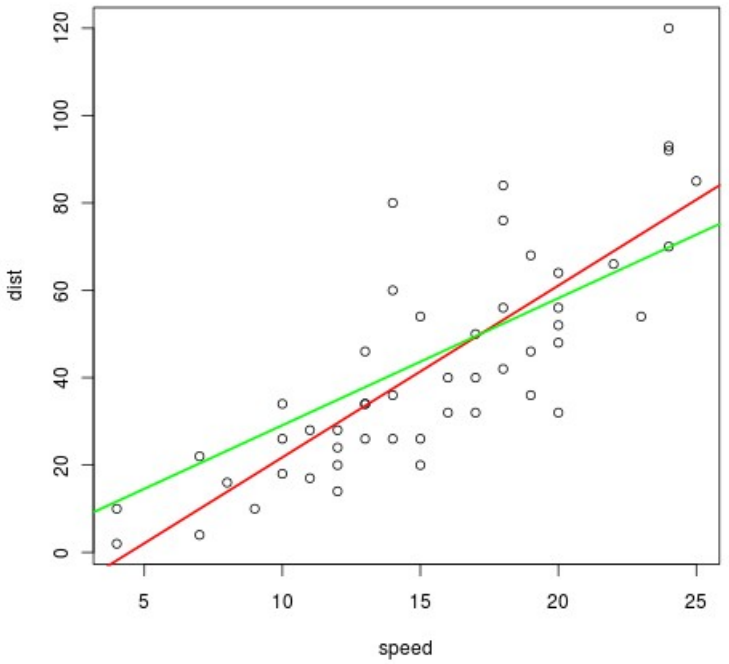
What if we don't want the intercept?

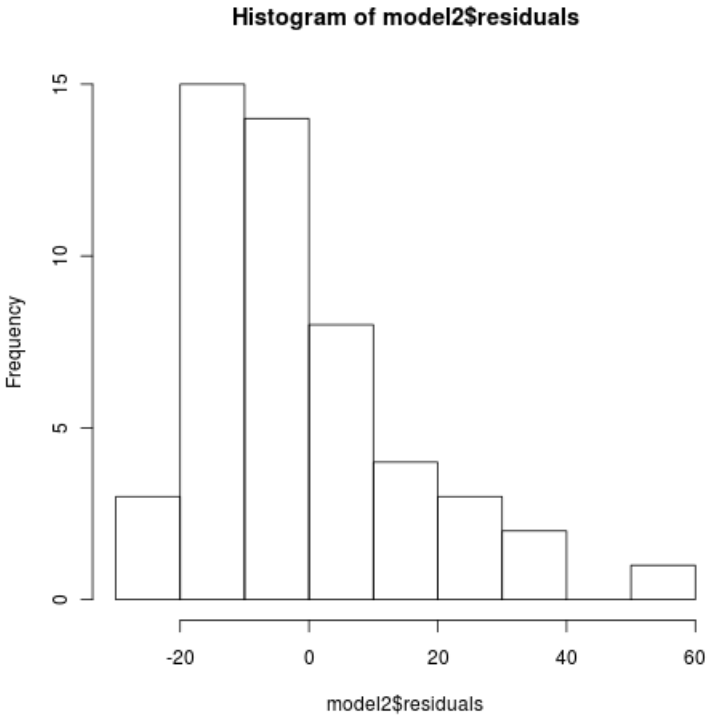
```
 > model2 <- lm(y ~ x-1)
> summary(model2)

Call:
lm(formula = y ~ x - 1)

Residuals:
    Min      1Q  Median      3Q     Max
-26.183 -12.637  -5.455   4.590  50.181

Coefficients:
  Estimate Std. Error t value Pr(>|t|)
x   2.9091     0.1414   20.58   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 16.26 on 49 degrees of freedom
Multiple R-squared: 0.8963,        Adjusted R-squared: 0.8942
F-statistic: 423.5 on 1 and 49 DF,  p-value: < 2.2e-16
```

R for statistical modeling

Which one is better?

```
> plot(cars)
> abline(model,lwd=2,col="red")
> abline(model2,lwd=2,col="green")
> hist(model2$residuals)
```





R for statistical modeling

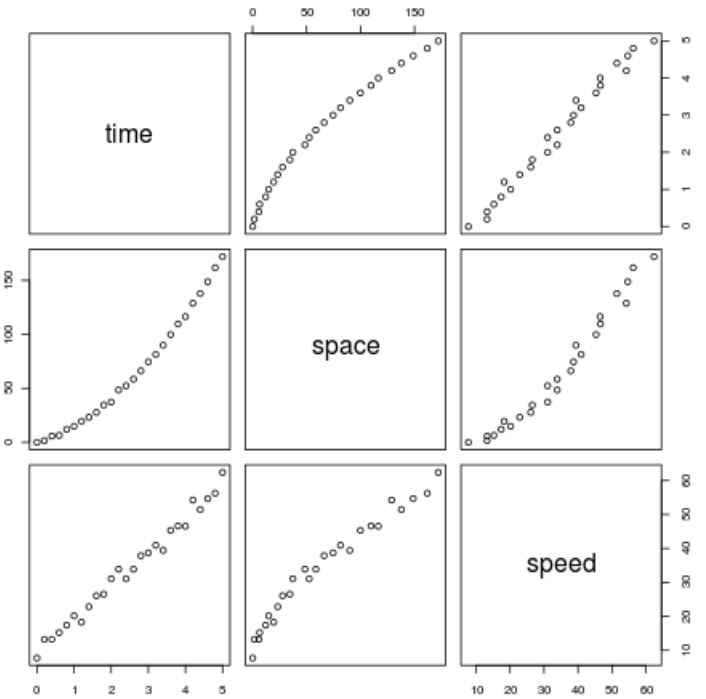
How do we import data from files? Consider the ascii file "regression.dat". It consists of 3 columns: time, space, speed. This file describes the motion of a freely falling body. Let us attempt two regressions; the first where the response variable is space; the second where the response variable is speed. In both cases the explanatory variable is time. The theory for the free-falling body is summarised in these equations:

$$x = x_0 + v_0 t + \frac{1}{2} g t^2 \qquad v = v_0 + g t$$

R for statistical modeling

First let's import data reading the ascii file with the function "read.table":

```
> gdata <- read.table("regression.dat",header=TRUE)
> class(gdata)
[1] "data.frame"
> str(gdata)
'data.frame':    26 obs. of  3 variables:
 $ time : num  0 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 ...
 $ space: num  0.07 1.65 5.79 6.55 12.1 ...
 $ speed: num  7.73 13.23 13.25 15.23 17.41 …
> plot(gdata)
```



There is always a default use for all R functions!

R for statistical modeling

Let's try and carry out statistical modeling for both Response variables using a linear dependence of both space and speed on time:

```
> t <- gdata$time
> x <- gdata$space
> v <- gdata$speed
> modelx1 <- lm(v ~ t)
> modelv <- lm(v ~ t)
> summary(modelv)

Call:
lm(formula = v ~ t)

Residuals:
    Min      1Q  Median      3Q     Max
-3.5973 -1.0383 -0.1530  0.9661  3.4543

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   9.1501     0.6951   13.16  1.8e-12 ***
t             9.9571     0.2384   41.76  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.824 on 24 degrees of freedom
Multiple R-squared: 0.9864,        Adjusted R-squared: 0.9859
F-statistic:  1744 on 1 and 24 DF,  p-value: < 2.2e-16
```

R for statistical modeling

```
> modelx <- lm(x ~ t)
> summary(modelx)

Call:
lm(formula = x ~ t)

Residuals:
    Min      1Q  Median      3Q     Max
-12.060  -7.851  -1.993   5.736  19.557

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -19.487      3.906  -4.989 4.27e-05 ***
t             34.418      1.340  25.690  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.25 on 24 degrees of freedom
Multiple R-squared: 0.9649,    Adjusted R-squared: 0.9634
F-statistic:   660 on 1 and 24 DF,  p-value: < 2.2e-16
```

R for statistical modeling

Both regressions look good. But we know space dependency on time should be quadratic... Let's try another model.

The quadratic dependency can be still modeled as a linear regression with two terms; the first one is t and this takes into account the intercept and linear coefficient of the model; the second term is t^2 and this takes into account the quadratic coefficient. In R this will have to be *protected* by the I() form: I(t^2). Thus, our model is:

$$x \sim t + I(t^2)$$

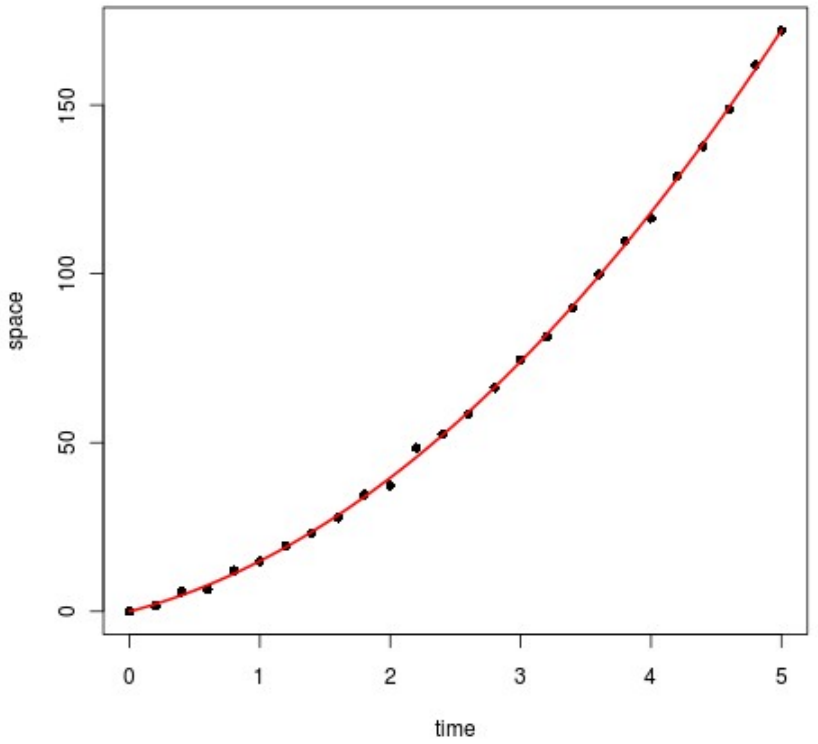
R for statistical modeling

```
> modelx <- lm(x ~ t + I(t^2))
> summary(modelx)

Call:
lm(formula = x ~ t + I(t^2))

Residuals:
     Min       1Q   Median       3Q      Max
-2.30725 -0.56688 -0.08133  0.59853  2.78933

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.01884    0.59295   0.032    0.975
t           10.03560    0.54918  18.274 3.43e-15 ***
I(t^2)       4.87655    0.10610  45.962  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.086 on 23 degrees of freedom
Multiple R-squared: 0.9996,         Adjusted R-squared: 0.9996
F-statistic: 3.042e+04 on 2 and 23 DF,  p-value: < 2.2e-16
```

BETTER!

R for statistical modeling

```
> plot(t,x,pch=16,xlab="time",ylab="space")
> tgrid <- seq(0,5,length=100)
> xpred <- predict(modelx,list(t=tgrid))
> xpred <- predict(modelx,list(t=tgrid))
> plot(t,x,pch=16,xlab="time",ylab="space")
> points(tgrid,xpred,type="l",col="red",lwd=2)
```



Try it yourself!

Exercise 1

Load dataset called “Formaldehyde”. Next, import data from file “formaldehyde.dat”. Be careful to assign a different name to data frame for the second set of data. Plot both sets of data in a same window, using two different colours and/or symbols. Highlight in the plot the observations where optden of the second dataset is greater than optden of the first.

Exercise 2

Generate simulated data for the distance of a free falling body. Carry out a regression and show/explain the regression fit.