# R: Statistical Functions

140.776 Statistical Computing

September 8, 2011

# Formula

Formula is an object in R. It is used by a lot of functions including lm, glm, boxplot, . . .

Example: a formula

```
z~x+y
```

in lm() fits a linear regression

```
z = a + b*x + c*y + err
```

- $\sim$: this operator is basic in the formation of such models.
- z $\sim$ *model*: response z is modelled by a linear predictor specified symbolically by *model*.
- $+$: terms in the model are separated by $+$ operators.

# Formula

You can create a formula object using formula() or as.formula():

```
> fo1<-formula(z~x+y)
> class(fo1)
[1] "formula"

> fo1<-"z~x+y"
> fo1
[1] "z~x+y"
> class(fo1)
[1] "character"
```

# Formula

```
> fo1<-formula("z~x+y")
> fo1
z ~ x + y
> class(fo1)
[1] "formula"
```

# Formula

It seems that R knows what a formula should look like.

```
> fo2<-z~x+y
> fo2
z ~ x + y
> class(fo2)
[1] "formula"
```

# Exercise

1. Load data from lm-manyx.txt.
2. What is the data structure?
3. Fit a regression $y = a_0 + a_1 x_1 + \ldots + a_N x_N + \epsilon$.
4. Is the intercept $a_0$ different from zero?

# Formula

To create a formula with many variables.

```
> xname <- paste("x", 1:5, sep="")
> fmla <- as.formula( paste( "y ~ ",
+ paste(xname, collapse= "+") ) )
> xname
[1] "x1" "x2" "x3" "x4" "x5"
> fmla
y ~ x1 + x2 + x3 + x4 + x5
```

# Formula

```
data<-read.table("lm-manyx.txt", sep="\t", header=TRUE)
xname<-paste("x", 1:100, sep="")
fmla <- as.formula( paste( "y ~ ", paste(xname, collapse= "+") ) )
fit<-lm(fmla, data=data)
summary(fit)
```

# Formula

Each term on the right hand side of a formula can be variable and factor names separated by : operators.

For example:

```
z~x+y+x:y
```

Here x:y means interactions between x and y. In other words,

```
lm(z~x+y+x:y)
```

fits a linear regression

```
z = a + b*x + c*y + d*x*y + err
```

# Formula

The * operator denotes factor crossing:

```
z~x*y
```

is equivalent to

```
z~x+y+x:y
```

# Formula

How about

```
v~(x+y+z)^2 ?
```

$v = a_0 + a_1 x + a_2 y + a_3 z + ?$

# Formula

The caret operator $\wedge$ indicates crossing to the specified degree:

```
v~(x+y+z)^2
```

is identical to

```
v~(x+y+z)*(x+y+z)
```

which in turn is identical to

```
v~x+y+z+x:y+x:z+y:z
```

# Formula

For example:

```
> x<-rnorm(100)
> y<-rnorm(100,1,2)
> z<-rnorm(100,2,1)
> v<-x+2y+3z+x*y+5x*z+rnorm(100)
> summary(lm(v~(x+y+z)^2))
Call:
lm(formula = v ~ (x + y + z)^2)
...
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.01134    0.30320  -0.037 0.970254
x            1.06714    0.30313   3.520 0.000669 ***
y            2.25134    0.11617  19.379  < 2e-16 ***
z            3.02615    0.14695  20.594  < 2e-16 ***
x:y          0.92384    0.05442  16.977  < 2e-16 ***
x:z          5.01335    0.14369  34.890  < 2e-16 ***
y:z         -0.10703    0.05328  -2.009 0.047448 *
```

# Formula

You get the same results by:

```
> summary(lm(v~(x+y+z)*(x+y+z)))
Call:
lm(formula = v ~ (x + y + z) * (x + y + z))
...
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.01134    0.30320  -0.037 0.970254
x            1.06714    0.30313   3.520 0.000669 ***
y            2.25134    0.11617  19.379  < 2e-16 ***
z            3.02615    0.14695  20.594  < 2e-16 ***
x:y          0.92384    0.05442  16.977  < 2e-16 ***
x:z          5.01335    0.14369  34.890  < 2e-16 ***
y:z         -0.10703    0.05328  -2.009 0.047448 *
```

# Formula

Sometimes you see

```
lm(y ~ x - 1)
```

# Formula

You can use - to remove terms. For example:

```
lm(y ~ x - 1)
```

fits a regression without intercept

```
y = a*x + err
```

# Formula

Another example:

```
v ~ (x + y + z)^2 - y:z
```

# Formula

```
v ~ (x + y + z)^2 - y:z
```

is identical to

```
v ~ x + y + z + x:y + x:z
```

# Formula

```
> summary(lm(v~(x+y+z)^2-1))
Call:
lm(formula = v ~ (x + y + z)^2 - 1)
...
Coefficients:
    Estimate Std. Error t value Pr(>|t|)
x    1.06908    0.29705   3.599 0.000512 ***
y    2.24893    0.09616  23.389  < 2e-16 ***
z    3.02107    0.05575  54.186  < 2e-16 ***
x:y  0.92396    0.05403  17.100  < 2e-16 ***
x:z  5.01208    0.13886  36.095  < 2e-16 ***
y:z -0.10595    0.04459  -2.376 0.019535 *
```

# Formula

```
> summary(lm(v~(x+y+z)^2-y:z))
Call:
lm(formula = v ~ (x + y + z)^2 - y:z)
...
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.31772    0.25923   1.226  0.22340
x            1.01309    0.30677   3.302  0.00136 **
y            2.04003    0.05011  40.708  < 2e-16 ***
z            2.85819    0.12278  23.280  < 2e-16 ***
x:y          0.92174    0.05528  16.675  < 2e-16 ***
x:z          5.02683    0.14583  34.470  < 2e-16 ***
```

# Formula

In addition to variable and factor names, formula can involve arithmetic expressions.

For example:

```
lm(log(v) ~ x+y+exp(z))
```

is legal.

# Formula

```
> x<-rgamma(100,1,2)
> y<-rnorm(100)
> z<-2*log(x)+y+rnorm(100)

> summary(lm(z~log(x)+y))
Call:
lm(formula = z ~ log(x) + y)
...
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.08265    0.13810  -0.598    0.551
log(x)       1.98617    0.06122  32.442   <2e-16 ***
y            0.84451    0.08329  10.140   <2e-16 ***
```

# Formula

```
lm(z~log(x)+y^2)
```

$$z = a_0 + a_1 * ? + a_2 * ?$$

# Formula

Let us try

```
lm(z~log(x)+y^2)
```

```
> z<-2*log(x)+y^2+rnorm(100)

> summary(lm(z~log(x)+y^2))
Call:
lm(formula = z ~ log(x) + y^2)
...
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.3068     0.2905   4.498 1.90e-05 ***
log(x)        2.0525     0.1288  15.936  < 2e-16 ***
y            -0.7513     0.1752  -4.288 4.26e-05 ***
```

# Formula

```
lm(z~log(x)+y^2)
```

does not give you $z = a_0 + a_1 * log(x) + a_2 * y^2$!

It gives you $z = a_0 + a_1 * log(x) + a_2 * y$.

# Formula

I() allows you to interpret arithmetic expressions as is.

```
> z<-2*log(x)+y^2+rnorm(100)

> summary(lm(z~log(x)+I(y^2)))
Call:
lm(formula = z ~ log(x) + I(y^2))
...
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.02738    0.17304  -0.158    0.875
log(x)       1.93257    0.06755  28.608   <2e-16 ***
I(y^2)       0.97666    0.05415  18.035   <2e-16 ***
```

# Formula

```
lm(v ~ x+I(y+z))
```

vs.

```
lm(v ~ x+y+z)
```

# Formula

```
lm(v ~ x+I(y+z))
```

fits the following regression $v = a + bx + c(y + z) + \epsilon$.

This is different from

```
lm(v ~ x+y+z)
```

which fits $v = a + bx + cy + dz + \epsilon$.

# Formula

If you use

```
lm(v ~ x+y+z)
```

```
> x<-rnorm(100)
> y<-rnorm(100)
> z<-rnorm(100)
> v<-1+2*x+3*y-4*z+rnorm(100)

> summary(lm(v~x+y+z))
Call:
lm(formula = v ~ x + y + z)
...
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.99238    0.09424   10.53   <2e-16 ***
x            1.91782    0.08642   22.19   <2e-16 ***
y            2.94171    0.08862   33.20   <2e-16 ***
z           -4.05110    0.11485  -35.27   <2e-16 ***
```

# Formula

On the other hand, if you use

```
lm(v ~ x+I(y+z))
```

```
> summary(lm(v~x+I(y+z)))
Call:
lm(formula = v ~ x + I(y + z))
...
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.9942     0.4759   4.190 6.15e-05 ***
x             2.4545     0.4432   5.538 2.61e-07 ***
I(y + z)      0.3976     0.3748   1.061    0.291
```

# Test nested models

anova() and anova.lmlist() allow you to test nested linear models.

For example:

```
> x<-rnorm(100)
> y<-rnorm(100,2,1)
> z<-x+2*y+rnorm(100)

> fit1<-lm(z~x+y)
> anova(fit1)
Analysis of Variance Table

Response: z
          Df Sum Sq Mean Sq F value    Pr(>F)
x          1 135.22  135.22  158.28 < 2.2e-16 ***
y          1 442.62  442.62  518.11 < 2.2e-16 ***
Residuals 97  82.87    0.85
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

# Test nested models

```
> fit2<-lm(z~y+x)
> anova(fit2)
Analysis of Variance Table

Response: z
          Df Sum Sq Mean Sq F value    Pr(>F)
y          1 477.34  477.34  558.76 < 2.2e-16 ***
x          1 100.50  100.50  117.64 < 2.2e-16 ***
Residuals 97  82.87    0.85
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

# Test nested models

```
> fit1<-lm(z~x+y)
> fit3<-lm(z~x+y+x:y)

> anova.lmlist(fit1,fit3)
Analysis of Variance Table

Model 1: z ~ x + y
Model 2: z ~ x + y + x:y
  Res.Df    RSS Df Sum of Sq      F  Pr(>F)
1     97 76.316
2     96 72.530  1     3.786 5.0112 0.02749 *
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```