Regression in R

Georges Monette

February 2019

Contents

1 Understanding Regression

To really understand regression, you need to be able to approach a problem from many different angles. I can think of at least 8 representations that complement each other. To master regression you need to know how to go from one representation to another and you need to know how to work within the right representation to think about your problem and to solve it.

Here are eight ways of thinking about regression. Some are very powerful for developing the mathematical theory of regression, other are best suited to visualize the interpretation of coefficients for a particular application.

1.1 1. Statistical: the matrix formulation of a model

$$Y = X\beta + \epsilon,\ \epsilon \sim N(0, \sigma^2)$$

and all the theory that follows, e.g.

$$Var(\hat{\beta}) = \sigma^2 \left(X'X\right)^{-1}$$

$$\hat{Y} = X(X'X)^{-1}X'Y = P_X Y$$

where P_X is the matrix of the orthogonal projection of $\mathbb{R}^n$ onto span(X).

1.2 2. Mathematical: the formula for the model

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 z_i + \beta_4 x_i z_i + \epsilon_i$$

$$\frac{\partial E(y)}{\partial x} = \beta_1 + 2\,\beta_2 x + \beta_4 z$$

1.3 3. Computing: commands and algorithms that fit the model

```
library(car)
fit <- lm(income ~  education * type, data = Prestige)
summary(fit)
```

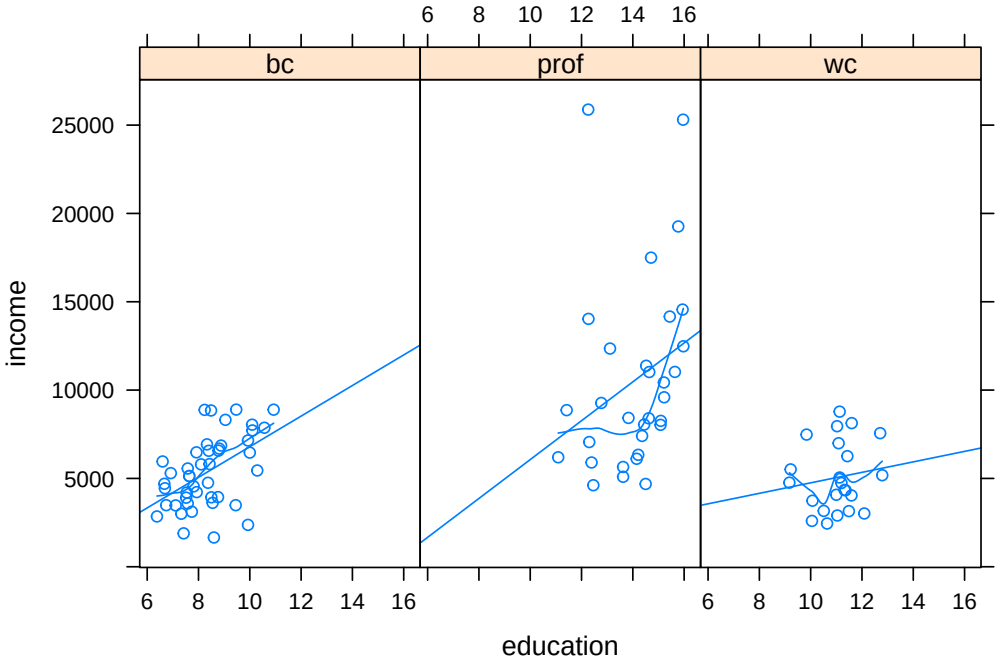
```
|
|   Call:
|   lm(formula = income ~ education * type, data = Prestige)
```

```
|
|   Residuals:
|       Min      1Q  Median      3Q     Max
|   -6330.8 -1769.2  -356.8  1166.5 17326.2
|
|   Coefficients:
|                     Estimate Std. Error t value Pr(>|t|)
|   (Intercept)        -1865.0     3682.3  -0.506   0.6137
|   education            866.0      436.4   1.984   0.0502 .
|   typeprof           -3068.4     7191.8  -0.427   0.6706
|   typewc              3646.5     9274.0   0.393   0.6951
|   education:typeprof   234.0      617.3   0.379   0.7055
|   education:typewc    -569.2      884.8  -0.643   0.5216
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
|
|   Residual standard error: 3333 on 92 degrees of freedom
|     (4 observations deleted due to missingness)
|   Multiple R-squared:  0.4105,    Adjusted R-squared:  0.3785
```

```
|   F-statistic: 12.81 on 5 and 92 DF,  p-value: 1.856e-09
```

1.4 4. Graphical: data space

```r
library(car)
library(lattice)
xyplot(income ~ education | type, Prestige,
       type = c('p','r','smooth'))
```


bc
prof
wc
income
education
25000
20000
15000
10000
5000
6
8
10
12
14
16

1.5 5. Graphical: beta space

```r
library(spida2)
library(latticeExtra)
fit <- lm(income ~  education + women, data = Prestige)
summary(fit)
```

```
|
|   Call:
|   lm(formula = income ~ education + women, data = Prestige)
|
|   Residuals:
|       Min      1Q  Median      3Q     Max
|   -7257.6 -1160.1  -238.6   681.1 16044.3
|
|   Coefficients:
|               Estimate Std. Error t value Pr(>|t|)
|   (Intercept) -1491.998   1162.299  -1.284    0.202
|   education     944.881    103.731   9.109 9.60e-15 ***
```

```
|   women          -64.056      8.921  -7.180 1.31e-10 ***
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
|
|   Residual standard error: 2839 on 99 degrees of freedom
|   Multiple R-squared:  0.5618,    Adjusted R-squared:  0.5529
|   F-statistic: 63.46 on 2 and 99 DF,  p-value: < 2.2e-16
```

```
plot(rbind(cell(fit),0),type= 'n',
  xlab = expression(beta[education]),
  ylab = expression(beta[women]))
lines(cell(fit,dfn=2), type = 'l', col = 'blue')
lines(cell(fit,dfn=1), type = 'l', col = 'red')
abline(h=0)
abline(v=0)
points(c(0,0), pch = 18)
```

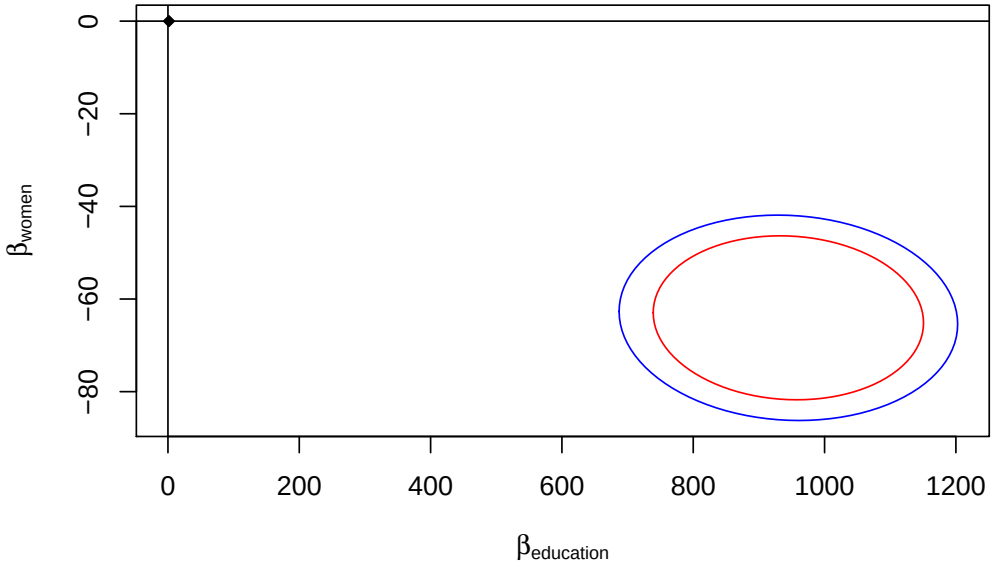

0
−20
−40
−60
−80
βwomen
0
200
400
600
800
1000
1200
βeducation

Figure: Confidence ellipse for two parameters jointly. The blue ellipse has 95% coverage in 2 dimensions and its perpendicular shadows onto the vertical and horizontal axes form Scheffe 95% confidence intervals for testing in a space of dimension 2. The similar shadows of the red ellipse provide ordinary 95% confidence intervals.

1.6 6. Graphical (different definition!): Path diagram of variables

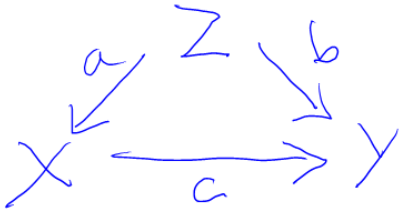
1.7 7. Geometric: Hilbert space representation of variables or 'variable space'

1.8 8. Most important: Real world interpretation

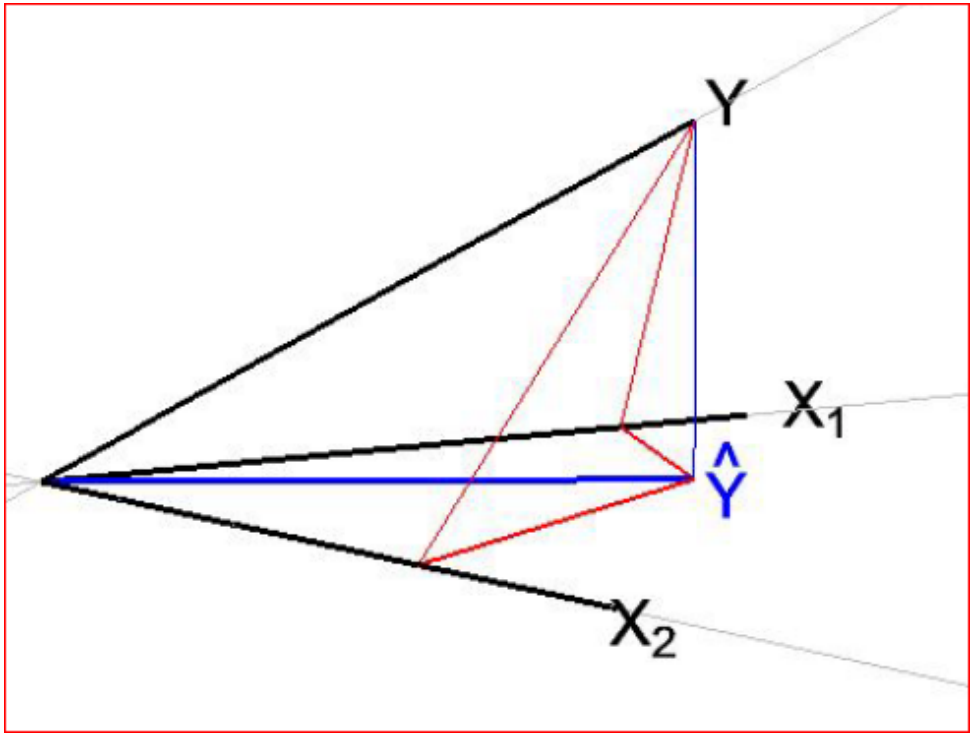
The most important representation is the interpretation of the model in the real world. Real world factors, such as the design, the nature of random assignment, the nature of random selection are fundamental in determining the interpretation of the model and on the strategy for model development, selection and interpretation.

Mediation

"Total effect"



Confounding


Y
X1
Ŷ
X2

This is where you determine the nature of the data: **observational or experimental**, and the nature of the questions: **predictive or causal** or descriptive.

2 Review of the matrix formulation and the general linear hypothesis (GLH)

$$Y = X\beta + \varepsilon$$

where

1. Y is a vector of length n representing n observations on a 'response' or 'dependent' variable,
2. X is a $n \times p$ matrix representing n observations on each of p 'predictor' or 'independent' variables. The first column frequently consists of 1's.
3. β is a vector of p parameters whose values are unknown and some aspect of which we wish to estimate. If the first column of X consists of 1's it is

customary to number the elements of β starting from 0: $\beta = (\beta_0, \beta_1, ..., \beta_{p-1})'$.

4. ε is a vector of length n representing 'errors' or 'residuals' that are not directly observed.

If X is of full column rank (i.e. $\text{rank}(X) = p$) and if we assume that $\varepsilon \sim N_n(0, \sigma^2 I)$ where I is the $n \times n$ identity matrix, and if $rank(X) = p$, then the **UMVUE** (Uniformly minimum variance unbiased estimator) of β is

$$\hat{\beta} = (X'X)^{-1}X'Y$$

with $E(\hat{\beta}) = \beta$ and $Var(\hat{\beta}) = \sigma^2(X'X)^{-1}$

2.1 Linear hypotheses

We can estimate or test hypotheses concerning one or more linear combinations of the βs by forming a $h \times p$ hypothesis matrix L and estimating the function of parameters:

$$\eta = L\beta$$

2.1.1 Exercise:

1. Why would we want to estimate a number of linear hypotheses simultaneously? Are the individual estimates of parameters different is we estimate them simultaneously? What difference does it make?

2.1.2 Example:

For a model with three parameters: $\beta = (\beta_0, \beta_1, \beta_2)'$ we can simultaneously estimate the sum and difference of β_1 and β_2 as follows.

Letting

$$L = \begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & -1 \end{bmatrix}$$

we get

$$\eta = \begin{bmatrix} \eta_1 \\ \eta_2 \end{bmatrix} = L\beta$$
$$= \begin{bmatrix} 0 & 1 & 1 \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}$$
$$= \begin{bmatrix} \beta_1 + \beta_2 \\ \beta_1 - \beta_2 \end{bmatrix}$$

2.2 Estimation and tests

Letting

$$\hat{\eta} = L\hat{\beta}$$

we have

$$E(\hat{\eta}) = E(L\hat{\beta}) = LE(\hat{\beta}) = L\beta$$

and

$$Var(\hat{\eta}) = \sigma^2 L(X'X)^{-1}L'$$

If L is of full row rank h and X is of full column rank we can test the hypothesis

$$H_0 : \eta = L\beta = 0$$

against the alternative that $\eta \neq 0$ (i.e. that $\eta_i \neq 0$ for at least one i) by using the null distribution:

$$\hat{\eta}' \left(\widehat{Var}(\hat{\eta})\right)^{-1} \hat{\eta} = \frac{\hat{\beta}' L' \left(L(X'X)^{-1}L'\right)^{-1} L\hat{\beta}}{s_e^2}$$
$$\sim h \times F_{h,\nu}$$

where s_e is the 'residual standard error':

$$s_e^2 = \frac{||Y - X\hat{\beta}||^2}{\nu}$$

with $\nu = n - p$ the degrees of freedom for the estimate s_e^2 of σ^2 and $F_{h,\nu}$ is the F distribution with h and ν degrees of freedom.

2.2.1 Notes

1. If the rows of L are not linearly independent then it isn't possible to invert $L(X'X)^{-1}L'$ but an equivalent hypothesis can be formed by

replacing L with a matrix whose rows form a basis of the row space of L.

2.2.2 Exercises:

1. Two L matrices with the same row space test equivalent simultaneous hypotheses (**Could you prove this?**).
2. For example, the hypothesis above is equivalent to the hypothesis that $\beta_1 = \beta_2 = 0$. **Why?**

3 Interpreting Regression Coefficients: Smoking and Life Expectancy

With complex models:

1. most regression coefficients in the standard output are of little interest and
2. most interesting questions are not answered with the standard regression coefficients.

Why do we pay attention to regression output? Because it may make some

sense for very simple additive models – but even then it is fraught with subtle traps most analysts do not understand.

We will illustrate these with the Smoking and Life Expectancy example using country-level data in 2004.

```
dall <- read.csv(paste0("http://",server,"/data/Smoking3.csv"))
dd <- subset( dall, sex == 'BTSX') # subset of a data frame
dd$LifeExp <- dd$lifeexp.Birth # Life expectancy at birth
dd$LE <- dd$LifeExp
dd$smoke <- dd$consumption.cigPC # cigarette consumption
                                 # per adult per year
dd$HE <- dd$HealthExpPC.Tot.ppp  # health expenditures per capita
                                 # in US$ PPP
dd$hiv <- dd$hiv_prev15_49  # prevalence of HIV in
                            # population 15 to 49
dd$special <- ifelse(
        dd$country %in% c('Angola','Sierra Leone',
                          'Equatorial Guinea'),
        1,
```

```
        0)  # indicator variable for 3 outlying countries
head(dd)
```

```
|                    country iso3 region HealthExpPC.Govt.exch Health
|   1            Afghanistan  AFG    EMR                 8.72
|   5                 Angola  AGO    AFR               114.61
|   7                Albania  ALB    EUR               114.21
|   10               Andorra  AND    EUR              2246.75
|   15 United Arab Emirates  ARE    EMR              1219.89
|   17             Argentina  ARG    AMR               540.82
|      HealthExpPC.Govt.ppp HealthExpPC.Tot.exch   total    govt pri
|   1                 7.87                55.93   50.47    7.87   4
|   5                132.04               186.26  214.58  132.04   8
|   7                253.49               254.64  565.20  253.49  31
|   10              2257.23              3058.98 3073.26 2257.23  81
|   15              1288.52              1639.87 1732.13 1288.52  44
|   17               869.44               891.80 1433.70  869.44  56
|      lifeexp.Birth lifeexp.At60 smoking.tobacco.current smoking.to
```

1	60	16	NA
5	51	16	NA
7	74	19	40
10	82	25	35
15	76	19	10
17	76	21	27

	smoking.cig.current	smoking.cig.daily	Pop.Total	Pop.MedAge	Po
1	NA	NA	29825	16.20	
5	NA	NA	20821	16.18	
7	40	36	3162	32.56	
10	35	31	78	NA	
15	8	5	9206	29.37	
17	26	20	41087	30.83	

	Pop.pCntOver60	Pop.pCntAnnGrowth	consumption.cigPC	hiv_prev15
1	3.82	-2.4	61	
5	3.84	-3.1	414	
7	14.93	-0.3	1116	
10	22.86	0.0	784	
15	0.81	-3.1	583	

```
|   17          14.97               -0.9              1042
|      smoke      HE hiv special
|   1     61   50.47 0.0       0
|   5    414  214.58 2.1       1
|   7   1116  565.20  NA       0
|   10   784 3073.26  NA       0
|   15   583 1732.13  NA       0
|   17  1042 1433.70 0.4       0
```

```r
fit.hiv2 <- lm( LifeExp ~
                  log(HE) * (smoke + I(smoke^2)) + hiv+special,
                dd,
                na.action = na.exclude)
summary(fit.hiv2)
```

```
|
|   Call:
|   lm(formula = LifeExp ~ log(HE) * (smoke + I(smoke^2)) + hiv +
|       special, data = dd, na.action = na.exclude)
```

```
|
|   Residuals:
|       Min      1Q  Median      3Q     Max
|   -9.0373 -2.3005  0.2043  2.0760  9.7344
|
|   Coefficients:
|                        Estimate Std. Error t value Pr(>|t|)
|   (Intercept)         3.283e+01  2.674e+00  12.280  < 2e-16 ***
|   log(HE)             6.091e+00  5.024e-01  12.124  < 2e-16 ***
|   smoke               3.642e-02  7.520e-03   4.844 3.31e-06 ***
|   I(smoke^2)         -1.518e-05  3.946e-06  -3.846 0.000181 ***
|   hiv                -7.351e-01  7.593e-02  -9.681  < 2e-16 ***
|   special            -1.822e+01  2.137e+00  -8.526 2.11e-14 ***
|   log(HE):smoke      -4.878e-03  1.155e-03  -4.223 4.30e-05 ***
|   log(HE):I(smoke^2)  2.007e-06  5.726e-07   3.504 0.000614 ***
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
|
|   Residual standard error: 3.63 on 141 degrees of freedom
```

```
|     (45 observations deleted due to missingness)
|   Multiple R-squared:  0.8613,    Adjusted R-squared:  0.8544
|   F-statistic:   125 on 7 and 141 DF,  p-value: < 2.2e-16
```

Do we need curvature in 'smoke'?

```
wald(fit.hiv2)
```

```
|    numDF denDF  F.value p.value
|       8   141 6968.992 <.00001
|                      Estimate   Std.Error DF  t-value   p-value Lo
|   (Intercept)         32.831342 2.673604  141 12.279805 <.00001  2
|   log(HE)              6.091328 0.502427  141 12.123808 <.00001
|   smoke                0.036424 0.007520  141  4.843564 <.00001
|   I(smoke^2)          -0.000015 0.000004  141 -3.846440 0.00018  -
|   hiv                 -0.735098 0.075932  141 -9.681048 <.00001  -
|   special            -18.223307 2.137312  141 -8.526276 <.00001 -2
|   log(HE):smoke       -0.004878 0.001155  141 -4.223209 0.00004  -
|   log(HE):I(smoke^2)   0.000002 0.000001  141  3.504458 0.00061
|                      Upper 0.95
```

```
|   (Intercept)          38.116875
|   log(HE)               7.084592
|   smoke                 0.051290
|   I(smoke^2)           -0.000007
|   hiv                  -0.584986
|   special             -13.997989
|   log(HE):smoke        -0.002595
|   log(HE):I(smoke^2)    0.000003
```

```
wald(fit.hiv2, '2') # using '2' as a regular expression
```

```
|     numDF denDF  F.value p.value
|   2     2   141 8.835643 0.00024
|                      Estimate Std.Error DF  t-value   p-value Lowe
|   I(smoke^2)         -1.5e-05 4e-06     141 -3.846440 0.00018 -2.3
|   log(HE):I(smoke^2)  2.0e-06 1e-06     141  3.504458 0.00061  1.0
|                      Upper 0.95
|   I(smoke^2)         -7e-06
|   log(HE):I(smoke^2)  3e-06
```

How about interaction?

```
wald(fit.hiv2, ':')
```

```
|     numDF denDF F.value p.value
|   :     2   141  9.2372 0.00017
|                       Estimate  Std.Error DF  t-value   p-value Low
|   log(HE):smoke       -0.004878 0.001155  141 -4.223209 0.00004 -0.
|   log(HE):I(smoke^2)   0.000002 0.000001  141  3.504458 0.00061  0.
|                       Upper 0.95
|   log(HE):smoke       -0.002595
|   log(HE):I(smoke^2)   0.000003
```

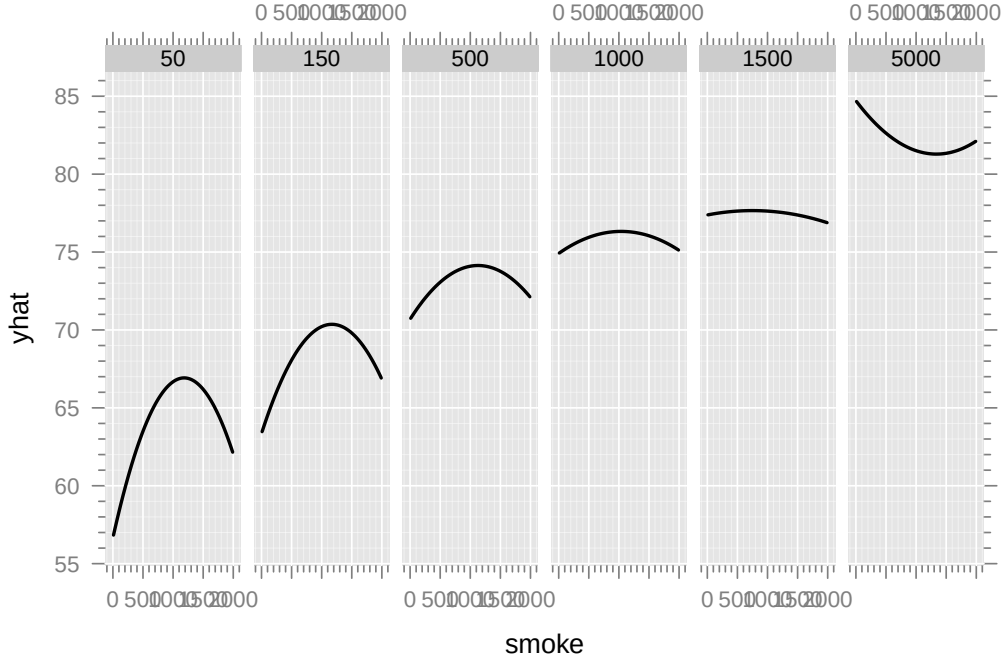
Create a prediction data frame over which to estimate fitted values. We will look at countries with low hiv and exclude outliers.

```
pred <- expand.grid(
  HE = c(50,150,500, 1000, 1500, 5000),
  smoke = seq(10,2000,20),
  hiv = 0,
```

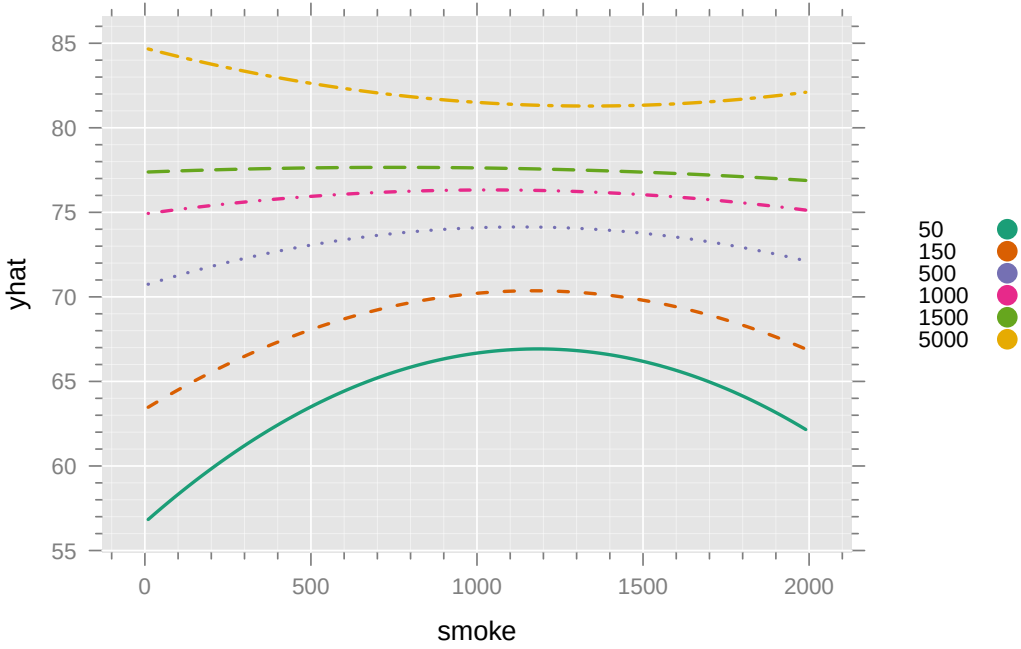
```
  special = 0)
```

Finding $\hat{Y}$ over a grid of values:

```
pred$yhat <- predict(fit.hiv2, newdata = pred)
gd(lwd = 2) # no groups
gd(lwd = c(2,2)) # groups
xyplot(yhat ~ smoke | factor(HE), pred, type = 'l',
       layout = c(6,1))
```

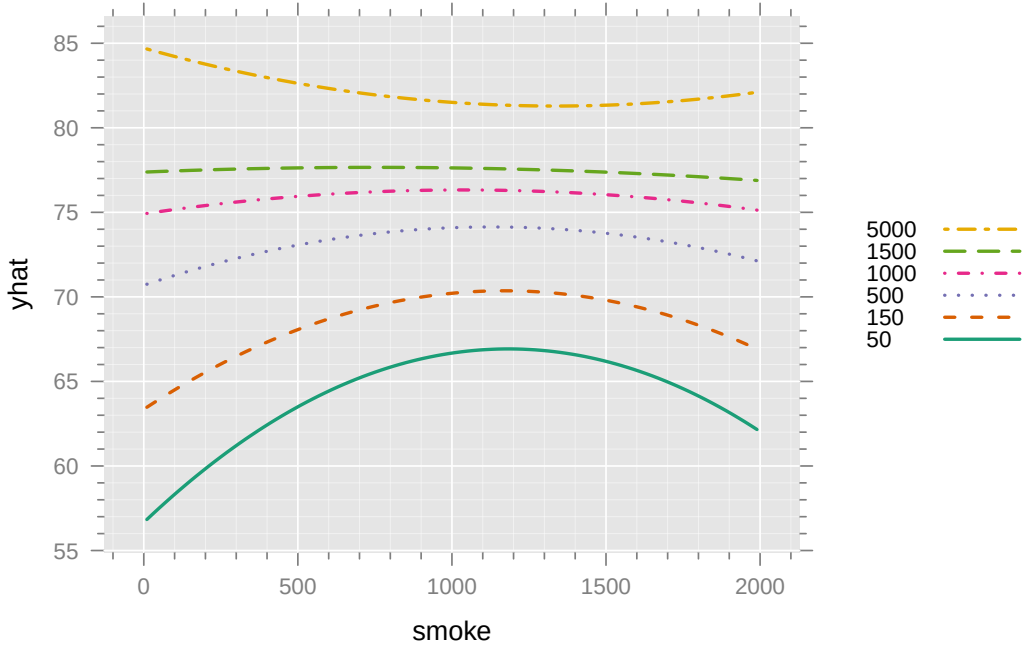

50
150
500
1000
1500
5000
yhat
85
80
75
70
65
60
55
0 500 1000 1500 2000
smoke

```
xyplot(yhat ~ smoke , groups = factor(HE), pred, type = 'l',
       auto.key=list(space='right'))
```


85
80
75
70
65
60
55
yhat
0
500
1000
1500
2000
smoke
50
150
500
1000
1500
5000

It's a good idea to make the order in the legend match the physical location in the graph, as much as feasible.

```
xyplot(yhat ~ smoke , groups = factor(HE), pred, type = 'l',
       auto.key=list(space='right', lines = T, points = F,
                     reverse.rows = TRUE))
```


yhat
85
80
75
70
65
60
55
0
500
1000
1500
2000
smoke
5000
1500
1000
500
150
50

Suppose we want to estimate the slope of these fitted curves: i.e. the 'effect'of an additional cigarette as a function of health expenditures and amount smoked.

We start with the mathematical formula for the model:

Letting $\eta = E(y|HE, Smoke, HIV, Special)$

$$\begin{aligned}\eta =& \beta_0 + \beta_1 \times ln(HE) \\ &+ \beta_2 \times Smoke \\ &+ \beta_3 \times Smoke^2 \\ &+ \beta_4 \times HIV \\ &+ \beta_5 \times Special \\ &+ \beta_6 \times ln(HE)\, Smoke \\ &+ \beta_7 \times ln(HE)\, Smoke^2\end{aligned}$$

To understand the interpretation of the coefficients β_i, we differentiate η with respect to each of the independent variables:

$$
\begin{aligned}
\frac{\partial \eta}{\partial HE} &= \beta_1 \frac{1}{HE} + \beta_6 \frac{Smoke}{HE} + \beta_7 \frac{Smoke^2}{HE} \\
\frac{\partial \eta}{\partial Smoke} &= \beta_2 + 2\beta_3 Smoke + \beta_6 ln(HE) + 2\beta_7 Smoke\, ln(HE) \\
\frac{\partial \eta}{\partial HIV} &= \beta_4 \\
\frac{\partial \eta}{\partial Special} &= \beta_5 \\
\frac{\partial^2 \eta}{\partial HE^2} &= \beta_1 \frac{-1}{HE^2} + \beta_6 \frac{-Smoke}{HE^2} + \beta_7 \frac{-Smoke^2}{HE^2} \\
\frac{\partial^2 \eta}{\partial Smoke^2} &= 2\beta_3 \\
\frac{\partial^2 \eta}{\partial HE\, \partial Smoke} &= \beta_6 \frac{1}{HE} + 2\beta_7 \frac{Smoke}{HE}
\end{aligned}
$$

Thus β_2 is the **partial derivative** of η with respect to $Smoke$ when $ln(HE) = Smoke = 0$.

When $ln(HE) = 5$ and $Smoke = 4$, the partial derivative of η with respect to

Smoke is

$$\frac{\partial \eta}{\partial Smoke} = \beta_2 + 8\beta_3 + 5\beta_6 + 40\beta_7$$

whose estimator is

$$\hat{\beta}_2 + 8\hat{\beta}_3 + 5\hat{\beta}_6 + 40\hat{\beta}_7$$

which we can express as a linear transformation of the $\hat{\beta}$ vector. Letting

$$L = \begin{bmatrix} 0 & 0 & 1 & 8 & 0 & 0 & 5 & 40 \end{bmatrix}$$

we have:

$$\hat{\phi} = L\hat{\beta} = \begin{bmatrix} 0 & 0 & 1 & 8 & 0 & 0 & 5 & 40 \end{bmatrix} \widehat{\begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \\ \beta_7 \end{bmatrix}}$$

If we wish to simultaneously estimate the 'effect' of Smoke and the 'effect' of HE given values of HE and Smoke, we can form the L matrix:

$$L = \begin{bmatrix} 0 & 0 & 1 & 2\,Smoke & 0 & 0 & ln(HE) & 2\,Smoke\,ln(HE) \\ 0 & 1 & 0 & 0 & 0 & 0 & \frac{Smoke}{HE} & \frac{Smoke^2}{HE} \end{bmatrix}$$

and

$$\hat{\phi} = L\hat{\beta}$$

In this case $\hat{\phi}$ is a column vector of length 2.

In both cases, inference about ϕ uses the fact that

$$Var(\hat{\phi}) = LVar(\hat{\beta})L'$$

and

$$Var(\hat{\beta}) = \sigma^2(X'X)^{-1}$$

With a normal linear model in which

$$Y = X\beta + \epsilon,\ \epsilon \sim N(0, \sigma^2\, I)$$

we have that

$$\left(\hat{\phi}-\phi\right)'\left(s^2L(X'X)^{-1}L'\right)^{-1}\left(\hat{\phi}-\phi\right)\sim h\times F_{h,\nu}$$

where h is the number of rows of L (assuming that L is of full row rank) and $\nu=n-p$ where n and p are the number of rows and columns of X respectively, again assuming that X is of full column rank.

We can compute these quantities in R from a fitted model. Note how the 'evalq' function evaluates an expression at the values given in the list provided as the 'envir' argument.

```
L <- evalq( rbind(
c( 0,0,1, 2*smoke, 0 ,0 , log(HE), 2*smoke*log(HE)),
c( 0,1,0, 0      , 0,       0, smoke / HE, smoke^2/HE)),
    envir = list( smoke = 4, HE = exp(5)))
L
```

```
|      [,1] [,2] [,3] [,4] [,5] [,6]       [,7]       [,8]
|  [1,]    0    0    1    8    0    0 5.00000000 40.0000000
|  [2,]    0    1    0    0    0    0 0.02695179  0.1078072
```

$\hat{\beta}$:

```
coef(fit.hiv2)
```

```
|          (Intercept)             log(HE)               smoke
|         3.283134e+01        6.091328e+00        3.642387e-02       -1
|                  hiv             special      log(HE):smoke log(HE)
|        -7.350980e-01       -1.822331e+01       -4.878054e-03        2
```

$\hat{\phi} = L\hat{\beta}$:

```
(phihat <- L %*% coef(fit.hiv2))
```

```
|               [,1]
|   [1,] 0.01199246
|   [2,] 6.09119673
```

$s^2(X'X)^{-1}$:

```
vcov(fit.hiv2)
```

```
|                          (Intercept)       log(HE)          smoke
```

```
|   (Intercept)         7.148161e+00 -1.298157e+00 -1.437171e-02  5.
|   log(HE)            -1.298157e+00  2.524328e-01  2.379512e-03 -8.
|   smoke              -1.437171e-02  2.379512e-03  5.655121e-05 -2.
|   I(smoke^2)          5.526193e-06 -8.798735e-07 -2.706223e-08  1.
|   hiv                -9.426054e-03 -2.319091e-03  1.603252e-05 -6.
|   special             4.335683e-01 -1.122056e-01 -1.123946e-03  4.
|   log(HE):smoke       2.438561e-03 -4.363636e-04 -8.441015e-06  3.
|   log(HE):I(smoke^2) -9.000438e-07  1.540858e-07  3.944264e-09 -2.
|                                 hiv       special log(HE):smoke log
|   (Intercept)        -9.426054e-03  4.335683e-01  2.438561e-03
|   log(HE)            -2.319091e-03 -1.122056e-01 -4.363636e-04
|   smoke               1.603252e-05 -1.123946e-03 -8.441015e-06
|   I(smoke^2)         -6.423962e-09  4.625775e-07  3.949943e-09
|   hiv                 5.765616e-03 -1.218220e-03  2.000199e-06
|   special            -1.218220e-03  4.568101e+00  2.166940e-04
|   log(HE):smoke       2.000199e-06  2.166940e-04  1.334160e-06
|   log(HE):I(smoke^2) -2.425728e-10 -8.002814e-08 -6.010943e-10
```

$\hat{Var}(\hat{\phi}) = L(s^2(X'X)^{-1})L'$:

```
(Vphihat <- L %*% vcov(fit.hiv2) %*% t(L))
```

```
|                 [,1]          [,2]
|    [1,] 5.453266e-06 0.0001967712
|    [2,] 1.967712e-04 0.2524093523
```

To test the hypothesis that $\phi = 0$, we have

$$F = \hat{\phi}' \left(\hat{Var}(\hat{\phi}) \right)^{-1} \hat{\phi}/h$$

```
(Ftest <- (t(phihat) %*% solve(Vphihat) %*% phihat)/2)
```

```
|             [,1]
|    [1,] 78.44759
```

```
1-pf(Ftest,2, fit.hiv2$df.residual)
```

```
|         [,1]
|    [1,]    0
```

```
pf(Ftest,2, fit.hiv2$df.residual, lower.tail = FALSE)
```

```
|                 [,1]
|   [1,] 1.254508e-23
```

Note how rounding error is reduced by using the 'lower.tail' parameter.

3.1 Functions to test linear hypotheses

The functions 'lht' in the 'car' package and 'wald' in the 'spida2' package can be used to test General Linear Hypotheses.

-

```
require(car)
lht(fit.hiv2,L)
```

```
|   Linear hypothesis test
|
|   Hypothesis:
```

```
|   smoke  + 8 I(smoke^2)  + 5 log(HE):smoke  + 40 log(HE):I(smoke^2
|   log(HE)  + 0.0269517879963419 log(HE):smoke  + 0.107807151985367
|
|   Model 1: restricted model
|   Model 2: LifeExp ~ log(HE) * (smoke + I(smoke^2)) + hiv + specia
|
|     Res.Df    RSS Df Sum of Sq      F    Pr(>F)
|   1    143 3925.4
|   2    141 1858.0  2    2067.4 78.448 < 2.2e-16 ***
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
wald(fit.hiv2, L)
```

```
|     numDF denDF  F.value p.value
|   1     2   141 78.44759 <.00001
|         Estimate Std.Error DF  t-value   p-value Lower 0.95 Upper 0
|   [1,] 0.011992 0.002335  141  5.135465 <.00001 0.007376   0.01660
|   [2,] 6.091197 0.502404  141 12.124111 <.00001 5.097979   7.08441
```

The 'lht' function can take a right-hand side to test hypotheses of the form $H_0 : \phi = \phi_0$. 'wald' can only test $H_0 : \phi = 0$. The 'wald' function can handle L matrices whose rows are not linearly independent. For example, in some uses of 'wald', the 'L' matrix is the whole design matrix.

The second argument of the 'wald' matrix can be a regular expression that is matched against the names of terms in the model. All terms matched by the regular expression are simultaneously tested. Thus one can test the 'overall' significance of an independent variable by testing whether all terms containing that variable are equal to 0. One can also use this approach to test higher-order interactions.

```
wald(fit.hiv2, "smoke")    # is there statistical evidence
```

```
|          numDF denDF  F.value p.value
|   smoke      4   141 8.024538   1e-05
|                       Estimate  Std.Error DF  t-value   p-value Low
|   smoke                0.036424 0.007520  141  4.843564 <.00001  0.
|   I(smoke^2)          -0.000015 0.000004  141 -3.846440 0.00018 -0.
|   log(HE):smoke       -0.004878 0.001155  141 -4.223209 0.00004 -0.
```

```
|   log(HE):I(smoke^2)  0.000002 0.000001  141  3.504458 0.00061  0.
|                      Upper 0.95
|   smoke               0.051290
|   I(smoke^2)         -0.000007
|   log(HE):smoke      -0.002595
|   log(HE):I(smoke^2)  0.000003
```

```
                          # that 'smoke' improves prediction?
wald(fit.hiv2, "HE")      # ditto for HE
```

```
|      numDF denDF  F.value p.value
|   HE     3   141 87.10337 <.00001
|                      Estimate  Std.Error DF  t-value   p-value Low
|   log(HE)             6.091328 0.502427  141 12.123808 <.00001  5.
|   log(HE):smoke      -0.004878 0.001155  141 -4.223209 0.00004 -0.
|   log(HE):I(smoke^2)  0.000002 0.000001  141  3.504458 0.00061  0.
|                      Upper 0.95
|   log(HE)             7.084592
|   log(HE):smoke      -0.002595
```

```
|   log(HE):I(smoke^2)  0.000003
```

```
wald(fit.hiv2, ":")       # ditto for interactions?
```

```
|     numDF denDF F.value p.value
|   :     2   141  9.2372 0.00017
|                       Estimate  Std.Error DF  t-value   p-value Low
|   log(HE):smoke       -0.004878 0.001155  141 -4.223209 0.00004 -0.
|   log(HE):I(smoke^2)   0.000002 0.000001  141  3.504458 0.00061  0.
|                       Upper 0.95
|   log(HE):smoke       -0.002595
|   log(HE):I(smoke^2)   0.000003
```

```
wald(fit.hiv2, "2)" )     # ditto for quadratic terms?
```

```
|      numDF denDF  F.value p.value
|   2)     2   141 8.835643 0.00024
|                       Estimate Std.Error DF  t-value   p-value Lowe
|   I(smoke^2)          -1.5e-05 4e-06     141 -3.846440 0.00018 -2.3
|   log(HE):I(smoke^2)   2.0e-06 1e-06     141  3.504458 0.00061  1.0
```

```
|                          Upper 0.95
|   I(smoke^2)             -7e-06
|   log(HE):I(smoke^2)      3e-06
```

3.2 Strategies to simplify models

There are many strategies for potentially simplifying large models. The resulting model will depend on the strategy.

One is to attack higher-order interactions and simplify the model by dropping groups of interactions that are not significant but, initially, leaving main effects and lower-order interactions. In many situations there are obvious moderator variables whose interactions should not be dropped as aggressively as those of other variables for which oversimplification to an additive model may be more innocuous. Remember the consequences of dropping an interaction. Main effects become weighted averages of conditional effects, weighted by inverse variance.

Another approach is to drop all terms for selected independent variables if they are not sufficiently significant in an overall test.

The two approaches can be combined depending on the role of variables and the goals of the analysis.

The choice of approach should be guided by many factors: which null hypotheses are likely to be reasonable, the interpretive value of having a simple additive model versus the added validity of estimating conditional effects that are not averaged over levels of variables that may be important, etc. There's a good discussion of these problems in (**???**).

3.3 Estimating effects over a grid

In a model with interactions and non-linear functions of some independent variables, it is often interesting to characterize how effects (partial derivatives) and inferences about effects vary over a range of predictors.

The 'effects' package (Fox and Hong (2009)) allows easy visualization of predicted values as a variables changes keeping other variables constant. For other variables that interact with the variable whose effect is visualized, the interacting variables are kept constant at a number of selected values. The package is very effective for the easy visualization of moderately complex

models with interactions.

The same can be achieved but much more laboriously with the 'Lfx' function in the 'spida2' package (Monette et al. (2018)). With the 'Lfx' function it is possible to estimate derivatives of all orders and features of general parametric splines with the 'gsp' function.

The 'Lfx' function generates an expression which can then be edited to generate large L matrices. The result of the wald test applied to this L matrix can be transformed into a data frame for plotting with error bands.

The following illustrates the use of the 'Lfx' function.

```
Lfx(fit.hiv2)
```

```
|   list( 1,
|   1 * M(log(HE)),
|   1 * smoke,
|   1 * M(I(smoke^2)),
|   1 * hiv,
|   1 * special,
```

```
|    1 * M(log(HE)) * smoke,
|    1 * M(log(HE)) * M(I(smoke^2))
|    )
```

The expression generated by 'Lfx' can be edited to generate desired effects. The result is then fed back to 'Lfx' along with a data frame on which to evaluate the edited expression. Note that the 'M' functions preserved the shape of multi-term blocks in the design matrix so that multiplying them by 0 is a way of generating a block of 0s of the right dimension. In the following, we edit the expression to estimate the effect of smoking by differentiating with respect to 'smoke':

```
pred <- expand.grid(
  HE = c(50,150,500, 1000, 1500, 5000),
  smoke = seq(10,2000,20),
  hiv = 0,
  special = 0)
head(pred)  # first 6 lines of 'pred'
```

```
|       HE smoke hiv special
|   1   50    10   0       0
```

```
|   2  150    10   0        0
|   3  500    10   0        0
|   4 1000    10   0        0
|   5 1500    10   0        0
|   6 5000    10   0        0
```

Predicted values as a function of HE and smoke

Use the list created above and edit it by differentiating each term with respect to 'smoke' to get the marginal 'effect' of an extra unit of 'smoke':

```
Lfx(fit.hiv2)
```

```
|   list( 1,
|   1 * M(log(HE)),
|   1 * smoke,
|   1 * M(I(smoke^2)),
|   1 * hiv,
|   1 * special,
|   1 * M(log(HE)) * smoke,
```

```
|   1 * M(log(HE)) * M(I(smoke^2))
|   )
```

Differentiated:

```
L <- Lfx(fit.hiv2,
     list( 0,
           0 * M(log(HE)),
           1 ,
           1 * M(I(2*smoke)),
           0 * hiv,
           0 * special,
           1 * M(log(HE)) * 1,
           1 * M(log(HE)) * M(I(2*smoke))
     ), pred)
dim(L)
```

```
|   [1] 600   8
```

```r
head(L)
```

	(Intercept)	log(HE)	smoke	I(smoke^2)	hiv	special	log(HE):smoke	log(HE):I(smoke^2)
1	0	0	1	20	0	0	3.912023	78.24046
2	0	0	1	20	0	0	5.010635	100.21271
3	0	0	1	20	0	0	6.214608	124.29216
4	0	0	1	20	0	0	6.907755	138.15511
5	0	0	1	20	0	0	7.313220	146.26441
6	0	0	1	20	0	0	8.517193	170.34386

```
ww <- wald(fit.hiv2, L)
```

```
|   Warning in wald(fit.hiv2, L): Poorly conditioned L matrix, calcu
|   be incorrect
```

```
ww <- as.data.frame(ww)
head(ww)
```

```
|              coef           se           U2           L2   HE smoke
|   1  0.0171942856 0.003272967 0.023740220  0.010648352   50    10
|   2  0.0118792893 0.002313918 0.016507124  0.007251454  150    10
|   3  0.0060545675 0.001764820 0.009584207  0.002524927  500    10
|   4  0.0027011781 0.001883655 0.006468487 -0.001066131 1000    10
|   5  0.0007395711 0.002094140 0.004927851 -0.003448709 1500    10
|   6 -0.0050851507 0.003067648 0.001050145 -0.011220447 5000    10
```

More informative labels:

```
ww$HEfac <- factor(ww$HE)
levels(ww$HEfac) <- paste("Health Exp/Cap:",levels(ww$HEfac))
```

Doing this preserved the order of factor levels

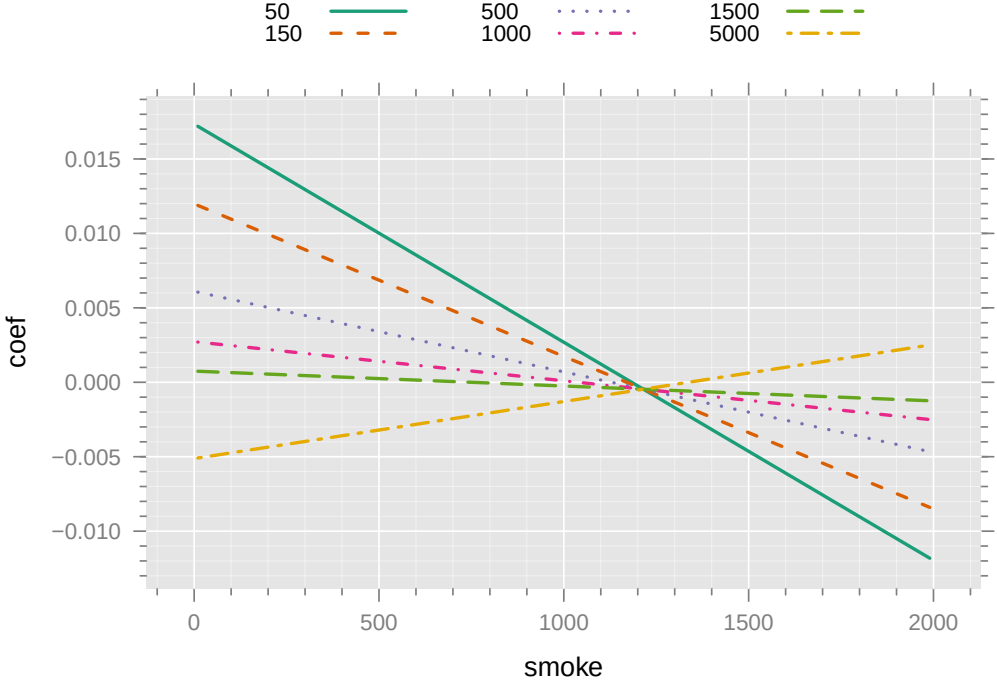
```
tab(ww, ~ HEfac + HE)
```

```
|                          HE
|   HEfac                   50 150 500 1000 1500 5000 Total
|     Health Exp/Cap: 50   100   0   0    0    0    0   100
|     Health Exp/Cap: 150    0 100   0    0    0    0   100
|     Health Exp/Cap: 500    0   0 100    0    0    0   100
|     Health Exp/Cap: 1000   0   0   0  100    0    0   100
|     Health Exp/Cap: 1500   0   0   0    0  100    0   100
|     Health Exp/Cap: 5000   0   0   0    0    0  100   100
|     Total                100 100 100  100  100  100   600
```

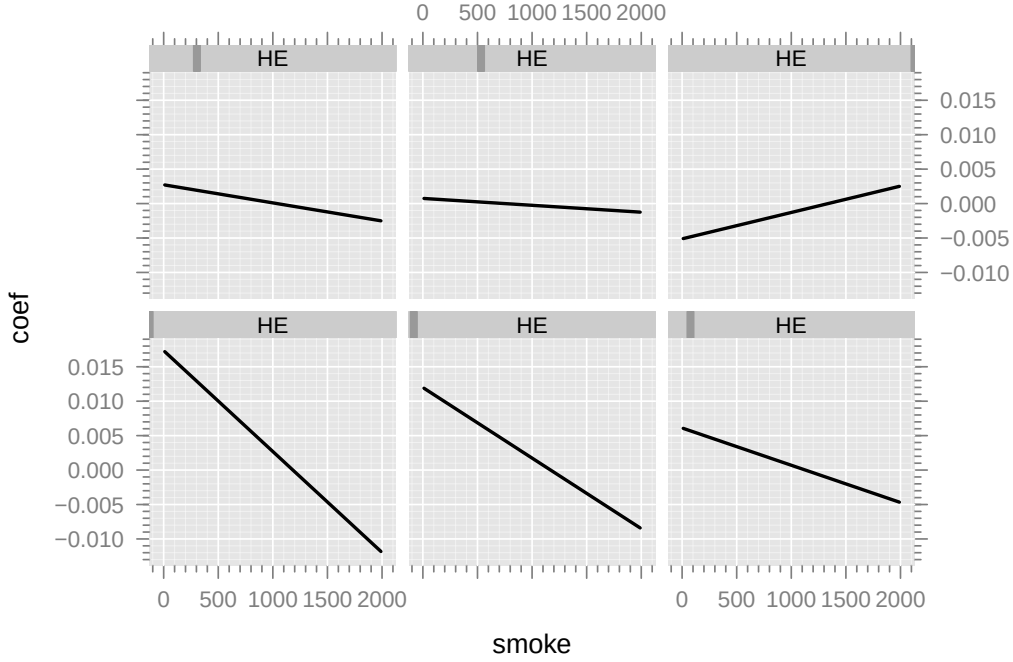
Levels are in numerical order when creating a factor from a numeric object! This was not always so! We are taking the derivative with respect to 'smoke' of these functions:

```
xyplot( coef ~ smoke, ww, groups = HE,
        auto.key = list(columns = 3, lines = T,
```

```
                    points = F),type = 'l')
```


50
150
500
1000
1500
5000
0.015
0.010
0.005
0.000
−0.005
−0.010
coef
0
500
1000
1500
2000
smoke

```
xyplot( coef ~ smoke | HE, ww, type = 'l')
```


HE
HE
HE
HE
HE
HE
0 500 1000 1500 2000
0.015
0.010
0.005
0.000
−0.005
−0.010
coef
smoke

```
xyplot( coef ~ smoke | HEfac, ww, type = 'l')
```

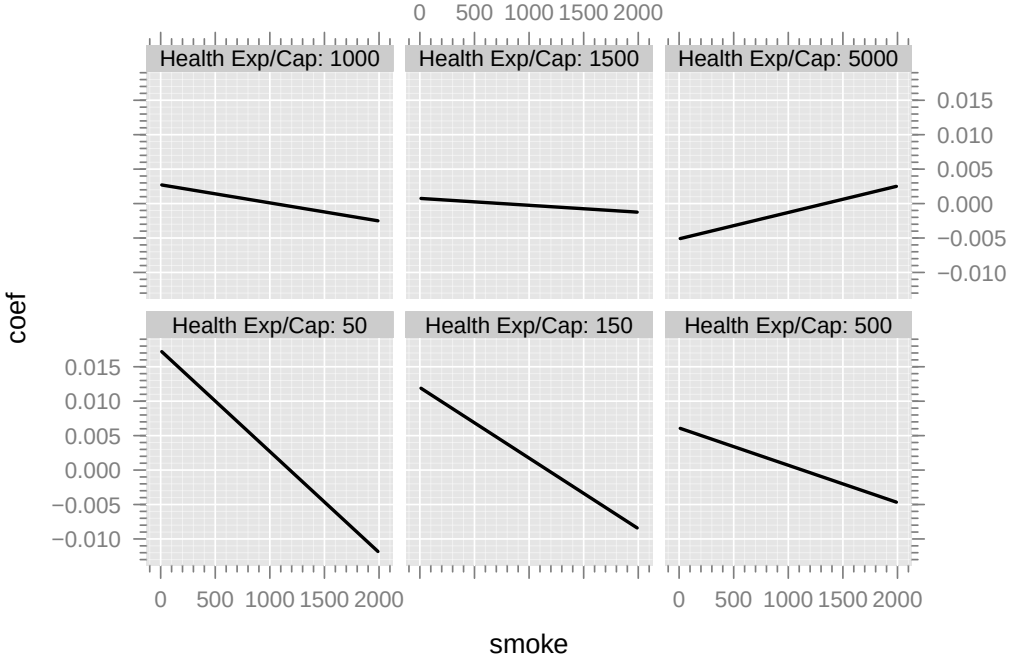
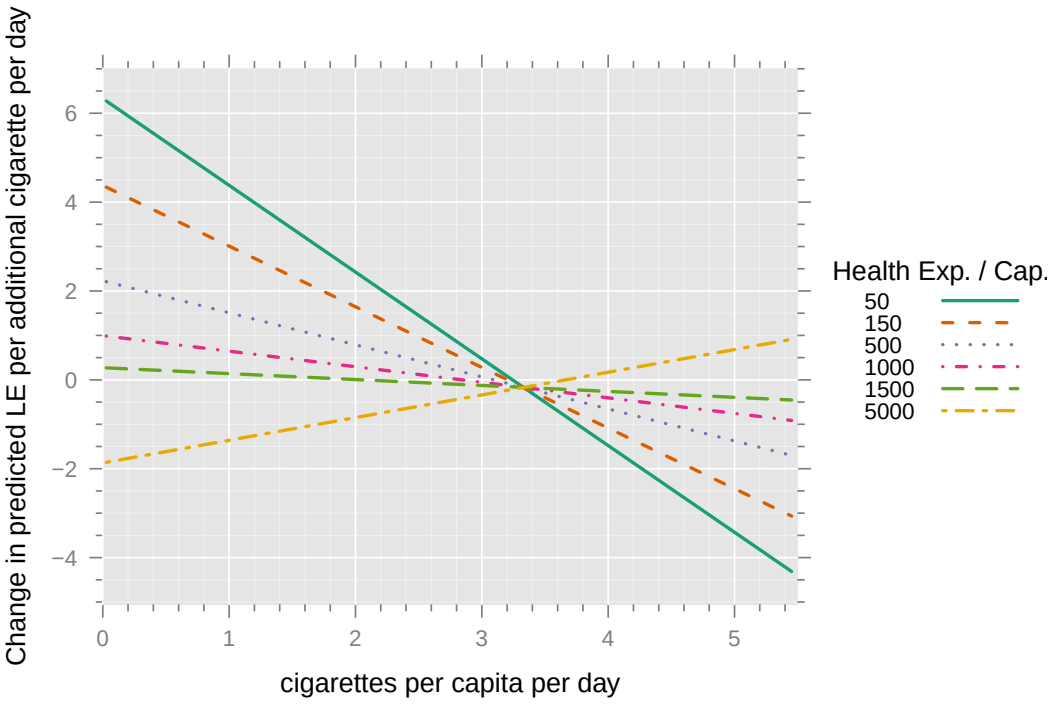

Health Exp/Cap: 1000
Health Exp/Cap: 1500
Health Exp/Cap: 5000
Health Exp/Cap: 50
Health Exp/Cap: 150
Health Exp/Cap: 500
coef
smoke
0
500
1000
1500
2000
0.015
0.010
0.005
0.000
−0.005
−0.010

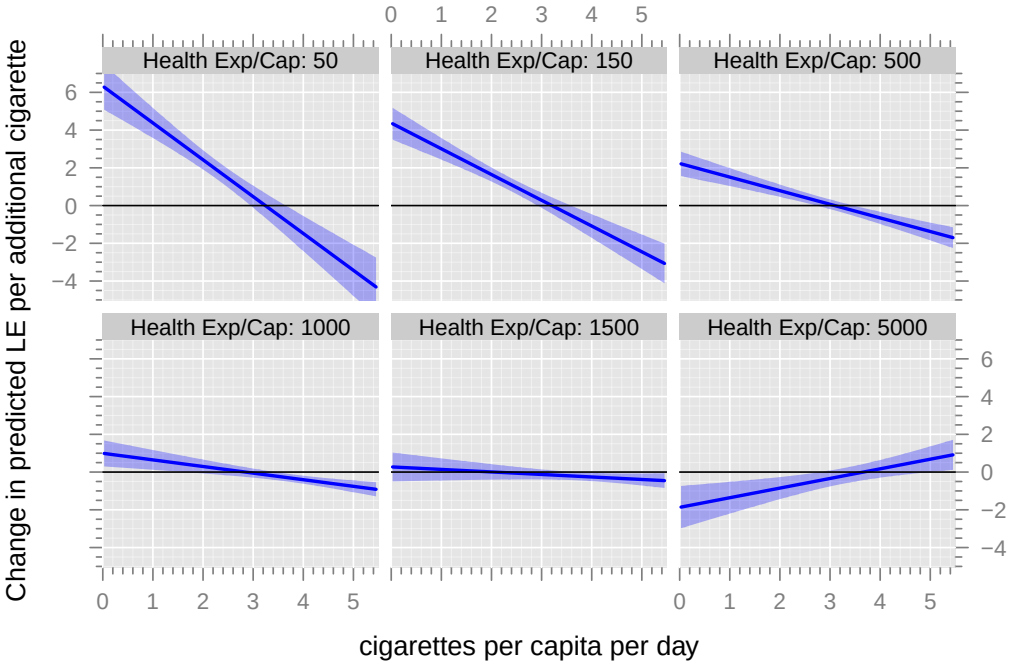
Figure 1: Change in Life Expectancy associated with an increase in cigarette consumption of 1 cigarette per day per capita for different levels of health expenditures per capita per year (US$).

With labels that are more informative:


Change in predicted LE per additional cigarette per day
6
4
2
0
−2
−4
0
1
2
3
4
5
cigarettes per capita per day
Health Exp. / Cap.
50
150
500
1000
1500
5000

Confidence bands:

```
xyplot( I(365*coef)  ~ I(smoke/365) | HEfac,
       data = ww, type = 'n',
       xlim = c(0,5.5),
       ylab = "Change in predicted LE per additional cigarette",
       xlab = "cigarettes per capita per day",
       fit = 365*(ww$coef),
       lower = 365*(ww$coef - ww$se),
       upper = 365*(ww$coef + ww$se),
       subscripts = T,
       as.table = T) +
  layer(panel.fit(...,alpha = .3)) +
  layer( panel.abline( h = 0, lwd = 1))
```


Health Exp/Cap: 50
Health Exp/Cap: 150
Health Exp/Cap: 500
Health Exp/Cap: 1000
Health Exp/Cap: 1500
Health Exp/Cap: 5000
Change in predicted LE per additional cigarette
cigarettes per capita per day
0 1 2 3 4 5
6 4 2 0 −2 −4

Transforming to 'meaningful' coefficients: cigarettes per day

```
## Double checking our previous calculation:
ptest <- expand.grid(smoke = 4, HE = exp(5), hiv = 0, special = 0)

(L2 <- Lfx( fit.hiv2,
      list( 0,
            0 * M(log(HE)),
            1 ,
            1 * M(I(2*smoke)),
            0 * hiv,
            0 * special,
            1 * M(log(HE)) * 1,
            1 * M(log(HE)) * M(I(2*smoke))
      ), ptest))
```

```
|     (Intercept) log(HE) smoke I(smoke^2) hiv special log(HE):smoke
|   1           0       0     1          8   0       0             5
|     log(HE):I(smoke^2)
```

```
|   1                 40
|   attr(,"data")
|     smoke       HE hiv special
|   1     4 148.4132   0       0
```

```
wald(fit.hiv2, list("At smoke = 4, HE = 148.4"=L2))
```

```
|                               numDF denDF F.value p.value
|   At smoke = 4, HE = 148.4        1   141  26.373 <.00001
|     Estimate Std.Error DF  t-value  p-value Lower 0.95 Upper 0.95
|   1 0.011992 0.002335  141 5.135465 <.00001 0.007376   0.016609
```

3.3.1 Exercises

1. Carry out a similar process to estimate the 'effect' of health expenditures per capita.
2. Study the relative contribution of private versus public health expenditures on life expectancy.
3. Explore the 'effects' package and compare its functionality with 'Lfx'

3.4 Wald tests vs Likelihood Ratio Tests (LRT)

Let's consider a test for the need for a quadratic term in 'smoke'. There are two terms in the model that contain the quadratic term and a test to remove it involves more than one parameter. We need a test of

$$H_0 : \beta_3 = \beta_7 = 0$$

We cannot simply test each hypothesis $H_0 : \beta_3 = 0$ and $H_0 : \beta_7 = 0$ separately. We will see many examples where individual hypotheses are not significant, yet a joint hypothesis is highly significant. This is not a example of this phenomenon since the p-value for each hypothesis is small. Nevertheless, a test of a joint hypothesis needs to be carried out correctly. We consider two ways: a Wald test and a Likelihood Ratio Test executed with the 'anova' function, a clear misnomer.

Wald test using indices of coefficients

```
wald(fit.hiv2, c(4,8))
```

```
|     numDF denDF  F.value p.value
```

```
|   1     2   141 8.835643 0.00024
|                      Estimate Std.Error DF  t-value   p-value Lowe
|   I(smoke^2)         -1.5e-05 4e-06     141 -3.846440 0.00018 -2.3
|   log(HE):I(smoke^2)  2.0e-06 1e-06     141  3.504458 0.00061  1.0
|                      Upper 0.95
|   I(smoke^2)         -7e-06
|   log(HE):I(smoke^2)  3e-06
```

```
wald(fit.hiv2, list("Quadratic in smoke" =c(4,8)))
```

```
|                      numDF denDF  F.value p.value
|   Quadratic in smoke     2   141 8.835643 0.00024
|                      Estimate Std.Error DF  t-value   p-value Lowe
|   I(smoke^2)         -1.5e-05 4e-06     141 -3.846440 0.00018 -2.3
|   log(HE):I(smoke^2)  2.0e-06 1e-06     141  3.504458 0.00061  1.0
|                      Upper 0.95
|   I(smoke^2)         -7e-06
|   log(HE):I(smoke^2)  3e-06
```

Wald test using regular expression

```
wald(fit.hiv2, "2")
```

```
|     numDF denDF  F.value p.value
|   2     2   141 8.835643 0.00024
|                       Estimate Std.Error DF  t-value   p-value Lowe
|   I(smoke^2)          -1.5e-05 4e-06     141 -3.846440 0.00018 -2.3
|   log(HE):I(smoke^2)   2.0e-06 1e-06     141  3.504458 0.00061  1.0
|                       Upper 0.95
|   I(smoke^2)          -7e-06
|   log(HE):I(smoke^2)   3e-06
```

Likelihood ratio test

We need to fit the 'null' model

```
fit0 <- update(fit.hiv2, .~ log(HE)*smoke + hiv + special)
summary(fit0)
```

```
|
|   Call:
```

```
lm(formula = LifeExp ~ log(HE) + smoke + hiv + special + log(HE)
    data = dd, na.action = na.exclude)

Residuals:
    Min      1Q  Median      3Q     Max
-9.5894 -2.2654  0.0006  2.4404  9.6020

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)    3.630e+01  2.082e+00  17.438  < 2e-16 ***
log(HE)        5.728e+00  3.613e-01  15.854  < 2e-16 ***
smoke          1.132e-02  3.149e-03   3.596 0.000444 ***
hiv           -7.617e-01  7.896e-02  -9.647  < 2e-16 ***
special       -1.802e+01  2.243e+00  -8.032 3.23e-13 ***
log(HE):smoke -1.660e-03  4.628e-04  -3.586 0.000459 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 3.824 on 143 degrees of freedom
```

```
|     (45 observations deleted due to missingness)
|   Multiple R-squared:  0.8439,    Adjusted R-squared:  0.8384
|   F-statistic: 154.6 on 5 and 143 DF,  p-value: < 2.2e-16
```

Then compare the null model with the 'full' model:

By default, 'anova' uses an F distribution for the LRT taking advantage of the linear Gaussian model.

```
anova(fit0, fit.hiv2)
```

```
|   Analysis of Variance Table
|
|   Model 1: LifeExp ~ log(HE) + smoke + hiv + special + log(HE):smo
|   Model 2: LifeExp ~ log(HE) * (smoke + I(smoke^2)) + hiv + specia
|     Res.Df    RSS Df Sum of Sq      F    Pr(>F)
|   1    143 2090.8
|   2    141 1858.0  2    232.86 8.8356 0.0002426 ***
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Using the general asymptotic distribution, chi-square, for the LRT gives a slightly different but very close result:

```
anova(fit0, fit.hiv2, test="LRT")
```

```
|   Analysis of Variance Table
|
|   Model 1: LifeExp ~ log(HE) + smoke + hiv + special + log(HE):smo
|   Model 2: LifeExp ~ log(HE) * (smoke + I(smoke^2)) + hiv + specia
|     Res.Df    RSS Df Sum of Sq  Pr(>Chi)
|   1    143 2090.8
|   2    141 1858.0  2    232.86 0.0001455 ***
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The Wald test and the LRT using the F statistics give identical results. This is the luxury of working with a Gaussian homoskedastic independent linear model.

3.4.1 Exercises

1. Explore the pros and cons of Wald tests versus Likelihood Ratio Tests. Construct an example where they give entirely different results.

3.5 Interpreting sequential tests

Type 1: sequential tests

```
anova(fit.hiv2)  # sequential - Type 1 tests
```

```
|   Analysis of Variance Table
|
|   Response: LifeExp
|                      Df Sum Sq Mean Sq  F value    Pr(>F)
|   log(HE)             1 8631.5  8631.5 655.0400 < 2.2e-16 ***
|   smoke               1  152.3   152.3  11.5572 0.0008779 ***
|   I(smoke^2)          1  399.7   399.7  30.3307 1.675e-07 ***
|   hiv                 1 1228.0  1228.0  93.1880 < 2.2e-16 ***
|   special             1  878.4   878.4  66.6576 1.637e-13 ***
```

```
|   log(HE):smoke         1   81.6    81.6   6.1932 0.0139881 *
|   log(HE):I(smoke^2)    1  161.8   161.8  12.2812 0.0006136 ***
|   Residuals           141 1858.0    13.2
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Type 2: Each term added last except for higher-order interactions

```
require(car)
Anova(fit.hiv2)  # Type 2 is default
```

```
|   Anova Table (Type II tests)
|
|   Response: LifeExp
|                      Sum Sq  Df  F value    Pr(>F)
|   log(HE)            3199.9   1 242.8357 < 2.2e-16 ***
|   smoke               129.5   1   9.8297 0.0020900 **
|   I(smoke^2)           71.0   1   5.3901 0.0216845 *
|   hiv                1235.0   1  93.7227 < 2.2e-16 ***
|   special             957.9   1  72.6974 2.109e-14 ***
```

```
|   log(HE):smoke        235.0   1  17.8355 4.297e-05 ***
|   log(HE):I(smoke^2)   161.8   1  12.2812 0.0006136 ***
|   Residuals           1858.0 141
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Type 3: Each term added last

Note that 'Type 1' and 'Type 2' sums of squares are interpreted consistently (as far as I've seen) in different packages. 'Type 3', however, has quite different interpretations. Each variable, or group of variables in the case of a factor with 3 or more levels, is added last but different packages set the other variables at different values. Some set them at their 0 values and others set them at their mean values. What does car::Anova do?

```
require(car)
Anova(fit.hiv2, type = 3)
```

```
|   Anova Table (Type III tests)
|
```

```
|   Response: LifeExp
|                       Sum Sq  Df F value    Pr(>F)
|   (Intercept)        1987.03   1 150.794 < 2.2e-16 ***
|   log(HE)            1936.86   1 146.987 < 2.2e-16 ***
|   smoke               309.14   1  23.460 3.315e-06 ***
|   I(smoke^2)          194.96   1  14.795 0.0001809 ***
|   hiv                1235.00   1  93.723 < 2.2e-16 ***
|   special             957.94   1  72.697 2.109e-14 ***
|   log(HE):smoke       235.02   1  17.835 4.297e-05 ***
|   log(HE):I(smoke^2)  161.83   1  12.281 0.0006136 ***
|   Residuals          1857.97 141
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Type 3 is identical to regression output except that it uses equivalent F tests and a single test for terms with multiple degrees of freedom

```
summary(fit.hiv2)
```

```
|
```

```
Call:
lm(formula = LifeExp ~ log(HE) * (smoke + I(smoke^2)) + hiv +
    special, data = dd, na.action = na.exclude)

Residuals:
    Min      1Q  Median      3Q     Max
-9.0373 -2.3005  0.2043  2.0760  9.7344

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)         3.283e+01  2.674e+00  12.280  < 2e-16 ***
log(HE)             6.091e+00  5.024e-01  12.124  < 2e-16 ***
smoke               3.642e-02  7.520e-03   4.844 3.31e-06 ***
I(smoke^2)         -1.518e-05  3.946e-06  -3.846 0.000181 ***
hiv                -7.351e-01  7.593e-02  -9.681  < 2e-16 ***
special            -1.822e+01  2.137e+00  -8.526 2.11e-14 ***
log(HE):smoke      -4.878e-03  1.155e-03  -4.223 4.30e-05 ***
log(HE):I(smoke^2)  2.007e-06  5.726e-07   3.504 0.000614 ***
---
```

```
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
|
|   Residual standard error: 3.63 on 141 degrees of freedom
|     (45 observations deleted due to missingness)
|   Multiple R-squared:  0.8613,    Adjusted R-squared:  0.8544
|   F-statistic:   125 on 7 and 141 DF,  p-value: < 2.2e-16
```

3.6 Working with factors

Controlling for WHO regions provides a non-trivial example of the use of factors in regression.

When you create a data frame in R, non-numeric variables are automatically turned into **factors**. Factors are both a strength of R and a frequent source of annoyance and confusion. See traps and pitfalls with factors.

Let's create a small data frame to illustrate how factors work:

```
set.seed(147)
sdf <- data.frame( x = c(1:7,6:10),
```

```
            g = rep(c('a','b','c'),c(2,5,5)))
sdf
```

```
|        x g
|    1   1 a
|    2   2 a
|    3   3 b
|    4   4 b
|    5   5 b
|    6   6 b
|    7   7 b
|    8   6 c
|    9   7 c
|    10  8 c
|    11  9 c
|    12 10 c
```

```
sdf$y <- with(sdf, x + c(1,0,2)[g]+.5* rnorm(12))
sdf
```

```
|        x g         y
|    1    1 a  2.120108
|    2    2 a  2.853852
|    3    3 b  2.722852
|    4    4 b  4.825593
|    5    5 b  4.623128
|    6    6 b  5.732132
|    7    7 b  7.221543
|    8    6 c  7.060536
|    9    7 c  9.465820
|    10   8 c  9.748093
|    11   9 c 11.580972
|    12  10 c 11.661232
```

```
sdf$g
```

```
|     [1] a a b b b b b c c c c c
|    Levels: a b c
```

```r
unclass(sdf$g)    # the innards of g
```

```
|    [1] 1 1 2 2 2 2 2 3 3 3 3 3
|   attr(,"levels")
|   [1] "a" "b" "c"
```

g is actually a numeric variable consisting of indices into a vector of 'levels'.

```r
sfit <- lm( y ~ x + g, sdf, na.action = na.exclude)
summary(sfit)
```

```
|
|   Call:
|   lm(formula = y ~ x + g, data = sdf, na.action = na.exclude)
|
|   Residuals:
|       Min      1Q  Median      3Q     Max
|   -0.7367 -0.3465 -0.1574  0.2736  0.8536
|
|   Coefficients:
```

```
|              Estimate Std. Error t value Pr(>|t|)
|   (Intercept)   0.9074     0.4421   2.052   0.0742 .
|   x             1.0530     0.1250   8.421 3.01e-05 ***
|   gb           -1.1476     0.6449  -1.779   0.1131
|   gc            0.5716     0.9408   0.608   0.5603
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
|
|   Residual standard error: 0.5662 on 8 degrees of freedom
|   Multiple R-squared:  0.9797,    Adjusted R-squared:  0.972
|   F-statistic: 128.4 on 3 and 8 DF,  p-value: 4.178e-07
```

Note:

- There is no term called 'ga' although there are 3 levels: 'a', 'b' and 'c'
- The 'missing' level, 'a', is called the **reference level**
- Each term shows a comparison with the reference level

To work out what the coefficients for 'gb' and 'gc' mean, you need to look at the X matrix:

```r
model.matrix(sfit, na.action = na.exclude)
```

```
|      (Intercept)  x gb gc
|   1            1  1  0  0
|   2            1  2  0  0
|   3            1  3  1  0
|   4            1  4  1  0
|   5            1  5  1  0
|   6            1  6  1  0
|   7            1  7  1  0
|   8            1  6  0  1
|   9            1  7  0  1
|   10           1  8  0  1
|   11           1  9  0  1
|   12           1 10  0  1
|   attr(,"assign")
|   [1] 0 1 2 2
|   attr(,"contrasts")
```

```
|   attr(,"contrasts")$g
|   [1] "contr.treatment"
```

```
model.matrix(~ x + g, sdf)
```

```
|      (Intercept)  x gb gc
|   1            1  1  0  0
|   2            1  2  0  0
|   3            1  3  1  0
|   4            1  4  1  0
|   5            1  5  1  0
|   6            1  6  1  0
|   7            1  7  1  0
|   8            1  6  0  1
|   9            1  7  0  1
|   10           1  8  0  1
|   11           1  9  0  1
|   12           1 10  0  1
|   attr(,"assign")
```

```
|   [1] 0 1 2 2
|   attr(,"contrasts")
|   attr(,"contrasts")$g
|   [1] "contr.treatment"
```

```
model.matrix(~ g, sdf)
```

```
|      (Intercept) gb gc
|   1            1  0  0
|   2            1  0  0
|   3            1  1  0
|   4            1  1  0
|   5            1  1  0
|   6            1  1  0
|   7            1  1  0
|   8            1  0  1
|   9            1  0  1
|   10           1  0  1
|   11           1  0  1
```

```
|   12           1  0  1
|   attr(,"assign")
|   [1] 0 1 1
|   attr(,"contrasts")
|   attr(,"contrasts")$g
|   [1] "contr.treatment"
```

If you work through the model:

$$E(y|x,g) = \beta_0 + \beta_x x + \beta_{gb} gb + \beta_{gc} gc$$

where $gb = 1$ if $g = b$ and 0 otherwise, and $gc = 1$ if $g = c$ and 0 otherwise, you see that

1. β_{gb} is the difference between the expected level for group 'b' versus the reference group 'a' keeping x constant and
2. β_{gc} is the same comparison for group 'c' compared with the reference group 'a'.

Plotting fits within groups and panels using latticeExtra:

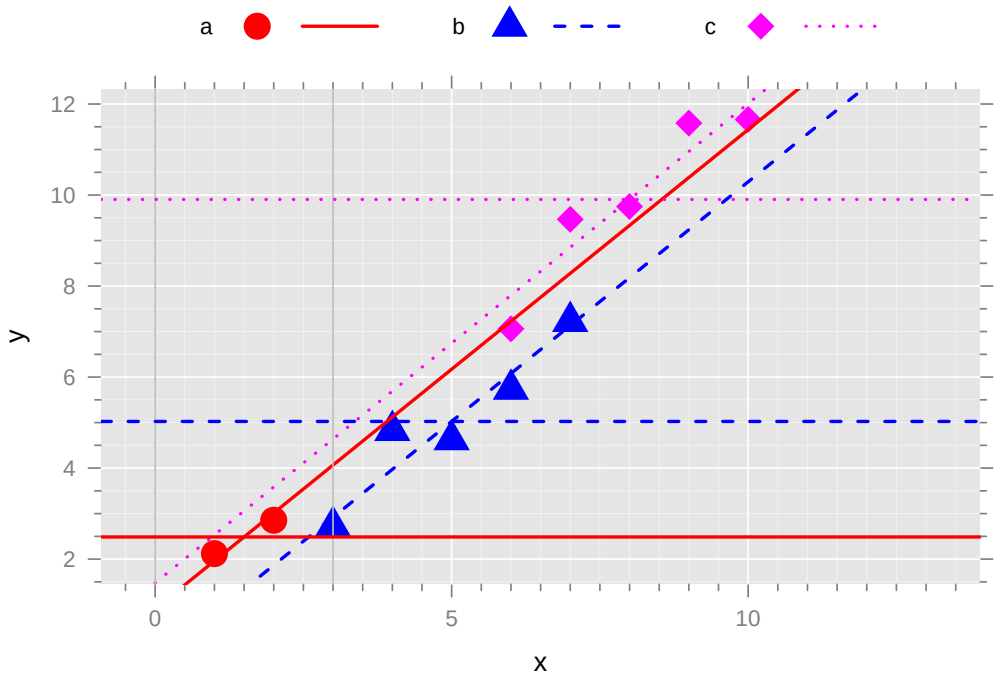
See more elegant but perhaps less flexible approaches in the 'car' and in the 'effects' package by John Fox

```
pred <- expand.grid( x = 0:13, g = levels( sdf$g))
```

The values over which we want to see predicted lines every combination of x and g

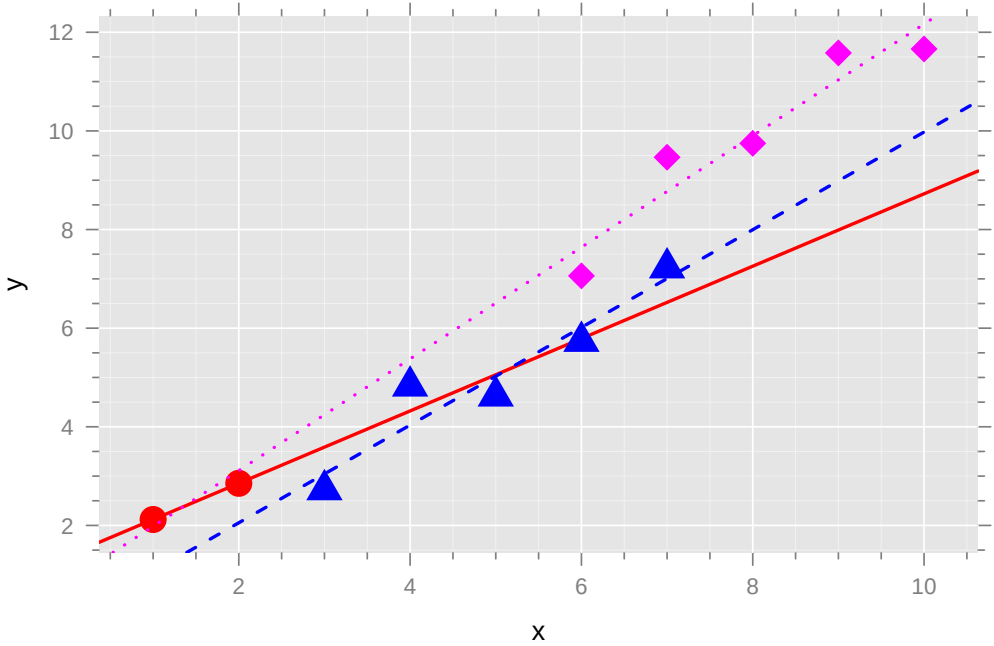
```
pred  <- merge( sdf, pred, all = T) # merge with data
pred$y1 <- predict(sfit, newdata = pred) # the predicted value
pred <- sortdf(pred, ~ x) # order so lines won't be interrupted
require(latticeExtra)
require(spida2)
gd(cex=2, lty=1:3,  # this gives control over line styles,
                    # colour, etc.
   pch = 16:18,
   col = c('red','blue','magenta'),
   lwd =2)  # from spida2
xyplot( y ~ x , pred, groups = g ,
        auto.key = list(columns = 3,lines = T),
```

```
      y1 = pred$y1, subscripts = T,
      sub =
"compare adjusted and unadjusted differences between groups") +
glayer( panel.lines( x, y1[subscripts],...,type = 'l')) +
layer( panel.abline( v = c(0,3), col = 'grey')) +
glayer( panel.abline( h = mean(y,na.rm=T),...))
```



compare adjusted and unadjusted differences between groups

```
## an alternative ... but
xyplot( y ~ x, sdf, groups = g, type = c('p','r'))
```


12
10
8
6
4
2
y
2
4
6
8
10
x

3.6.1 Exercises:

1. Draw by hand the values of estimated coefficients in the plot.
2. How would you estimate the differences between horizontal lines?
3. Does 'g' matter?

```
summary(sfit)  # p-values not significant
```

```
|
|   Call:
|   lm(formula = y ~ x + g, data = sdf, na.action = na.exclude)
|
|   Residuals:
|       Min      1Q  Median      3Q     Max
|   -0.7367 -0.3465 -0.1574  0.2736  0.8536
|
|   Coefficients:
|               Estimate Std. Error t value Pr(>|t|)
|   (Intercept)   0.9074     0.4421   2.052   0.0742 .
|   x             1.0530     0.1250   8.421 3.01e-05 ***
```

```
|   gb           -1.1476     0.6449  -1.779   0.1131
|   gc            0.5716     0.9408   0.608   0.5603
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
|
|   Residual standard error: 0.5662 on 8 degrees of freedom
|   Multiple R-squared:  0.9797,    Adjusted R-squared:  0.972
|   F-statistic: 128.4 on 3 and 8 DF,  p-value: 4.178e-07
```

But

```
wald(sfit, "g")  # simultaneous test that both are 0: different ans
```

```
|     numDF denDF  F.value p.value
|   g     2     8 9.711573 0.00724
|      Estimate  Std.Error DF t-value   p-value Lower 0.95 Upper 0.9
|   gb -1.147573 0.644944  8  -1.779337 0.11306 -2.634817  0.339671
|   gc  0.571585 0.940783  8   0.607563 0.56032 -1.597865  2.741035
```

Using the GLH (General Linear Hypothesis)

```
Lmu.6 <-list( "at x = 6" = rbind(
 'g = a' = c(1,6,0,0),
 'g = b' = c(1,6,1,0),
 'g = c' = c(1,6,0,1)))
Lmu.6
```

```
|   $`at x = 6`
|         [,1] [,2] [,3] [,4]
|   g = a    1    6    0    0
|   g = b    1    6    1    0
|   g = c    1    6    0    1
```

```
wald(sfit, Lmu.6)
```

```
|             numDF denDF  F.value p.value
|   at x = 6     3     8 613.0644 <.00001
|          Estimate Std.Error DF t-value  p-value Lower 0.95 Upper 0.
|   g = a 7.225664 0.690607  8  10.46277 1e-05    5.633120   8.818207
|   g = b 6.078091 0.282401  8  21.52290 <.00001 5.426872   6.729309
```

```
|   g = c 7.797249 0.355897  8  21.90875 <.00001 6.976550   8.617948
```

```
Ldiff <- rbind(
 'b - a' = c(0,0,1,0),
 'c - a' = c(0,0,0,1),
 'c - b' = c(0,0,-1,1))
Ldiff
```

```
|         [,1] [,2] [,3] [,4]
|   b - a    0    0    1    0
|   c - a    0    0    0    1
|   c - b    0    0   -1    1
```

```
wald(sfit,Ldiff)
```

```
|     numDF denDF  F.value p.value
|   1     2     8 9.711573 0.00724
|         Estimate  Std.Error DF t-value   p-value Lower 0.95 Upper
|   b - a -1.147573 0.644944  8  -1.779337 0.11306 -2.634817  0.3396
|   c - a  0.571585 0.940783  8   0.607563 0.56032 -1.597865  2.7410
```

```
|   c - b  1.719159 0.518616  8   3.314900 0.01062  0.523229  2.9150
```

```
wald(sfit, 'g')
```

```
|     numDF denDF  F.value p.value
|   g     2     8 9.711573 0.00724
|      Estimate  Std.Error DF t-value   p-value Lower 0.95 Upper 0.9
|   gb -1.147573 0.644944  8  -1.779337 0.11306 -2.634817  0.339671
|   gc  0.571585 0.940783  8   0.607563 0.56032 -1.597865  2.741035
```

This illustrated the crucial point that separate tests of

$$H_0 : \beta_1 = 0$$

and

$$H_0 : \beta_2 = 0$$

can yield very different 'conclusions' that a test of the joint hypothesis:

$$H_0 : \beta_1 = \beta_2 = 0$$

Later, we will see how the relationship between confidence ellipses(oids) and tests makes this clear.

3.6.2 Reparametrization to answer different questions

```
sdf$g2 <- relevel(sdf$g, 'b')  # makes 'b' the reference level
fitr <- lm( y ~ x + g2, sdf)
summary(fitr)
```

```
|
|   Call:
|   lm(formula = y ~ x + g2, data = sdf)
|
|   Residuals:
|       Min      1Q  Median      3Q     Max
|   -0.7367 -0.3465 -0.1574  0.2736  0.8536
|
|   Coefficients:
|               Estimate Std. Error t value Pr(>|t|)
|   (Intercept)  -0.2402     0.6746  -0.356   0.7310
|   x             1.0530     0.1250   8.421 3.01e-05 ***
|   g2a           1.1476     0.6449   1.779   0.1131
```

```
|   g2c           1.7192     0.5186   3.315   0.0106 *
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
|
|   Residual standard error: 0.5662 on 8 degrees of freedom
|   Multiple R-squared:  0.9797,    Adjusted R-squared:  0.972
|   F-statistic: 128.4 on 3 and 8 DF,  p-value: 4.178e-07
```

```r
fitr2 <- lm( y ~ x + g2 -1 , sdf)   # dropping the intercept
summary(fitr2)
```

```
|
|   Call:
|   lm(formula = y ~ x + g2 - 1, data = sdf)
|
|   Residuals:
|       Min      1Q  Median      3Q     Max
|   -0.7367 -0.3465 -0.1574  0.2736  0.8536
|
```

```
|   Coefficients:
|       Estimate Std. Error t value Pr(>|t|)
|   x     1.0530     0.1250   8.421 3.01e-05 ***
|   g2b  -0.2402     0.6746  -0.356   0.7310
|   g2a   0.9074     0.4421   2.052   0.0742 .
|   g2c   1.4790     1.0319   1.433   0.1897
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
|
|   Residual standard error: 0.5662 on 8 degrees of freedom
|   Multiple R-squared:  0.9961,    Adjusted R-squared:  0.9941
|   F-statistic: 508.3 on 4 and 8 DF,  p-value: 1.176e-09
```

```r
fitr3 <- lm( y ~ I(x-6) + g2 -1 , sdf)   # recentering x
summary(fitr3)   # compare with earlier
```

```
|
|   Call:
|   lm(formula = y ~ I(x - 6) + g2 - 1, data = sdf)
```

```
|
|   Residuals:
|       Min      1Q  Median      3Q     Max
|   -0.7367 -0.3465 -0.1574  0.2736  0.8536
|
|   Coefficients:
|             Estimate Std. Error t value Pr(>|t|)
|   I(x - 6)   1.0530     0.1250   8.421 3.01e-05 ***
|   g2b        6.0781     0.2824  21.523 2.29e-08 ***
|   g2a        7.2257     0.6906  10.463 6.05e-06 ***
|   g2c        7.7972     0.3559  21.909 1.99e-08 ***
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
|
|   Residual standard error: 0.5662 on 8 degrees of freedom
|   Multiple R-squared:  0.9961,    Adjusted R-squared:  0.9941
|   F-statistic: 508.3 on 4 and 8 DF,  p-value: 1.176e-09
```

```
wald(sfit, Lmu.6)
```

```
|            numDF denDF  F.value p.value
|   at x = 6     3     8 613.0644 <.00001
|         Estimate Std.Error DF t-value  p-value Lower 0.95 Upper 0.
|   g = a 7.225664 0.690607  8  10.46277 1e-05   5.633120   8.818207
|   g = b 6.078091 0.282401  8  21.52290 <.00001 5.426872   6.729309
|   g = c 7.797249 0.355897  8  21.90875 <.00001 6.976550   8.617948
```

3.6.3 Equivalent models

What makes the last three models equivalent?

```
summary(lm( model.matrix(fitr) ~ model.matrix(sfit)-1))
```

```
|   Warning in summary.lm(object, ...): essentially perfect fit: sum
|   unreliable

|   Warning in summary.lm(object, ...): essentially perfect fit: sum
|   unreliable
```

```
|   Warning in summary.lm(object, ...): essentially perfect fit: sum
|   unreliable

|   Warning in summary.lm(object, ...): essentially perfect fit: sum
|   unreliable

|   Response (Intercept) :
|
|   Call:
|   lm(formula = `(Intercept)` ~ model.matrix(sfit) - 1)
|
|   Residuals:
|          Min           1Q       Median           3Q          Max
|   -7.504e-16 -4.690e-17  0.000e+00  4.690e-17  7.504e-16
|
|   Coefficients:
|                                      Estimate Std. Error    t value P
|   model.matrix(sfit)(Intercept)  1.000e+00  2.966e-16  3.371e+15
```

```
|   model.matrix(sfit)x          -3.752e-17  8.390e-17 -4.470e-01
|   model.matrix(sfit)gb         -6.379e-16  4.327e-16 -1.474e+00
|   model.matrix(sfit)gc         -5.253e-16  6.312e-16 -8.320e-01
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
|
|   Residual standard error: 3.799e-16 on 8 degrees of freedom
|   Multiple R-squared:      1, Adjusted R-squared:      1
|   F-statistic: 2.079e+31 on 4 and 8 DF,  p-value: < 2.2e-16
|
|
|   Response x :
|
|   Call:
|   lm(formula = x ~ model.matrix(sfit) - 1)
|
|   Residuals:
|         Min          1Q      Median          3Q         Max
|   -8.742e-16 -8.917e-17 -2.706e-17  1.417e-16  8.742e-16
```

```
|
|   Coefficients:
|                                    Estimate Std. Error    t value P
|   model.matrix(sfit)(Intercept) -1.026e-15  3.623e-16 -2.831e+00
|   model.matrix(sfit)x            1.000e+00  1.025e-16  9.759e+15
|   model.matrix(sfit)gb          -8.052e-16  5.285e-16 -1.523e+00
|   model.matrix(sfit)gc          -9.918e-16  7.709e-16 -1.287e+00
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
|
|   Residual standard error: 4.64e-16 on 8 degrees of freedom
|   Multiple R-squared:      1, Adjusted R-squared:      1
|   F-statistic: 5.459e+32 on 4 and 8 DF,  p-value: < 2.2e-16
|
|
|   Response g2a :
|
|   Call:
|   lm(formula = g2a ~ model.matrix(sfit) - 1)
```

```
|
|   Residuals:
|         Min          1Q      Median          3Q         Max
|   -5.639e-17 -2.701e-17 -8.327e-18  3.385e-17  6.024e-17
|
|   Coefficients:
|                                     Estimate Std. Error    t value P
|   model.matrix(sfit)(Intercept)  1.000e+00  3.675e-17  2.721e+16
|   model.matrix(sfit)x           -4.223e-17  1.039e-17 -4.063e+00
|   model.matrix(sfit)gb          -1.000e+00  5.361e-17 -1.865e+16
|   model.matrix(sfit)gc          -1.000e+00  7.820e-17 -1.279e+16
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
|
|   Residual standard error: 4.706e-17 on 8 degrees of freedom
|   Multiple R-squared:      1, Adjusted R-squared:      1
|   F-statistic: 2.258e+32 on 4 and 8 DF,  p-value: < 2.2e-16
|
|
```

```
Response g2c :

Call:
lm(formula = g2c ~ model.matrix(sfit) - 1)

Residuals:
       Min         1Q     Median         3Q        Max
-1.054e-17 -6.564e-18 -8.950e-20  4.320e-18  1.981e-17

Coefficients:
                                Estimate Std. Error  t value Pr(>
model.matrix(sfit)(Intercept) 0.000e+00  7.899e-18 0.00e+00
model.matrix(sfit)x           0.000e+00  2.234e-18 0.00e+00
model.matrix(sfit)gb          0.000e+00  1.152e-17 0.00e+00
model.matrix(sfit)gc          1.000e+00  1.681e-17 5.95e+16   <2
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.012e-17 on 8 degrees of freedom
```

```
|   Multiple R-squared:      1, Adjusted R-squared:      1
|   F-statistic: 1.222e+34 on 4 and 8 DF,  p-value: < 2.2e-16
```

```
   # note the Resid. SE
```

Each model matrix spans exactly the same linear space. Thus their columns are just different bases for the same space and the βs for one model are just a linear transformation of the βs for the other model.

3.6.4 Exercise:

What do the coefficients estimate in each of the following models?. Indicate how each coefficient is related to the first graph shown below.

Factor alone: g - 1

```
summary( fit1 <- lm( y ~ g - 1, sdf))
```

```
|
|   Call:
|   lm(formula = y ~ g - 1, data = sdf)
```

```
|
|   Residuals:
|       Min      1Q  Median      3Q     Max
|   -2.8428 -0.4108 -0.1774  0.9497  2.1965
|
|   Coefficients:
|      Estimate Std. Error t value Pr(>|t|)
|   ga   2.4870     1.1855   2.098   0.0653 .
|   gb   5.0250     0.7498   6.702 8.83e-05 ***
|   gc   9.9033     0.7498  13.209 3.39e-07 ***
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
|
|   Residual standard error: 1.677 on 9 degrees of freedom
|   Multiple R-squared:  0.9613,    Adjusted R-squared:  0.9485
|   F-statistic: 74.59 on 3 and 9 DF,  p-value: 1.118e-06
```

```
    # an example where '=' would not work
model.matrix(fit1)
```

```
|       ga gb gc
|    1   1  0  0
|    2   1  0  0
|    3   0  1  0
|    4   0  1  0
|    5   0  1  0
|    6   0  1  0
|    7   0  1  0
|    8   0  0  1
|    9   0  0  1
|    10  0  0  1
|    11  0  0  1
|    12  0  0  1
|    attr(,"assign")
|    [1] 1 1 1
|    attr(,"contrasts")
|    attr(,"contrasts")$g
|    [1] "contr.treatment"
```

```r
summary( fit2 <- lm( y ~ x + g - 1, sdf))
```

```
|
|   Call:
|   lm(formula = y ~ x + g - 1, data = sdf)
|
|   Residuals:
|       Min      1Q  Median      3Q     Max
|   -0.7367 -0.3465 -0.1574  0.2736  0.8536
|
|   Coefficients:
|      Estimate Std. Error t value Pr(>|t|)
|   x    1.0530     0.1250   8.421 3.01e-05 ***
|   ga   0.9074     0.4421   2.052   0.0742 .
|   gb  -0.2402     0.6746  -0.356   0.7310
|   gc   1.4790     1.0319   1.433   0.1897
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
|
|   Residual standard error: 0.5662 on 8 degrees of freedom
|   Multiple R-squared:  0.9961,    Adjusted R-squared:  0.9941
|   F-statistic: 508.3 on 4 and 8 DF,  p-value: 1.176e-09
```

```r
    # an example where '=' would not work
model.matrix(fit2)
```

```
|       x ga gb gc
|   1   1  1  0  0
|   2   2  1  0  0
|   3   3  0  1  0
|   4   4  0  1  0
|   5   5  0  1  0
|   6   6  0  1  0
|   7   7  0  1  0
|   8   6  0  0  1
|   9   7  0  0  1
|   10  8  0  0  1
|   11  9  0  0  1
```

```
|   12 10  0  0  1
|   attr(,"assign")
|   [1] 1 2 2 2
|   attr(,"contrasts")
|   attr(,"contrasts")$g
|   [1] "contr.treatment"
```

Factor with interaction: g * x

```
sfit2 <- lm( y ~ g * x, sdf)
summary(sfit2)
```

```
|
|   Call:
|   lm(formula = y ~ g * x, data = sdf)
|
|   Residuals:
|        Min       1Q   Median       3Q      Max
|   -0.57949 -0.34154 -0.07762  0.29828  0.79094
|
```

```
|   Coefficients:
|               Estimate Std. Error t value Pr(>|t|)
|   (Intercept)   1.3864     1.4179   0.978    0.366
|   gb           -1.3133     1.7596  -0.746    0.484
|   gc           -0.5363     2.1597  -0.248    0.812
|   x             0.7337     0.8968   0.818    0.445
|   gb:x          0.2566     0.9189   0.279    0.789
|   gc:x          0.3979     0.9189   0.433    0.680
|
|   Residual standard error: 0.6341 on 6 degrees of freedom
|   Multiple R-squared:  0.9809,    Adjusted R-squared:  0.9649
|   F-statistic: 61.51 on 5 and 6 DF,  p-value: 4.499e-05
```

From which you might conclude that 'nothing is significant'!

This illustrates that it is often wrong *wrong* **wrong** to form conclusions on the basis of scanning p-values in regression output.

Factor nesting continuous variable: g / x - 1

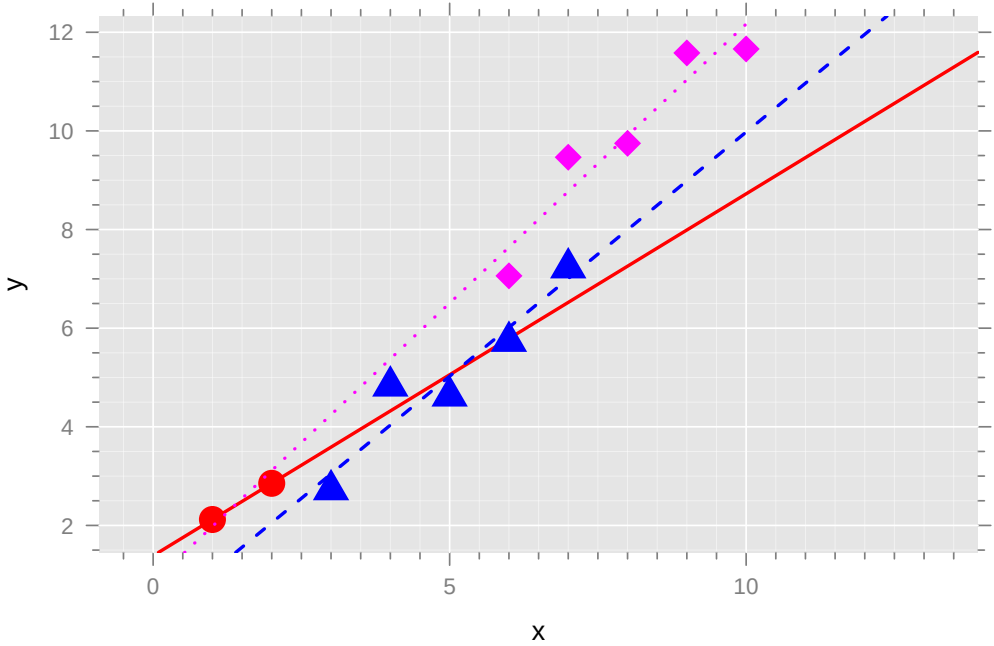
```
sfit3 <- lm( y ~ g / x - 1 , sdf)
summary(sfit3)
```

```
|
|   Call:
|   lm(formula = y ~ g/x - 1, data = sdf)
|
|   Residuals:
|        Min       1Q   Median       3Q      Max
|   -0.57949 -0.34154 -0.07762  0.29828  0.79094
|
|   Coefficients:
|        Estimate Std. Error t value Pr(>|t|)
|   ga    1.38636    1.41789   0.978  0.36595
|   gb    0.07309    1.04193   0.070  0.94636
|   gc    0.85010    1.62903   0.522  0.62048
|   ga:x  0.73374    0.89675   0.818  0.44450
|   gb:x  0.99039    0.20052   4.939  0.00261 **
```

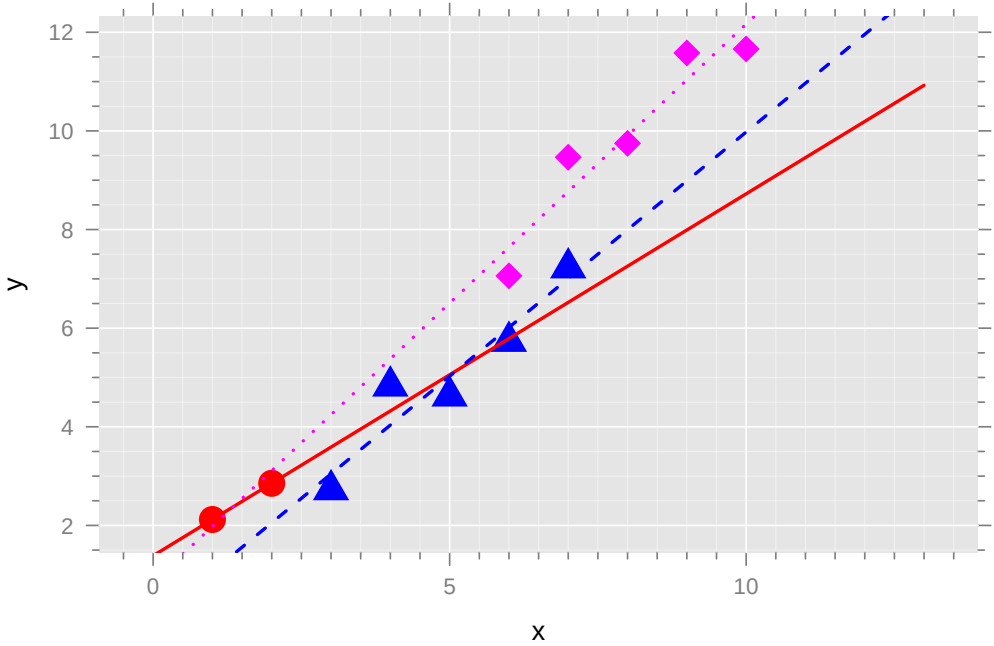
```
|   gc:x  1.13165    0.20052   5.644  0.00133 **
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
|
|   Residual standard error: 0.6341 on 6 degrees of freedom
|   Multiple R-squared:  0.9963,    Adjusted R-squared:  0.9926
|   F-statistic: 270.2 on 6 and 6 DF,  p-value: 4.985e-07
```

Plotting the fitted model:

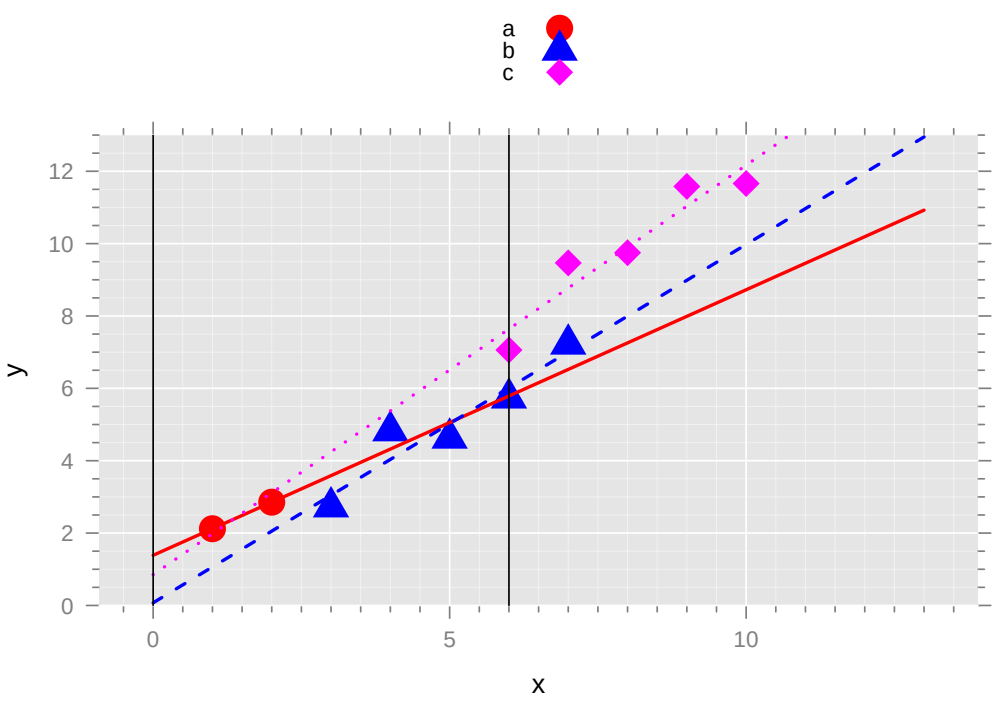
```
pred$y2 <- predict( sfit2, newdata = pred)
xyplot( y ~ x, pred, groups = g, type = c('p','r'))
```


12
10
8
6
4
2
y
0
5
10
x

```r
# or
xyplot( y ~ x, pred, groups = g,
  subscripts = T, y2 = pred$y2, y1 = pred$y1) +
glayer( panel.xyplot( x, y2[subscripts], ..., type = 'l'))
```


12
10
8
6
4
2
y
0
5
10
x

```r
# or
xyplot( y ~ x, pred, groups = g,
  ylim = c(0,13), auto.key = T,
  subscripts = T, y2 = pred$y2, y1 = pred$y1) +
glayer( panel.xyplot( x, y2[subscripts], ..., type = 'l')) +
layer( panel.abline( v = c(0,6)))
```


a
b
c
12
10
8
6
4
2
0
y
0
5
10
x

The model is:

$$E(y|x,g) = \beta_0 + \beta_x x + \beta_{gb} gb + \beta_{gc} gc + \beta_{x:gb} x \times gb + \beta_{x:gc} x \times gc$$

Taking partial derivatives, we see that β_{gb} is the difference between between group 'b' minus group 'a' when $x = 0$. There might not be strong evidence of differences between groups outside the range of the data.

What would happen if we were to explore the difference between group 'b' and group 'c' when $x = 6$:

```
L.bc.6 <- rbind( 'c - b|x=6'=
             c(0,0,-1,1,-6,6))
wald( sfit2, L.bc.6)
```

```
|     numDF denDF   F.value p.value
|   1     1     6 0.4570639 0.52419
|              Estimate Std.Error DF t-value  p-value Lower 0.95 Uppe
|   c - b|x=6 2.117587 3.132223  6  0.676065 0.52419 -5.546688  9.78
```

We could also do this by reparametrizing:

```r
sfit2.x6 <- lm( y ~ I(x-6) * relevel(g,'b'), sdf)
summary(sfit2.x6)
```

```
|
|   Call:
|   lm(formula = y ~ I(x - 6) * relevel(g, "b"), data = sdf)
|
|   Residuals:
|        Min       1Q   Median       3Q      Max
|   -0.57949 -0.34154 -0.07762  0.29828  0.79094
|
|   Coefficients:
|                              Estimate Std. Error t value Pr(>|t|)
|   (Intercept)                  6.0154     0.3473  17.320 2.37e-06 *
|   I(x - 6)                     0.9904     0.2005   4.939  0.00261 *
|   relevel(g, "b")a            -0.2266     4.0750  -0.056  0.95746
|   relevel(g, "b")c             1.6246     0.6016   2.701  0.03555 *
|   I(x - 6):relevel(g, "b")a   -0.2566     0.9189  -0.279  0.78939
```

```
|   I(x - 6):relevel(g, "b")c   0.1413     0.2836   0.498  0.63611
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
|
|   Residual standard error: 0.6341 on 6 degrees of freedom
|   Multiple R-squared:  0.9809,    Adjusted R-squared:  0.9649
|   F-statistic: 61.51 on 5 and 6 DF,  p-value: 4.499e-05
```

Here are some other ways of exploring the model:

```
wald(sfit2, ":")
```

```
|     numDF denDF   F.value p.value
|   :     2     6 0.1890466 0.83249
|        Estimate Std.Error DF t-value p-value Lower 0.95 Upper 0.95
|   gb:x 0.256648 0.918898  6  0.27930 0.78939 -1.991815  2.505111
|   gc:x 0.397910 0.918898  6  0.43303 0.68013 -1.850553  2.646374
```

```
wald(sfit2, "g")
```

```
|     numDF denDF  F.value p.value
```

```
|   g     4     6 3.965856 0.06566
|         Estimate  Std.Error DF t-value   p-value Lower 0.95 Upper 0
|   gb    -1.313275 1.759556  6  -0.746367 0.48365 -5.618754  2.99220
|   gc    -0.536269 2.159667  6  -0.248311 0.81217 -5.820784  4.74824
|   gb:x   0.256648 0.918898  6   0.279300 0.78939 -1.991815  2.50511
|   gc:x   0.397910 0.918898  6   0.433030 0.68013 -1.850553  2.64637
```

```
wald(sfit2, "x")
```

```
|     numDF denDF  F.value p.value
|   x     3     6 18.97152 0.00183
|         Estimate Std.Error DF t-value  p-value Lower 0.95 Upper 0.9
|   x     0.733744 0.896753  6  0.818223 0.44450 -1.460531  2.928020
|   gb:x 0.256648 0.918898  6  0.279300 0.78939 -1.991815  2.505111
|   gc:x 0.397910 0.918898  6  0.433030 0.68013 -1.850553  2.646374
```

Using type 2 Anova gives you tests for 'g' and 'x' that assume that higher-order interactions involving 'g' and 'x' are all 0. Note that the error term used is the error term for the full model including interactions. This can lead to inconsistencies with tests based on a model in which interactions has been

dropped.

```
Anova(sfit2)    # type 2 anova
```

```
|   Anova Table (Type II tests)
|
|   Response: y
|              Sum Sq Df F value    Pr(>F)
|   g           6.2264  2  7.7427 0.0217785 *
|   x          22.7323  1 56.5365 0.0002865 ***
|   g:x         0.1520  2  0.1890 0.8324941
|   Residuals   2.4125  6
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Anova(sfit)    # here the gain in degrees of freedom
```

```
|   Anova Table (Type II tests)
|
|   Response: y
```

```
|               Sum Sq Df F value    Pr(>F)
|   x          22.7323  1 70.9133 3.013e-05 ***
|   g           6.2264  2  9.7116  0.007243 **
|   Residuals   2.5645  8
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
                # outweighs the increase in SSE
```

3.7 Using Lfx with factors

The 'M' function associated with 'Lfx' can generate code to test for differences between factor levels

```
Lfx(sfit2)
```

```
|   list( 1,
|   1 * M(g),
|   1 * x,
|   1 * M(g) * x
```

```
|   )
```

The idea is to use the 'Lfx' expression to difference and then to apply to a data frame.

```
dpred <- expand.grid(x = 0:12,
                     g = levels(sdf$g) ,
                     g0 = levels(sdf$g))
dim(dpred)
```

```
|   [1] 117   3
```

```
some(dpred)
```

```
|        x g g0
|   4    3 a  a
|   5    4 a  a
|   7    6 a  a
|   9    8 a  a
|   30   3 c  a
|   48   8 a  b
```

```
|   70   4 c  b
|   91  12 a  c
|   92   0 b  c
|   100  8 b  c
```

we don't need to compare g with g0 with the same levels and we only need comparisons in one direction (perhaps not)

```
dpred <- subset( dpred, g0 < g)
dim(dpred)
```

```
|   [1] 39  3
```

```
some(dpred)
```

```
|       x g g0
|   14  0 b  a
|   28  1 c  a
|   31  4 c  a
|   34  7 c  a
|   36  9 c  a
```

```
|   38 11 c  a
|   69  3 c  b
|   71  5 c  b
|   72  6 c  b
|   75  9 c  b
```

```
Lfx(sfit2)
```

```
|   list( 1,
|   1 * M(g),
|   1 * x,
|   1 * M(g) * x
|   )
```

'difference' g - g0, just like differentiating wrt to g except that 'M' function generates differences

```
Lmat <- Lfx( sfit2,
         list( 0,
               0 * x,
```

```
                1 * M(g,g0),
                1 * x * M(g,g0)
        ), dpred)
Lmat
```

	(Intercept)	gb	gc	x	gb:x	gc:x
14	0	0	1	0	0	0
15	0	0	1	0	1	0
16	0	0	1	0	2	0
17	0	0	1	0	3	0
18	0	0	1	0	4	0
19	0	0	1	0	5	0
20	0	0	1	0	6	0
21	0	0	1	0	7	0
22	0	0	1	0	8	0
23	0	0	1	0	9	0
24	0	0	1	0	10	0
25	0	0	1	0	11	0
26	0	0	1	0	12	0

27	0	0	0	1	0	0
28	0	0	0	1	0	1
29	0	0	0	1	0	2
30	0	0	0	1	0	3
31	0	0	0	1	0	4
32	0	0	0	1	0	5
33	0	0	0	1	0	6
34	0	0	0	1	0	7
35	0	0	0	1	0	8
36	0	0	0	1	0	9
37	0	0	0	1	0	10
38	0	0	0	1	0	11
39	0	0	0	1	0	12
66	0	0	-1	1	0	0
67	0	0	-1	1	-1	1
68	0	0	-1	1	-2	2
69	0	0	-1	1	-3	3
70	0	0	-1	1	-4	4
71	0	0	-1	1	-5	5

```
|   72           0  0 -1 1   -6    6
|   73           0  0 -1 1   -7    7
|   74           0  0 -1 1   -8    8
|   75           0  0 -1 1   -9    9
|   76           0  0 -1 1  -10   10
|   77           0  0 -1 1  -11   11
|   78           0  0 -1 1  -12   12
|   attr(,"data")
|       x g g0
|   14  0 b  a
|   15  1 b  a
|   16  2 b  a
|   17  3 b  a
|   18  4 b  a
|   19  5 b  a
|   20  6 b  a
|   21  7 b  a
|   22  8 b  a
|   23  9 b  a
```

```
|   24 10 b  a
|   25 11 b  a
|   26 12 b  a
|   27  0 c  a
|   28  1 c  a
|   29  2 c  a
|   30  3 c  a
|   31  4 c  a
|   32  5 c  a
|   33  6 c  a
|   34  7 c  a
|   35  8 c  a
|   36  9 c  a
|   37 10 c  a
|   38 11 c  a
|   39 12 c  a
|   66  0 c  b
|   67  1 c  b
|   68  2 c  b
```

```
|   69  3 c  b
|   70  4 c  b
|   71  5 c  b
|   72  6 c  b
|   73  7 c  b
|   74  8 c  b
|   75  9 c  b
|   76 10 c  b
|   77 11 c  b
|   78 12 c  b
```

```r
wald(sfit2, Lmat)
```

```
|     numDF denDF F.value p.value
|   1     4     6 63.0834   5e-05
|      Estimate  Std.Error DF t-value   p-value Lower 0.95 Upper 0.9
|   14 -0.536269  2.159667 6  -0.248311 0.81217  -5.820784  4.748246
|   15 -0.279621  1.759556 6  -0.158915 0.87895  -4.585100  4.025859
|   16 -0.022973  1.793506 6  -0.012809 0.99020  -4.411523  4.365578
```

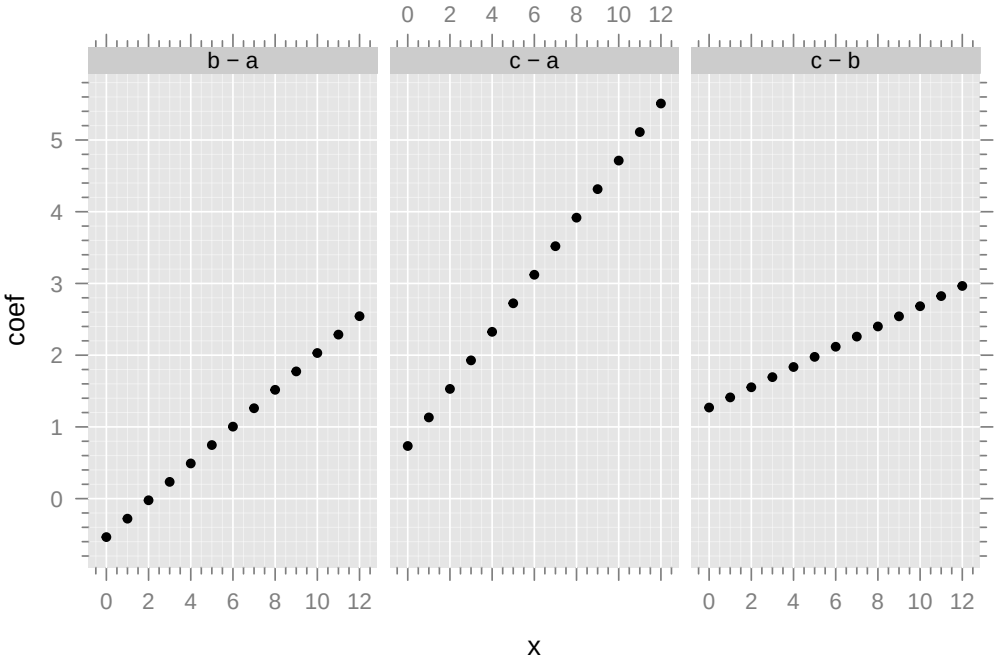
	17	0.233675	2.241882	6	0.104232	0.92038	-5.252013	5.719364
	18	0.490323	2.919616	6	0.167941	0.87215	-6.653720	7.634366
	19	0.746971	3.702841	6	0.201729	0.84679	-8.313553	9.807496
	20	1.003619	4.537251	6	0.221195	0.83228	-10.098634	12.105872
	21	1.260267	5.399168	6	0.233419	0.82320	-11.951020	14.471555
	22	1.516915	6.277271	6	0.241652	0.81710	-13.843013	16.876843
	23	1.773563	7.165612	6	0.247510	0.81277	-15.760057	19.307183
	24	2.030211	8.060806	6	0.251862	0.80955	-17.693872	21.754294
	25	2.286859	8.960801	6	0.255207	0.80709	-19.639432	24.213150
	26	2.543507	9.864282	6	0.257850	0.80514	-21.593523	26.680537
	27	0.733744	0.896753	6	0.818223	0.44450	-1.460531	2.928020
	28	1.131654	0.200520	6	5.643597	0.00133	0.641000	1.622309
	29	1.529565	0.982344	6	1.557057	0.17046	-0.874144	3.933273
	30	1.927475	1.891702	6	1.018910	0.34756	-2.701354	6.556304
	31	2.325385	2.807281	6	0.828341	0.43918	-4.543783	9.194554
	32	2.723296	3.724495	6	0.731185	0.49222	-6.390215	11.836806
	33	3.121206	4.642375	6	0.672330	0.52640	-8.238276	14.480688
	34	3.519116	5.560591	6	0.632867	0.55016	-10.087161	17.125393
	35	3.917027	6.479001	6	0.604573	0.56761	-11.936518	19.770572

36	4.314937	7.397532	6	0.583294	0.58094	-13.786173	22.416047
37	4.712847	8.316145	6	0.566711	0.59146	-15.636026	25.061720
38	5.110757	9.234814	6	0.553423	0.59997	-17.486018	27.707533
39	5.508668	10.153525	6	0.542537	0.60700	-19.336112	30.353448
66	1.270013	1.748093	6	0.726513	0.49488	-3.007418	5.547443
67	1.411275	1.944114	6	0.725922	0.49522	-3.345801	6.168351
68	1.552537	2.159667	6	0.718878	0.49924	-3.731978	6.837052
69	1.693800	2.389472	6	0.708859	0.50501	-4.153028	7.540628
70	1.835062	2.629796	6	0.697796	0.51143	-4.599817	8.269941
71	1.976324	2.878004	6	0.686700	0.51792	-5.065898	9.018547
72	2.117587	3.132223	6	0.676065	0.52419	-5.546688	9.781861
73	2.258849	3.391102	6	0.666111	0.53010	-6.038878	10.556576
74	2.400111	3.653649	6	0.656908	0.53561	-6.540046	11.340269
75	2.541374	3.919128	6	0.648454	0.54070	-7.048388	12.131136
76	2.682636	4.186982	6	0.640709	0.54539	-7.562539	12.927811
77	2.823898	4.456781	6	0.633618	0.54971	-8.081452	13.729249
78	2.965161	4.728193	6	0.627123	0.55368	-8.604311	14.534633

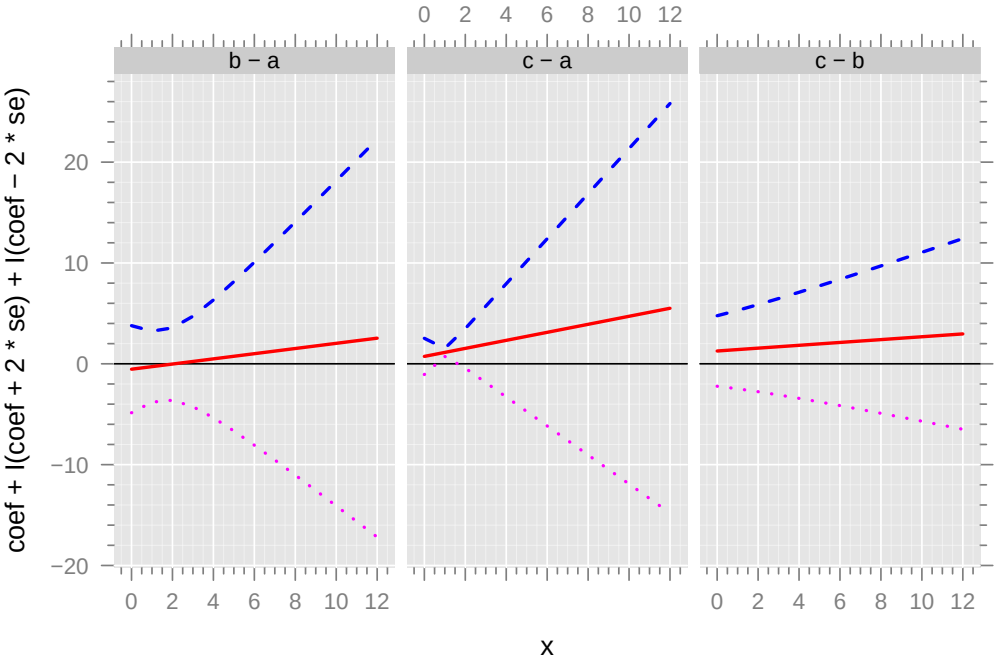
```r
ww <- as.data.frame(wald(sfit2, Lmat))
head(ww)
```

	coef	se	U2	L2	x	g	g0
14	-0.5362685	2.159667	3.783066	-4.855603	0	b	a
15	-0.2796206	1.759556	3.239492	-3.798733	1	b	a
16	-0.0229726	1.793506	3.564039	-3.609984	2	b	a
17	0.2336754	2.241882	4.717440	-4.250089	3	b	a
18	0.4903233	2.919616	6.329555	-5.348909	4	b	a
19	0.7469713	3.702841	8.152652	-6.658710	5	b	a

```r
ww$gap <- with(ww, paste( g, '-', g0))
xyplot( coef ~ x | gap, ww)
```

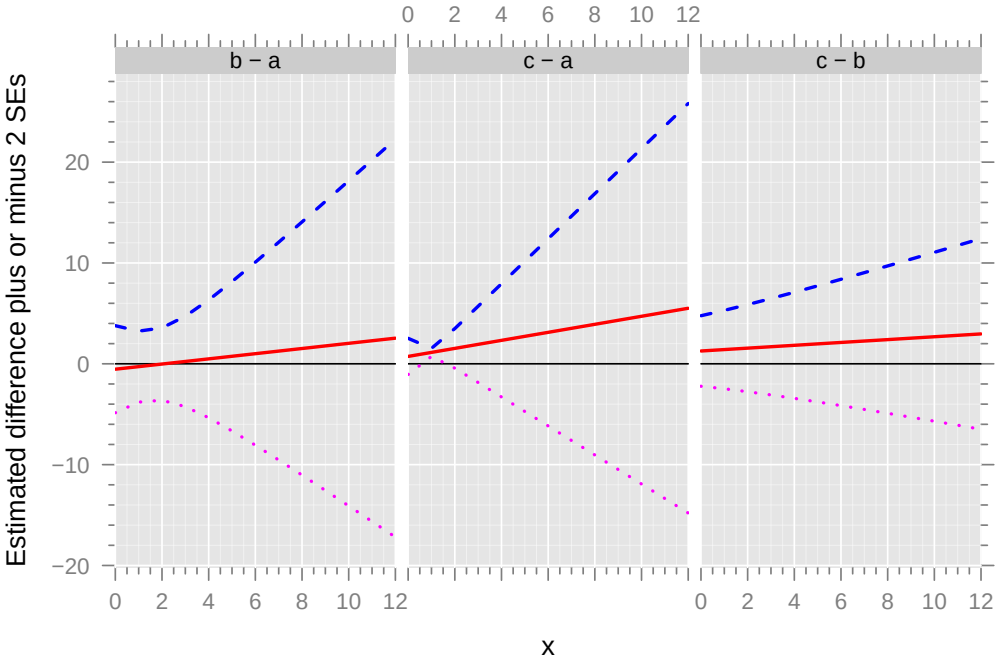

b − a
c − a
c − b
coef
x
0
1
2
3
4
5
0
2
4
6
8
10
12

```
td( col = c('black', 'blue','blue'), lty = 1, lwd = 2)
xyplot( coef +I(coef+2*se) + I(coef-2*se) ~ x | gap,
        ww, type = 'l') +
layer_( panel.abline( h = 0))
```


b − a
c − a
c − b
coef + I(coef + 2 * se) + I(coef − 2 * se)
x
20
10
0
−10
−20
0
2
4
6
8
10
12

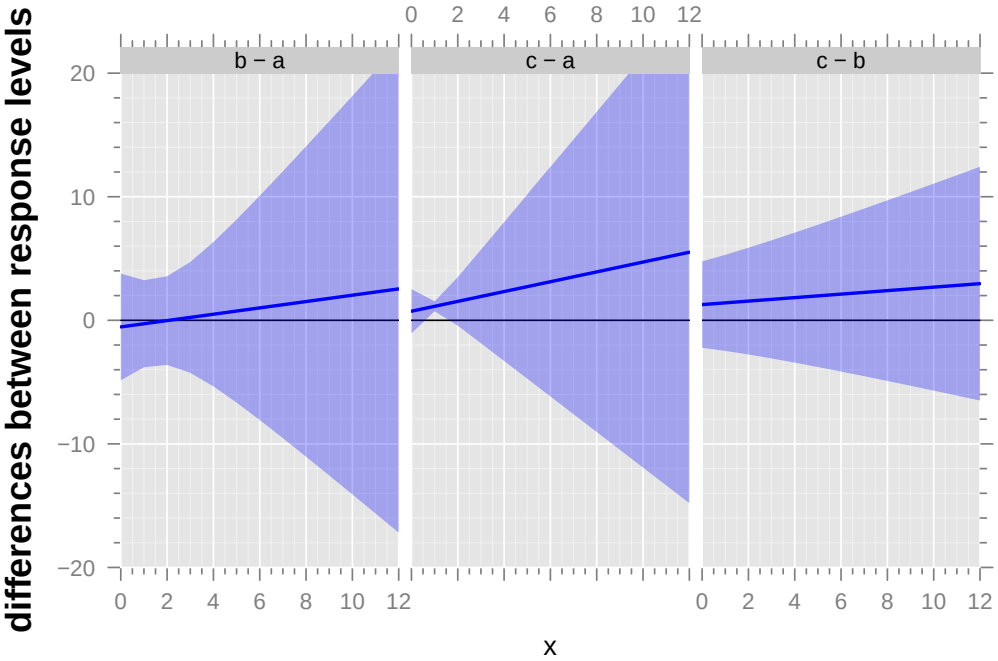
or

```
xyplot( coef +I(coef+2*se) + I(coef-2*se) ~ x | gap,
        ww, type = 'l',
        ylab = "Estimated difference plus or minus 2 SEs",
        xlim = c(0,12)) +
  layer_( panel.abline( h = 0))
```


b − a
c − a
c − b
Estimated difference plus or minus 2 SEs
x
0
2
4
6
8
10
12
20
10
0
−10
−20

or

```
gd(col = 'blue')
xyplot( coef  ~ x | gap, ww, type = 'l',
        ylab = list("differences between response levels", cex = 1.3
        xlim = c(0,12),
        ylim = c(-20,20),
        upper = ww$coef + 2* ww$se,
        lower = ww$coef - 2* ww$se,
        subscripts = T) +
  layer(panel.fit(...)) +
  layer_( panel.abline( h = 0))
```


differences between response levels
b − a
c − a
c − b
20
10
0
−10
−20
0
2
4
6
8
10
12
x

Note that if the significant gap between 'c' and 'b' around $x = 6$ is a question inspired by the data and not a 'prior' hypothesis, then some adjustment should be made for **multiplicity**.

3.7.1 Exercises

1. Explore how to modify the appearance of the 'strips', i.e. where it says 'c - b'
2. Plot approximate '95% confidence bands' for each group with +/- 2 SEs
3. Plot approximate '95% prediction bands' for each group.

reset the random seed:

```
set.seed(NULL)
```

3.8 Using WHO regions as predictors of Life Expectancy

```r
fitr <- lm( LifeExp ~
              (smoke + I(smoke^2)) * region + hiv + special, dd)
summary(fitr)
```

```
|
|   Call:
|   lm(formula = LifeExp ~ (smoke + I(smoke^2)) * region + hiv +
|       special, data = dd)
|
|   Residuals:
|        Min       1Q   Median       3Q      Max
|   -10.3209  -3.1528   0.1424   3.0536  10.7776
|
|   Coefficients:
|                            Estimate Std. Error t value Pr(>|t|)
|   (Intercept)             5.645e+01  1.512e+00  37.324  < 2e-16 ***
|   smoke                   9.687e-03  1.343e-02   0.721 0.472152
|   I(smoke^2)              1.503e-05  1.811e-05   0.830 0.408190
```

```
|   regionAMR                1.079e+01  2.846e+00   3.792 0.000226 ***
|   regionEMR               -1.088e+00  3.106e+00  -0.350 0.726834
|   regionEUR                2.211e+01  3.982e+00   5.551  1.5e-07 ***
|   regionSEAR               1.146e+01  3.977e+00   2.881 0.004634 **
|   regionWPR                8.297e+00  7.772e+00   1.068 0.287671
|   hiv                     -2.405e-01  1.112e-01  -2.162 0.032401 *
|   special                 -1.006e+01  2.958e+00  -3.402 0.000885 ***
|   smoke:regionAMR          1.876e-02  1.693e-02   1.108 0.269968
|   smoke:regionEMR          1.857e-02  1.532e-02   1.212 0.227560
|   smoke:regionEUR         -9.910e-03  1.430e-02  -0.693 0.489629
|   smoke:regionSEAR         2.485e-03  2.294e-02   0.108 0.913918
|   smoke:regionWPR          4.084e-03  2.122e-02   0.192 0.847700
|   I(smoke^2):regionAMR    -3.295e-05  1.988e-05  -1.658 0.099715 .
|   I(smoke^2):regionEMR    -2.432e-05  1.844e-05  -1.319 0.189551
|   I(smoke^2):regionEUR    -1.559e-05  1.817e-05  -0.858 0.392392
|   I(smoke^2):regionSEAR   -2.559e-05  2.434e-05  -1.051 0.295049
|   I(smoke^2):regionWPR    -1.788e-05  1.938e-05  -0.923 0.357846
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
|
|   Residual standard error: 4.556 on 132 degrees of freedom
|     (42 observations deleted due to missingness)
|   Multiple R-squared:  0.8045,    Adjusted R-squared:  0.7764
|   F-statistic:  28.6 on 19 and 132 DF,  p-value: < 2.2e-16
```

```
wald(fitr,":")
```

```
|     numDF denDF  F.value p.value
|   :    10   132 3.198898 0.00104
|                          Estimate  Std.Error DF  t-value   p-value
|   smoke:regionAMR         0.018757 0.016932  132  1.107787 0.26997
|   smoke:regionEMR         0.018572 0.015320  132  1.212307 0.22756
|   smoke:regionEUR        -0.009910 0.014304  132 -0.692837 0.48963
|   smoke:regionSEAR        0.002485 0.022944  132  0.108305 0.91392
|   smoke:regionWPR         0.004084 0.021225  132  0.192431 0.84770
|   I(smoke^2):regionAMR   -0.000033 0.000020  132 -1.657885 0.09972
|   I(smoke^2):regionEMR   -0.000024 0.000018  132 -1.318706 0.18955
|   I(smoke^2):regionEUR   -0.000016 0.000018  132 -0.858101 0.39239
```

```
|   I(smoke^2):regionSEAR -0.000026 0.000024  132 -1.051283 0.29505
|   I(smoke^2):regionWPR  -0.000018 0.000019  132 -0.922703 0.35785
|                         Upper 0.95
|   smoke:regionAMR       0.052251
|   smoke:regionEMR       0.048876
|   smoke:regionEUR       0.018384
|   smoke:regionSEAR      0.047871
|   smoke:regionWPR       0.046069
|   I(smoke^2):regionAMR  0.000006
|   I(smoke^2):regionEMR  0.000012
|   I(smoke^2):regionEUR  0.000020
|   I(smoke^2):regionSEAR 0.000023
|   I(smoke^2):regionWPR  0.000020
```

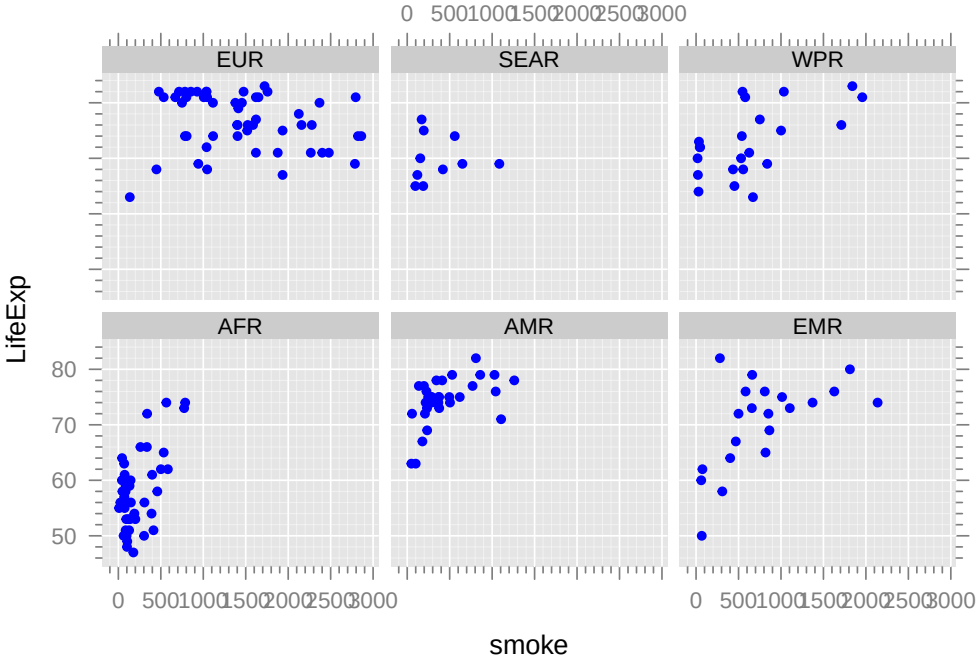
```
wald(fitr,"2):")
```

```
|       numDF denDF  F.value p.value
|   2):     5   132 2.093536 0.07012
|                         Estimate Std.Error DF  t-value   p-value L
```

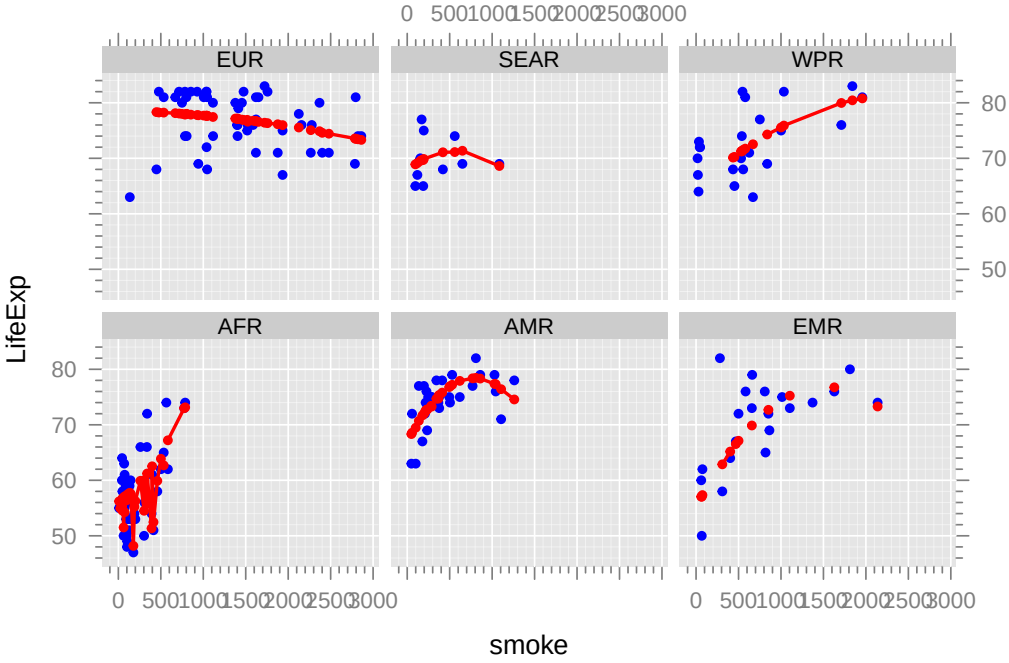
```
|   I(smoke^2):regionAMR  -3.3e-05 2.0e-05   132 -1.657885 0.09972 -
|   I(smoke^2):regionEMR  -2.4e-05 1.8e-05   132 -1.318706 0.18955 -
|   I(smoke^2):regionEUR  -1.6e-05 1.8e-05   132 -0.858101 0.39239 -
|   I(smoke^2):regionSEAR -2.6e-05 2.4e-05   132 -1.051283 0.29505 -
|   I(smoke^2):regionWPR  -1.8e-05 1.9e-05   132 -0.922703 0.35785 -
|                          Upper 0.95
|   I(smoke^2):regionAMR  6.0e-06
|   I(smoke^2):regionEMR  1.2e-05
|   I(smoke^2):regionEUR  2.0e-05
|   I(smoke^2):regionSEAR 2.3e-05
|   I(smoke^2):regionWPR  2.0e-05
```

Using data values for prediction instead of creating a separate prediction data frame: This can work with curvilinear models if the data is sufficiently dense.

```
dd$yq <- predict( fitr, dd)
xyplot( LifeExp ~ smoke | region, dd)
```


EUR
SEAR
WPR
AFR
AMR
EMR
LifeExp
smoke
0
500
1000
1500
2000
2500
3000
50
60
70
80

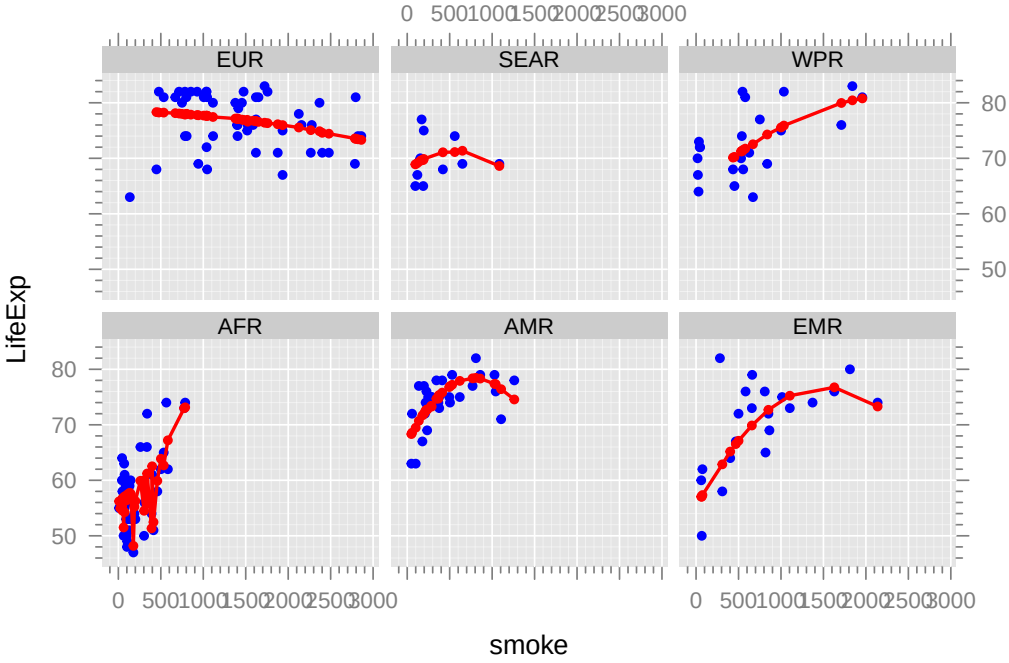
```r
# reorder for nice lines:
dd <- dd[order(dd$region,dd$smoke),]
xyplot( LifeExp ~ smoke | region,
        dd, subscripts = T, yq = dd$yq)  +
  layer( panel.xyplot( x, yq[subscripts],...,
                       type = 'b', col = 'red'))
```


EUR
SEAR
WPR
AFR
AMR
EMR
LifeExp
smoke
0
500
1000
1500
2000
2500
3000
50
60
70
80

presents a problem because of missing hiv values.

Try again keeping non-missing data together to avoid interrupting lines

```
dd <- dd[order(is.na(dd$yq),dd$region,dd$smoke),]
xyplot( LifeExp ~ smoke | region,
        dd, subscripts = T, yq = dd$yq)  +
  layer( panel.xyplot( x, yq[subscripts],...,
                       type = 'b', col = 'red'))
```


EUR
SEAR
WPR
AFR
AMR
EMR
LifeExp
smoke
0
500
1000
1500
2000
2500
3000
50
60
70
80

Including Health Expenditure:

```
fitrhe <- lm(
  LifeExp ~
    (smoke + I(smoke^2)) * region * log(HE) + hiv + special,
  dd)
summary(fitrhe)
```

```
|
|   Call:
|   lm(formula = LifeExp ~ (smoke + I(smoke^2)) * region * log(HE) +
|       hiv + special, data = dd)
|
|   Residuals:
|       Min      1Q  Median      3Q     Max
|   -6.9890 -1.4067  0.1123  1.2544  9.7261
|
|   Coefficients:
|                                    Estimate Std. Error t value Pr(>
```

	(Intercept)	5.088e+01	7.646e+00	6.654	1.12
	smoke	-1.832e-02	5.188e-02	-0.353	0.72
	I(smoke^2)	4.173e-05	7.885e-05	0.529	0.59
	regionAMR	-1.375e+01	1.544e+01	-0.890	0.37
	regionEMR	6.606e+00	1.782e+01	0.371	0.71
	regionEUR	-8.512e+00	1.948e+01	-0.437	0.66
	regionSEAR	1.698e+02	8.098e+01	2.097	0.03
	regionWPR	-6.176e+00	2.585e+01	-0.239	0.81
	log(HE)	1.527e+00	1.679e+00	0.910	0.36
	hiv	-3.930e-01	1.003e-01	-3.917	0.00
	special	-1.296e+01	2.343e+00	-5.532	2.13
	smoke:regionAMR	6.425e-02	8.185e-02	0.785	0.43
	smoke:regionEMR	2.158e-02	6.486e-02	0.333	0.74
	smoke:regionEUR	2.975e-02	5.868e-02	0.507	0.61
	smoke:regionSEAR	-1.262e+00	6.146e-01	-2.053	0.04
	smoke:regionWPR	1.699e-02	7.631e-02	0.223	0.82
	I(smoke^2):regionAMR	-5.165e-05	9.443e-05	-0.547	0.58
	I(smoke^2):regionEMR	-4.252e-05	8.567e-05	-0.496	0.62
	I(smoke^2):regionEUR	-4.762e-05	7.938e-05	-0.600	0.54

	I(smoke^2):regionSEAR	1.839e-03	8.954e-04	2.054	0.04
	I(smoke^2):regionWPR	-3.620e-05	8.313e-05	-0.435	0.66
	smoke:log(HE)	5.178e-03	9.762e-03	0.530	0.59
	I(smoke^2):log(HE)	-6.228e-06	1.370e-05	-0.454	0.65
	regionAMR:log(HE)	3.852e+00	2.780e+00	1.386	0.16
	regionEMR:log(HE)	-1.179e+00	3.844e+00	-0.307	0.75
	regionEUR:log(HE)	3.195e+00	2.932e+00	1.090	0.27
	regionSEAR:log(HE)	-3.376e+01	1.688e+01	-2.000	0.04
	regionWPR:log(HE)	2.831e+00	4.291e+00	0.660	0.51
	smoke:regionAMR:log(HE)	-1.044e-02	1.392e-02	-0.750	0.45
	smoke:regionEMR:log(HE)	-2.242e-03	1.204e-02	-0.186	0.85
	smoke:regionEUR:log(HE)	-6.670e-03	1.044e-02	-0.639	0.52
	smoke:regionSEAR:log(HE)	2.685e-01	1.292e-01	2.077	0.04
	smoke:regionWPR:log(HE)	-4.551e-03	1.300e-02	-0.350	0.72
	I(smoke^2):regionAMR:log(HE)	6.664e-06	1.592e-05	0.419	0.67
	I(smoke^2):regionEMR:log(HE)	5.200e-06	1.448e-05	0.359	0.72
	I(smoke^2):regionEUR:log(HE)	6.979e-06	1.376e-05	0.507	0.61
	I(smoke^2):regionSEAR:log(HE)	-3.900e-04	1.869e-04	-2.087	0.03
	I(smoke^2):regionWPR:log(HE)	5.384e-06	1.424e-05	0.378	0.70

```
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
|
|   Residual standard error: 3.194 on 111 degrees of freedom
|     (45 observations deleted due to missingness)
|   Multiple R-squared:  0.9154,    Adjusted R-squared:  0.8873
|   F-statistic: 32.48 on 37 and 111 DF,  p-value: < 2.2e-16
```

```
length(coef(fitrhe))
```

```
|   [1] 38
```

Should have > 380 observations using Harrell's rules of thumb for valid regression

```
wald(fitrhe, ":")
```

```
|     numDF denDF F.value p.value
|   :    27   111 1.37843 0.12542
|                                  Estimate   Std.Error DF  t-value
|   smoke:regionAMR                  0.064254  0.081849 111  0.785035
```

	smoke:regionEMR	0.021575	0.064860	111	0.332641
	smoke:regionEUR	0.029747	0.058678	111	0.506947
	smoke:regionSEAR	-1.261784	0.614556	111	-2.053164
	smoke:regionWPR	0.016993	0.076313	111	0.222673
	I(smoke^2):regionAMR	-0.000052	0.000094	111	-0.546953
	I(smoke^2):regionEMR	-0.000043	0.000086	111	-0.496344
	I(smoke^2):regionEUR	-0.000048	0.000079	111	-0.599894
	I(smoke^2):regionSEAR	0.001839	0.000895	111	2.054470
	I(smoke^2):regionWPR	-0.000036	0.000083	111	-0.435470
	smoke:log(HE)	0.005178	0.009762	111	0.530439
	I(smoke^2):log(HE)	-0.000006	0.000014	111	-0.454460
	regionAMR:log(HE)	3.851782	2.780032	111	1.385517
	regionEMR:log(HE)	-1.179181	3.844186	111	-0.306744
	regionEUR:log(HE)	3.194526	2.932055	111	1.089518
	regionSEAR:log(HE)	-33.758269	16.879974	111	-1.999901
	regionWPR:log(HE)	2.830507	4.290773	111	0.659673
	smoke:regionAMR:log(HE)	-0.010442	0.013922	111	-0.750002
	smoke:regionEMR:log(HE)	-0.002242	0.012041	111	-0.186234
	smoke:regionEUR:log(HE)	-0.006670	0.010440	111	-0.638910

	smoke:regionSEAR:log(HE)	0.268460	0.129242	111	2.077190
	smoke:regionWPR:log(HE)	-0.004551	0.012996	111	-0.350166
	I(smoke^2):regionAMR:log(HE)	0.000007	0.000016	111	0.418627
	I(smoke^2):regionEMR:log(HE)	0.000005	0.000014	111	0.359028
	I(smoke^2):regionEUR:log(HE)	0.000007	0.000014	111	0.507212
	I(smoke^2):regionSEAR:log(HE)	-0.000390	0.000187	111	-2.087196
	I(smoke^2):regionWPR:log(HE)	0.000005	0.000014	111	0.378247

		Lower 0.95	Upper 0.95
	smoke:regionAMR	-0.097934	0.226443
	smoke:regionEMR	-0.106950	0.150100
	smoke:regionEUR	-0.086528	0.146022
	smoke:regionSEAR	-2.479567	-0.044001
	smoke:regionWPR	-0.134226	0.168211
	I(smoke^2):regionAMR	-0.000239	0.000135
	I(smoke^2):regionEMR	-0.000212	0.000127
	I(smoke^2):regionEUR	-0.000205	0.000110
	I(smoke^2):regionSEAR	0.000065	0.003614
	I(smoke^2):regionWPR	-0.000201	0.000129
	smoke:log(HE)	-0.014165	0.024521

```
|   I(smoke^2):log(HE)              -0.000033  0.000021
|   regionAMR:log(HE)               -1.657038  9.360602
|   regionEMR:log(HE)               -8.796691  6.438330
|   regionEUR:log(HE)               -2.615536  9.004588
|   regionSEAR:log(HE)             -67.207064 -0.309474
|   regionWPR:log(HE)               -5.671945 11.332960
|   smoke:regionAMR:log(HE)         -0.038029  0.017146
|   smoke:regionEMR:log(HE)         -0.026102  0.021617
|   smoke:regionEUR:log(HE)         -0.027359  0.014018
|   smoke:regionSEAR:log(HE)         0.012358  0.524561
|   smoke:regionWPR:log(HE)         -0.030303  0.021202
|   I(smoke^2):regionAMR:log(HE)    -0.000025  0.000038
|   I(smoke^2):regionEMR:log(HE)    -0.000024  0.000034
|   I(smoke^2):regionEUR:log(HE)    -0.000020  0.000034
|   I(smoke^2):regionSEAR:log(HE)   -0.000760 -0.000020
|   I(smoke^2):regionWPR:log(HE)    -0.000023  0.000034
```

```
wald(fitrhe, "2")
```

```
|     numDF denDF   F.value p.value
|   2    12   111 0.7935878 0.65615
|                                    Estimate  Std.Error DF  t-value
|   I(smoke^2)                        0.000042 0.000079  111  0.529253
|   I(smoke^2):regionAMR             -0.000052 0.000094  111 -0.546953
|   I(smoke^2):regionEMR             -0.000043 0.000086  111 -0.496344
|   I(smoke^2):regionEUR             -0.000048 0.000079  111 -0.599894
|   I(smoke^2):regionSEAR             0.001839 0.000895  111  2.054470
|   I(smoke^2):regionWPR             -0.000036 0.000083  111 -0.435470
|   I(smoke^2):log(HE)               -0.000006 0.000014  111 -0.454460
|   I(smoke^2):regionAMR:log(HE)      0.000007 0.000016  111  0.418627
|   I(smoke^2):regionEMR:log(HE)      0.000005 0.000014  111  0.359028
|   I(smoke^2):regionEUR:log(HE)      0.000007 0.000014  111  0.507212
|   I(smoke^2):regionSEAR:log(HE) -0.000390 0.000187  111 -2.087196
|   I(smoke^2):regionWPR:log(HE)      0.000005 0.000014  111  0.378247
|                                    Lower 0.95 Upper 0.95
|   I(smoke^2)                       -0.000115   0.000198
|   I(smoke^2):regionAMR             -0.000239   0.000135
|   I(smoke^2):regionEMR             -0.000212   0.000127
```

```
|   I(smoke^2):regionEUR           -0.000205   0.000110
|   I(smoke^2):regionSEAR           0.000065   0.003614
|   I(smoke^2):regionWPR           -0.000201   0.000129
|   I(smoke^2):log(HE)             -0.000033   0.000021
|   I(smoke^2):regionAMR:log(HE)   -0.000025   0.000038
|   I(smoke^2):regionEMR:log(HE)   -0.000024   0.000034
|   I(smoke^2):regionEUR:log(HE)   -0.000020   0.000034
|   I(smoke^2):regionSEAR:log(HE) -0.000760  -0.000020
|   I(smoke^2):regionWPR:log(HE)   -0.000023   0.000034
```

```
wald(fitrhe, "2|:.*:") # quadratic terms and 3 and
```

```
|          numDF denDF  F.value p.value
|   2|:.*:    17   111 1.286303 0.21443
|                                Estimate  Std.Error DF  t-value
|   I(smoke^2)                    0.000042 0.000079  111  0.529253
|   I(smoke^2):regionAMR          -0.000052 0.000094  111 -0.546953
|   I(smoke^2):regionEMR          -0.000043 0.000086  111 -0.496344
|   I(smoke^2):regionEUR          -0.000048 0.000079  111 -0.599894
```

```
|   I(smoke^2):regionSEAR          0.001839 0.000895  111  2.054470
|   I(smoke^2):regionWPR          -0.000036 0.000083  111 -0.435470
|   I(smoke^2):log(HE)            -0.000006 0.000014  111 -0.454460
|   smoke:regionAMR:log(HE)       -0.010442 0.013922  111 -0.750002
|   smoke:regionEMR:log(HE)       -0.002242 0.012041  111 -0.186234
|   smoke:regionEUR:log(HE)       -0.006670 0.010440  111 -0.638910
|   smoke:regionSEAR:log(HE)       0.268460 0.129242  111  2.077190
|   smoke:regionWPR:log(HE)       -0.004551 0.012996  111 -0.350166
|   I(smoke^2):regionAMR:log(HE)   0.000007 0.000016  111  0.418627
|   I(smoke^2):regionEMR:log(HE)   0.000005 0.000014  111  0.359028
|   I(smoke^2):regionEUR:log(HE)   0.000007 0.000014  111  0.507212
|   I(smoke^2):regionSEAR:log(HE) -0.000390 0.000187  111 -2.087196
|   I(smoke^2):regionWPR:log(HE)   0.000005 0.000014  111  0.378247
|                                 Lower 0.95 Upper 0.95
|   I(smoke^2)                    -0.000115   0.000198
|   I(smoke^2):regionAMR          -0.000239   0.000135
|   I(smoke^2):regionEMR          -0.000212   0.000127
|   I(smoke^2):regionEUR          -0.000205   0.000110
|   I(smoke^2):regionSEAR          0.000065   0.003614
```

```
|   I(smoke^2):regionWPR           -0.000201   0.000129
|   I(smoke^2):log(HE)             -0.000033   0.000021
|   smoke:regionAMR:log(HE)        -0.038029   0.017146
|   smoke:regionEMR:log(HE)        -0.026102   0.021617
|   smoke:regionEUR:log(HE)        -0.027359   0.014018
|   smoke:regionSEAR:log(HE)        0.012358   0.524561
|   smoke:regionWPR:log(HE)        -0.030303   0.021202
|   I(smoke^2):regionAMR:log(HE)   -0.000025   0.000038
|   I(smoke^2):regionEMR:log(HE)   -0.000024   0.000034
|   I(smoke^2):regionEUR:log(HE)   -0.000020   0.000034
|   I(smoke^2):regionSEAR:log(HE) -0.000760  -0.000020
|   I(smoke^2):regionWPR:log(HE)   -0.000023   0.000034
```

```r
                    # higher way interaction
fitr2 <- lm( LifeExp ~
               (smoke + log(HE) + region)^2 + hiv + special, dd)
summary(fitr2)
```

```
|
```

```
Call:
lm(formula = LifeExp ~ (smoke + log(HE) + region)^2 + hiv + spec
    data = dd)

Residuals:
    Min      1Q  Median      3Q     Max
-6.9873 -1.8511  0.1576  1.7496  8.8413

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)         4.534e+01  3.590e+00  12.628  < 2e-16 ***
smoke               1.726e-02  4.899e-03   3.522 0.000594 ***
log(HE)             2.623e+00  8.903e-01   2.947 0.003816 **
regionAMR           6.004e+00  6.061e+00   0.991 0.323748
regionEMR           3.025e+00  8.003e+00   0.378 0.706085
regionEUR          -4.070e+00  7.195e+00  -0.566 0.572584
regionSEAR          7.659e+00  6.956e+00   1.101 0.272966
regionWPR          -2.520e+00  5.886e+00  -0.428 0.669332
hiv                -4.235e-01  9.424e-02  -4.494 1.55e-05 ***
```

```
|   special             -1.352e+01  2.183e+00  -6.195 7.35e-09 ***
|   smoke:log(HE)       -6.207e-04  6.668e-04  -0.931 0.353725
|   smoke:regionAMR     -1.142e-02  4.176e-03  -2.735 0.007128 **
|   smoke:regionEMR     -8.776e-03  4.180e-03  -2.099 0.037745 *
|   smoke:regionEUR     -1.372e-02  3.758e-03  -3.651 0.000379 ***
|   smoke:regionSEAR    -1.321e-02  4.980e-03  -2.652 0.009019 **
|   smoke:regionWPR     -1.100e-02  4.111e-03  -2.675 0.008450 **
|   log(HE):regionAMR    8.114e-01  1.180e+00   0.687 0.493061
|   log(HE):regionEMR    3.281e-01  1.700e+00   0.193 0.847248
|   log(HE):regionEUR    2.389e+00  1.156e+00   2.067 0.040730 *
|   log(HE):regionSEAR   7.592e-01  1.455e+00   0.522 0.602617
|   log(HE):regionWPR    2.073e+00  1.159e+00   1.789 0.075940 .
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
|
|   Residual standard error: 3.254 on 128 degrees of freedom
|     (45 observations deleted due to missingness)
|   Multiple R-squared:  0.8988,    Adjusted R-squared:  0.883
|   F-statistic: 56.83 on 20 and 128 DF,  p-value: < 2.2e-16
```

```r
wald(fitr2, ':')
```

```
|      numDF denDF  F.value p.value
|   :    11   128 2.478905 0.00752
|                       Estimate  Std.Error DF  t-value   p-value Low
|   smoke:log(HE)       -0.000621 0.000667  128 -0.930769 0.35373 -0.
|   smoke:regionAMR     -0.011419 0.004176  128 -2.734757 0.00713 -0.
|   smoke:regionEMR     -0.008776 0.004180  128 -2.099411 0.03774 -0.
|   smoke:regionEUR     -0.013722 0.003758  128 -3.651365 0.00038 -0.
|   smoke:regionSEAR    -0.013207 0.004980  128 -2.651798 0.00902 -0.
|   smoke:regionWPR     -0.010997 0.004111  128 -2.674936 0.00845 -0.
|   log(HE):regionAMR    0.811401 1.180355  128  0.687421 0.49306 -1.
|   log(HE):regionEMR    0.328123 1.699935  128  0.193021 0.84725 -3.
|   log(HE):regionEUR    2.389164 1.155748  128  2.067201 0.04073  0.
|   log(HE):regionSEAR   0.759205 1.454596  128  0.521935 0.60262 -2.
|   log(HE):regionWPR    2.073361 1.158788  128  1.789249 0.07594 -0.
|                       Upper 0.95
|   smoke:log(HE)        0.000699
```

```
|   smoke:regionAMR    -0.003157
|   smoke:regionEMR    -0.000505
|   smoke:regionEUR    -0.006286
|   smoke:regionSEAR   -0.003352
|   smoke:regionWPR    -0.002862
|   log(HE):regionAMR   3.146934
|   log(HE):regionEMR   3.691734
|   log(HE):regionEUR   4.676008
|   log(HE):regionSEAR  3.637371
|   log(HE):regionWPR   4.366221
```

```
wald(fitr2, 'HE):|:log')
```

```
|              numDF denDF   F.value p.value
|   HE):|:log     6   128 0.9991023 0.42891
|                      Estimate  Std.Error DF  t-value   p-value Low
|   smoke:log(HE)      -0.000621 0.000667  128 -0.930769 0.35373 -0.
|   log(HE):regionAMR   0.811401 1.180355  128  0.687421 0.49306 -1.
|   log(HE):regionEMR   0.328123 1.699935  128  0.193021 0.84725 -3.
```

```
|   log(HE):regionEUR   2.389164 1.155748  128  2.067201 0.04073  0.
|   log(HE):regionSEAR  0.759205 1.454596  128  0.521935 0.60262 -2.
|   log(HE):regionWPR   2.073361 1.158788  128  1.789249 0.07594 -0.
|                      Upper 0.95
|   smoke:log(HE)      0.000699
|   log(HE):regionAMR  3.146934
|   log(HE):regionEMR  3.691734
|   log(HE):regionEUR  4.676008
|   log(HE):regionSEAR 3.637371
|   log(HE):regionWPR  4.366221
```

```
fitr3 <- lm( LifeExp ~
               region* smoke + log(HE)+ hiv + special, dd)
summary(fitr3)
```

```
|
|   Call:
|   lm(formula = LifeExp ~ region * smoke + log(HE) + hiv + special,
|       data = dd)
```

```
|
|   Residuals:
|      Min      1Q  Median      3Q     Max
|   -7.016 -1.863   0.165   1.627   8.700
|
|   Coefficients:
|                       Estimate Std. Error t value Pr(>|t|)
|   (Intercept)        41.364150   1.472822  28.085  < 2e-16 ***
|   regionAMR           8.712382   1.438295   6.057 1.31e-08 ***
|   regionEMR           3.799335   1.759763   2.159 0.032631 *
|   regionEUR           9.842506   1.887832   5.214 6.84e-07 ***
|   regionSEAR         10.122815   1.839269   5.504 1.83e-07 ***
|   regionWPR           7.460923   1.969657   3.788 0.000229 ***
|   smoke               0.010666   0.002712   3.933 0.000134 ***
|   log(HE)             3.688028   0.314332  11.733  < 2e-16 ***
|   hiv                -0.501073   0.079030  -6.340 3.25e-09 ***
|   special           -14.766685   2.013362  -7.334 1.91e-11 ***
|   regionAMR:smoke    -0.009904   0.003175  -3.120 0.002216 **
|   regionEMR:smoke    -0.007235   0.003068  -2.358 0.019822 *
```

```
|   regionEUR:smoke   -0.011760   0.002835  -4.148 5.93e-05 ***
|   regionSEAR:smoke  -0.009663   0.004494  -2.150 0.033335 *
|   regionWPR:smoke   -0.008164   0.003110  -2.625 0.009666 **
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
|
|   Residual standard error: 3.254 on 134 degrees of freedom
|     (45 observations deleted due to missingness)
|   Multiple R-squared:  0.894, Adjusted R-squared:  0.883
|   F-statistic: 80.76 on 14 and 134 DF,  p-value: < 2.2e-16
```

```
wald(fitr3, ":")
```

```
|     numDF denDF F.value p.value
|   :     5   134 4.25484 0.00127
|                     Estimate  Std.Error DF  t-value   p-value Lower
|   regionAMR:smoke  -0.009904 0.003175  134 -3.119787 0.00222 -0.01
|   regionEMR:smoke  -0.007235 0.003068  134 -2.357952 0.01982 -0.01
|   regionEUR:smoke  -0.011760 0.002835  134 -4.147578 0.00006 -0.01
```

regionSEAR:smoke	-0.009663	0.004494	134	-2.150191	0.03334	-0.01
regionWPR:smoke	-0.008164	0.003110	134	-2.625207	0.00967	-0.01

	Upper 0.95
regionAMR:smoke	-0.003625
regionEMR:smoke	-0.001166
regionEUR:smoke	-0.006152
regionSEAR:smoke	-0.000775
regionWPR:smoke	-0.002013

```
wald(fitr3, 'region')
```

	numDF	denDF	F.value	p.value
region	10	134	8.121058	<.00001

	Estimate	Std.Error	DF	t-value	p-value	Lower
regionAMR	8.712382	1.438295	134	6.057439	<.00001	5.86
regionEMR	3.799335	1.759763	134	2.159003	0.03263	0.31
regionEUR	9.842506	1.887832	134	5.213655	<.00001	6.10
regionSEAR	10.122815	1.839269	134	5.503717	<.00001	6.48
regionWPR	7.460923	1.969657	134	3.787931	0.00023	3.56

```
|   regionAMR:smoke  -0.009904 0.003175  134 -3.119787 0.00222 -0.01
|   regionEMR:smoke  -0.007235 0.003068  134 -2.357952 0.01982 -0.01
|   regionEUR:smoke  -0.011760 0.002835  134 -4.147578 0.00006 -0.01
|   regionSEAR:smoke -0.009663 0.004494  134 -2.150191 0.03334 -0.01
|   regionWPR:smoke  -0.008164 0.003110  134 -2.625207 0.00967 -0.01
|                    Upper 0.95
|   regionAMR        11.557078
|   regionEMR         7.279840
|   regionEUR        13.576310
|   regionSEAR       13.760568
|   regionWPR        11.356561
|   regionAMR:smoke  -0.003625
|   regionEMR:smoke  -0.001166
|   regionEUR:smoke  -0.006152
|   regionSEAR:smoke -0.000775
|   regionWPR:smoke  -0.002013
```

```
wald(fitr3, 'HE')
```

```
|      numDF denDF  F.value p.value
|   HE     1   134 137.6607 <.00001
|            Estimate Std.Error DF  t-value  p-value Lower 0.95 Upper
|   log(HE) 3.688028 0.314332  134 11.73289 <.00001 3.066333   4.309
```

```
Anova(fitr3)
```

```
|   Anova Table (Type II tests)
|
|   Response: LifeExp
|                 Sum Sq  Df  F value    Pr(>F)
|   region        634.67   5  11.9873 1.339e-09 ***
|   smoke           6.95   1   0.6561   0.41936
|   log(HE)      1457.70   1 137.6607 < 2.2e-16 ***
|   hiv           425.67   1  40.1992 3.250e-09 ***
|   special       569.62   1  53.7926 1.910e-11 ***
|   region:smoke  225.27   5   4.2548   0.00127 **
|   Residuals    1418.94 134
|   ---
```

```
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
anova(fitr3)
```

```
|   Analysis of Variance Table
|
|   Response: LifeExp
|                 Df Sum Sq Mean Sq  F value    Pr(>F)
|   region         5 8942.0 1788.40 168.8905 < 2.2e-16 ***
|   smoke          1  141.2  141.20  13.3348 0.0003725 ***
|   log(HE)        1 1788.2 1788.22 168.8740 < 2.2e-16 ***
|   hiv            1  337.6  337.64  31.8856 9.375e-08 ***
|   special        1  537.9  537.94  50.8014 5.711e-11 ***
|   region:smoke   5  225.3   45.05   4.2548 0.0012701 **
|   Residuals    134 1418.9   10.59
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
wald(fitr3, 'smoke')
```

```
|          numDF denDF  F.value p.value
|   smoke      6   134 3.655058 0.00216
|                     Estimate  Std.Error DF  t-value   p-value Lower
|   smoke              0.010666 0.002712  134  3.933292 0.00013  0.00
|   regionAMR:smoke   -0.009904 0.003175  134 -3.119787 0.00222 -0.01
|   regionEMR:smoke   -0.007235 0.003068  134 -2.357952 0.01982 -0.01
|   regionEUR:smoke   -0.011760 0.002835  134 -4.147578 0.00006 -0.01
|   regionSEAR:smoke  -0.009663 0.004494  134 -2.150191 0.03334 -0.01
|   regionWPR:smoke   -0.008164 0.003110  134 -2.625207 0.00967 -0.01
|                     Upper 0.95
|   smoke              0.016029
|   regionAMR:smoke   -0.003625
|   regionEMR:smoke   -0.001166
|   regionEUR:smoke   -0.006152
|   regionSEAR:smoke  -0.000775
|   regionWPR:smoke   -0.002013
```

```r
library(p3d)
Plot3d(LifeExp ~  smoke + HE | region, dd)
```

```
|     region     col
|   1    AFR    blue
|   2    AMR   green
|   3    EMR  orange
|   4    EUR magenta
|   5   SEAR    cyan
|   6    WPR     red
|
|   Use left mouse to rotate, middle mouse (or scroll) to zoom, righ
```

```r
Fit3d( fitr3, other.vars=list(hiv=0,special=0))
```

```
|   Warning in log(HE): NaNs produced
```

```r
#Id3d(par=2, labels = dd$country)
par3d(windowRect=c(10,10,700,700))
rgl.snapshot('regions.png')
```

3.8.1 Exercise

1. Explore this data further producing informative graphs.

4 Exploring Regression Using R

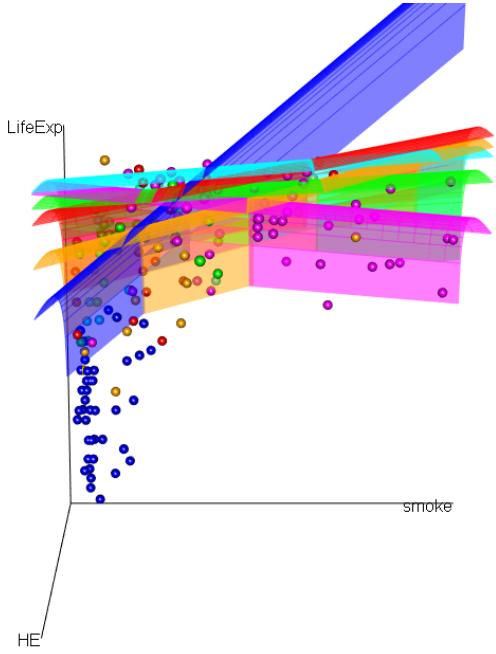
The following is an example of exploring data using regression in R. But it's very unrealistic.

Real data analysis does not start with a neat rectangular data set.

Hadley Wickham: "Data analysis is the process by which data becomes understanding, knowledge and insight"

The process involves much more than running regressions:

- subject matter understanding
- getting data


LifeExp
smoke
HE

- tidying the data
- formulating research questions
- transforming data to variables for analysis
- exploratory visualization
- deciding on a starting model
  - not too big, not too small
  - includes key variables based on subject matter and questions
- modeling fitting
- model diagnostics
- refining the model: dropping some terms and adding others
- formulating parameter functions for estimation and testing
- interpreting results
- GO BACK and iterate a varying number of previous steps in various orders

4.1 Interactive 3D

```
Init3d(cex=1)
ds <- dd
ds$Life <- ds$LE
ds$Cigarettes <- ds$smoke
ds$Health <- ds$HE
ds$area <-ds$region
ds$area <- tr(ds$region,
              c("AFR", "AMR", "EMR", "EUR", "SEAR", "WPR"),
              c("Africa","South Asia","Other")[c(1,3,3,3,2,3)])


Plot3d( Life ~ Cigarettes + Health |region,ds)
```

```
|     region     col
|   1    AFR    blue
|   2    AMR   green
|   3    EMR  orange
|   4    EUR magenta
```

```
|   5   SEAR    cyan
|   6    WPR     red
|
|   Use left mouse to rotate, middle mouse (or scroll) to zoom, righ
```

```
fg()
spinto()
Axes3d()
#  Id3d()
fit <- lm( Life ~ Cigarettes, ds)
summary(fit)
```

```
|
|   Call:
|   lm(formula = Life ~ Cigarettes, data = ds)
|
|   Residuals:
|        Min       1Q   Median       3Q      Max
|   -19.2997  -5.7258   0.6864   6.5300  14.9811
```

```
|
|   Coefficients:
|                Estimate Std. Error t value Pr(>|t|)
|   (Intercept) 6.508e+01  8.560e-01   76.03  < 2e-16 ***
|   Cigarettes  6.915e-03  8.547e-04    8.09 7.99e-14 ***
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
|
|   Residual standard error: 7.969 on 183 degrees of freedom
|     (9 observations deleted due to missingness)
|   Multiple R-squared:  0.2635,    Adjusted R-squared:  0.2594
|   F-statistic: 65.46 on 1 and 183 DF,  p-value: 7.987e-14
```

```
wald(fit)
```

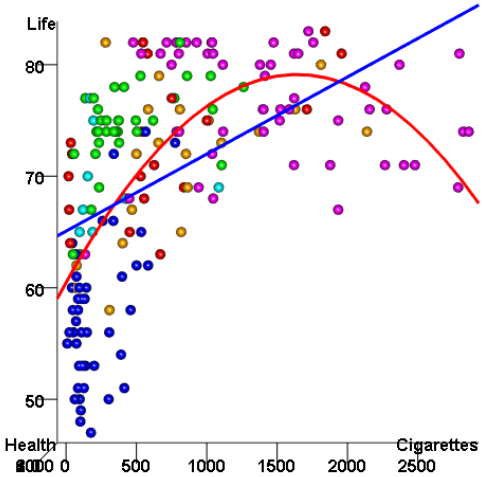
```
|    numDF denDF  F.value p.value
|        2   183 7194.457 <.00001
|                Estimate  Std.Error DF  t-value   p-value Lower 0.95
|   (Intercept) 65.075840 0.855974  183 76.025515 <.00001 63.386994
```

```
|   Cigarettes   0.006915 0.000855   183   8.090493 <.00001   0.005228
```

```
Fit3d(fit, lwd = 3)
fitsq <- lm( Life ~ Cigarettes+I(Cigarettes^2), ds)
Fit3d(fitsq, lwd = 3, col = 'red')
spin(0,0,0)
#  Id3d(pad=1)
# Id3d("Canada")
# Id3d("United States")
par3d(windowRect=c(10,10,700,700))
rgl.snapshot('quadsmoke.png')
Pop3d(2)
```

4.1.1 Controlling for Health

```
spin(-90,0,0)

fitlin <- lm( Life ~ Cigarettes + Health, ds)
summary(fitlin)
```


Life
80
70
60
50
Health
0
500
1000
1500
2000
2500
Cigarettes

```
|
|   Call:
|   lm(formula = Life ~ Cigarettes + Health, data = ds)
|
|   Residuals:
|       Min      1Q  Median      3Q     Max
|   -17.960  -4.309   1.161   5.304  11.772
|
|   Coefficients:
|                Estimate Std. Error t value Pr(>|t|)
|   (Intercept) 6.368e+01  7.464e-01  85.313  < 2e-16 ***
|   Cigarettes  4.312e-03  7.658e-04   5.631 6.84e-08 ***
|   Health      3.125e-03  3.585e-04   8.717 1.90e-15 ***
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
|
|   Residual standard error: 6.633 on 179 degrees of freedom
```

```
|     (12 observations deleted due to missingness)
|   Multiple R-squared:  0.4786,    Adjusted R-squared:  0.4728
|   F-statistic: 82.16 on 2 and 179 DF,  p-value: < 2.2e-16
```

```
fith <- lm( Life ~ Cigarettes + Health + log( Health),ds)
Fit3d(fitlin, col = 'pink')
Fit3d(fith, col = 'red')
```

```
|   Warning in log(Health): NaNs produced
```

```
Pop3d(2)
```

4.2 A more interesting model?

1. Health Expenditures
2. Proportion provided through government

```
ds$propGovt <- with(ds, govt/total) # proportion of health exp.
                                    # from Govt
Plot3d( Life ~ Health + propGovt |area, ds)
```

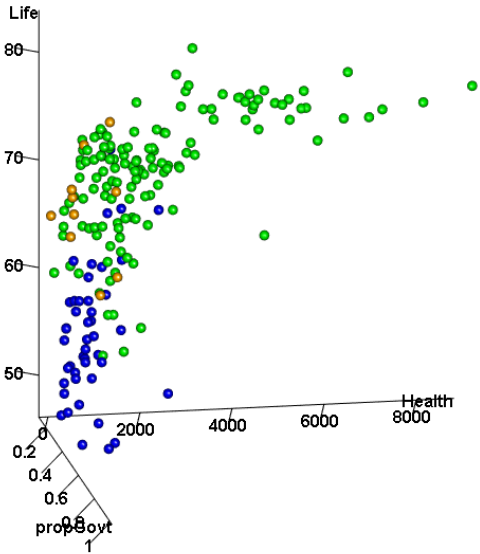
```
|           area    col
|   1     Africa   blue
|   2      Other  green
|   3 South Asia orange
|
|   Use left mouse to rotate, middle mouse (or scroll) to zoom, righ
```

```
fg()
Axes3d()
spin(-10,15,0)
par3d(windowRect=c(10,10,700,700))
rgl.snapshot('health-pgovt.png')
```

Try something that looks sensible:

The relationship between Life Expectance and Health Expentitures per capita and the proportion of health expenditures funelled through the government

```
ds$propGovt <- with(ds, govt/total)
                    # proportion of health exp. from Govt
```


Life
80
70
60
50
Health
0
2000
4000
6000
8000
0.2
0.4
0.6
0.8
1
propGovt

```
fit <- lm( Life ~ (Health + log(Health) + propGovt)  * area , ds,
          na.action = na.exclude)
summary(fit)
```

```
|
|   Call:
|   lm(formula = Life ~ (Health + log(Health) + propGovt) * area,
|       data = ds, na.action = na.exclude)
|
|   Residuals:
|        Min       1Q   Median       3Q      Max
|   -13.4282  -1.9480   0.1869   2.1919  14.7809
|
|   Coefficients:
|                                Estimate Std. Error t value Pr(>|t|
|   (Intercept)                 39.041364   5.426332   7.195 1.68e-1
|   Health                      -0.002672   0.003847  -0.695 0.48822
|   log(Health)                  2.368302   1.239444   1.911 0.05764
```

```
|   propGovt                    16.083451   3.854716   4.172 4.71e-0
|   areaOther                    4.316456   6.528366   0.661 0.50934
|   areaSouth Asia              15.629164  15.611788   1.001 0.31813
|   Health:areaOther             0.002218   0.003872   0.573 0.56736
|   Health:areaSouth Asia        0.006672   0.015873   0.420 0.67473
|   log(Health):areaOther        2.454985   1.393231   1.762 0.07977
|   log(Health):areaSouth Asia   1.029188   4.155759   0.248 0.80468
|   propGovt:areaOther         -17.211452   4.343678  -3.962 0.00010
|   propGovt:areaSouth Asia    -21.993937   8.781094  -2.505 0.01315
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
|
|   Residual standard error: 4.007 on 178 degrees of freedom
|     (4 observations deleted due to missingness)
|   Multiple R-squared:  0.817, Adjusted R-squared:  0.8057
|   F-statistic: 72.24 on 11 and 178 DF,  p-value: < 2.2e-16
```

```
Plot3d( Life ~ Health + propGovt | area, ds)
```

```
|           area    col
|   1     Africa   blue
|   2      Other  green
|   3 South Asia orange
|
|   Use left mouse to rotate, middle mouse (or scroll) to zoom, righ
```

```
Fit3d( fit)
```

```
|   Warning in log(Health): NaNs produced
```

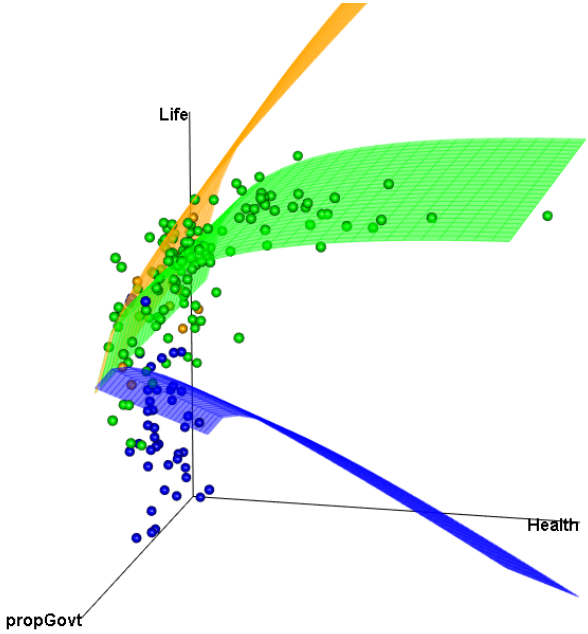
```
spin(13,15,10)
par3d(windowRect=c(10,10,700,700))
rgl.snapshot('health-pgovt-fit.png')
```

Question: Is this model too big for the data?

Should we drop Health expenditures?

None of the coefficients relating to Health are significant.

Can we conclude that "Health" does not add to the predictive power of this


Life
Health
propGovt

model?

4.2.0.1 Type II SS

Type II sums of squares are slightly less prone to massive misinterpretation since each test %>% satistifies the POM.

```
Anova(fit)  # Type II:
```

```
|   Anova Table (Type II tests)
|
|   Response: Life
|                     Sum Sq  Df F value    Pr(>F)
|   Health             19.55   1  1.2175 0.2713384
|   log(Health)       942.29   1 58.6882 1.142e-12 ***
|   propGovt           24.05   1  1.4979 0.2226053
|   area             2187.86   2 68.1331 < 2.2e-16 ***
|   Health:area         6.63   2  0.2065 0.8136300
|   log(Health):area   50.69   2  1.5785 0.2091650
|   propGovt:area     269.57   2  8.3948 0.0003281 ***
|   Residuals        2857.93 178
```

```
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
            # slightly less prone to massive misinterpretation
```

4.2.1 Simultaneous tests of groups of coefficients:

```
wald(fit, "Health")
```

```
|          numDF denDF F.value p.value
|   Health     6   178 29.4052 <.00001
|                              Estimate  Std.Error DF  t-value   p-v
|   Health                     -0.002672 0.003847  178 -0.694585 0.4
|   log(Health)                 2.368302 1.239444  178  1.910778 0.0
|   Health:areaOther            0.002218 0.003872  178  0.573005 0.5
|   Health:areaSouth Asia       0.006672 0.015873  178  0.420359 0.6
|   log(Health):areaOther       2.454985 1.393231  178  1.762080 0.0
|   log(Health):areaSouth Asia  1.029188 4.155759  178  0.247653 0.8
|                              Upper 0.95
```

```
|   Health                        0.004919
|   log(Health)                   4.814197
|   Health:areaOther              0.009858
|   Health:areaSouth Asia         0.037995
|   log(Health):areaOther         5.204360
|   log(Health):areaSouth Asia 9.230082
```

Note:

In the above, note that the overall evidence is **VERY strong** although individual p-values not even significant

I cannot sufficiently stress the importance of the principle this illustrates.

100% of beginning graduates students in statistics programs will fall into the trap of mis-interpreting p-values in regression output. It's as bad a professional error as a doctor amputating the wrong leg.

4.3 Two valid tests:

The Likelihood Ratio Test and the Wald test:

Null model:

```
fit0 <- lm( Life ~ propGovt  * area , ds,
            na.action = na.exclude)
```

OR you can use 'update':

```
fit0 <- update( fit, . ~ propGovt * area)
summary(fit0)
```

```
|
|   Call:
|   lm(formula = Life ~ propGovt + area + propGovt:area, data = ds,
|       na.action = na.exclude)
|
|   Residuals:
|        Min       1Q   Median       3Q      Max
|   -16.8782  -3.3650   0.6224   3.8491  17.9132
|
|   Coefficients:
```

```
|                              Estimate Std. Error t value Pr(>|t|)
|   (Intercept)                  48.387      2.668  18.133  < 2e-16 ***
|   propGovt                     19.124      5.023   3.807 0.000191 ***
|   areaOther                    20.277      3.153   6.431 1.06e-09 ***
|   areaSouth Asia               19.891      5.055   3.935 0.000118 ***
|   propGovt:areaOther           -9.998      5.631  -1.776 0.077451 .
|   propGovt:areaSouth Asia     -16.757      9.671  -1.733 0.084828 .
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
|
|   Residual standard error: 5.561 on 184 degrees of freedom
|     (4 observations deleted due to missingness)
|   Multiple R-squared:  0.6356,    Adjusted R-squared:  0.6257
|   F-statistic: 64.19 on 5 and 184 DF,  p-value: < 2.2e-16
```

1) LRT:

```
anova( fit, fit0)
```

```
|   Analysis of Variance Table
```

```
|
|   Model 1: Life ~ (Health + log(Health) + propGovt) * area
|   Model 2: Life ~ propGovt + area + propGovt:area
|     Res.Df    RSS Df Sum of Sq      F    Pr(>F)
|   1    178 2857.9
|   2    184 5690.7 -6   -2832.7 29.405 < 2.2e-16 ***
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2) Wald test:

- Doesn't require fitting a new model – works for linear parameters, not necessarily so good for non-linear parameters
- Tests the simultaneous hypotheses that ALL coefficients are = to 0 simultaneously – which is equivalent to dropping Health entirely:

```
wald(fit, "Health")
```

```
|           numDF denDF F.value p.value
|   Health      6   178 29.4052 <.00001
```

	Estimate	Std.Error	DF	t-value	p-v
Health	-0.002672	0.003847	178	-0.694585	0.4
log(Health)	2.368302	1.239444	178	1.910778	0.0
Health:areaOther	0.002218	0.003872	178	0.573005	0.5
Health:areaSouth Asia	0.006672	0.015873	178	0.420359	0.6
log(Health):areaOther	2.454985	1.393231	178	1.762080	0.0
log(Health):areaSouth Asia	1.029188	4.155759	178	0.247653	0.8

	Upper 0.95
Health	0.004919
log(Health)	4.814197
Health:areaOther	0.009858
Health:areaSouth Asia	0.037995
log(Health):areaOther	5.204360
log(Health):areaSouth Asia	9.230082

Note that the F-values are identical – which works in the case of OLS regression with normal error but rarely otherwise.

4.4 Explore interactions

There many approaches to simplifying a model. The most widespread is to trim down non-significant interactions.

If you do this it is vital to never drop a group of terms unless you either:

1. do it one term at a time making sure that you observe the principle of marginality (see the Appendix for more on the principle of marginality) as you go along, or
2. you only drop groups of terms when you have tested them **as a group**.

We could refit and use LRTs or we can use Wald tests:

```
wald(fit, ":")  # Use REGULAR EXPRESSION matching
```

```
|     numDF denDF  F.value p.value
|   :     6   178 4.077738 0.00074
|                              Estimate   Std.Error DF  t-value   p-
|   Health:areaOther             0.002218 0.003872  178  0.573005 0.
|   Health:areaSouth Asia        0.006672 0.015873  178  0.420359 0.
```

```
|   log(Health):areaOther          2.454985 1.393231  178  1.762080 0.
|   log(Health):areaSouth Asia     1.029188 4.155759  178  0.247653 0.
|   propGovt:areaOther           -17.211452 4.343678  178 -3.962414 0.
|   propGovt:areaSouth Asia      -21.993937 8.781094  178 -2.504692 0.
|                                Lower 0.95 Upper 0.95
|   Health:areaOther              -0.005422  0.009858
|   Health:areaSouth Asia         -0.024651  0.037995
|   log(Health):areaOther         -0.294391  5.204360
|   log(Health):areaSouth Asia    -7.171707  9.230082
|   propGovt:areaOther           -25.783184 -8.639720
|   propGovt:areaSouth Asia      -39.322380 -4.665493
```

```
                # to test whether there's any evidence of interacti
```

Explore how regular expressions work:

```
?regex
```

Testing all interactions that involve 'Health'

```
wald(fit, "Health.*:")
```

```
|                numDF denDF  F.value p.value
|   Health.*:        4   178 3.611567  0.0074
|                                Estimate Std.Error DF  t-value  p-val
|   Health:areaOther             0.002218 0.003872  178 0.573005 0.567
|   Health:areaSouth Asia        0.006672 0.015873  178 0.420359 0.674
|   log(Health):areaOther        2.454985 1.393231  178 1.762080 0.079
|   log(Health):areaSouth Asia   1.029188 4.155759  178 0.247653 0.804
|                                Upper 0.95
|   Health:areaOther             0.009858
|   Health:areaSouth Asia        0.037995
|   log(Health):areaOther        5.204360
|   log(Health):areaSouth Asia   9.230082
```

Testing all interactions that involve 'Govt' (always to check to make sure that you captured exactly the right terms)

```
wald(fit, "Govt:")
```

```
|          numDF denDF  F.value p.value
|   Govt:     2   178 8.394828 0.00033
|                            Estimate  Std.Error DF  t-value   p-valu
|   propGovt:areaOther       -17.21145 4.343678  178 -3.962414 0.0001
|   propGovt:areaSouth Asia -21.99394 8.781094  178 -2.504692 0.0131
|                            Upper 0.95
|   propGovt:areaOther       -8.639720
|   propGovt:areaSouth Asia -4.665493
```

4.4.1 Some comments on reading a model

```
summary(fit)
```

4.4.1.1 Table of coefficients and p-values:

```
|
|   Call:
```

```
lm(formula = Life ~ (Health + log(Health) + propGovt) * area,
    data = ds, na.action = na.exclude)

Residuals:
     Min       1Q   Median       3Q      Max
-13.4282  -1.9480   0.1869   2.1919  14.7809

Coefficients:
                              Estimate Std. Error t value Pr(>|t|
(Intercept)                 39.041364   5.426332   7.195 1.68e-1
Health                      -0.002672   0.003847  -0.695 0.48822
log(Health)                  2.368302   1.239444   1.911 0.05764
propGovt                    16.083451   3.854716   4.172 4.71e-0
areaOther                    4.316456   6.528366   0.661 0.50934
areaSouth Asia              15.629164  15.611788   1.001 0.31813
Health:areaOther             0.002218   0.003872   0.573 0.56736
Health:areaSouth Asia        0.006672   0.015873   0.420 0.67473
log(Health):areaOther        2.454985   1.393231   1.762 0.07977
log(Health):areaSouth Asia   1.029188   4.155759   0.248 0.80468
```

```
|   propGovt:areaOther          -17.211452   4.343678  -3.962 0.00010
|   propGovt:areaSouth Asia     -21.993937   8.781094  -2.505 0.01315
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
|
|   Residual standard error: 4.007 on 178 degrees of freedom
|     (4 observations deleted due to missingness)
|   Multiple R-squared:  0.817, Adjusted R-squared:  0.8057
|   F-statistic: 72.24 on 11 and 178 DF,  p-value: < 2.2e-16
```

Problems and limitations:

1. Except for very simple models this is generally misleading and not meaningful
2. Very few know how to interpret these correctly, even statisticians
3. The p-value answers how much evidence is there that this term adds to the model when all other terms are already in the model.
4. Only one degree of freedom per coefficient: never asks whether groups of terms are significant which is essential with categorical factors with 3 or more levels.

5. Often it's meaningless to change one term keeping others constant, e.g. `x` if `x^2` is also in the model.
6. Terms that are marginal to higher order interactions have a specific conditional interpretation that is generally just a very small and arbitrary part of the picture.
7. The interpretation of tests does not respect the principle of marginality.

```
anova(fit)
```

4.4.1.2 Type I (sequential) Tests and Sums of Squares

```
|   Analysis of Variance Table
|
|   Response: Life
|                     Df Sum Sq Mean Sq  F value    Pr(>F)
|   Health             1 6071.7  6071.7 378.1620 < 2.2e-16 ***
|   log(Health)        1 4091.9  4091.9 254.8571 < 2.2e-16 ***
|   propGovt           1   14.8    14.8   0.9238 0.3377716
```

```
|   area                2 2187.9  1093.9  68.1331 < 2.2e-16 ***
|   Health:area         2   79.0    39.5   2.4611 0.0882394 .
|   log(Health):area    2   44.2    22.1   1.3773 0.2549413
|   propGovt:area       2  269.6   134.8   8.3948 0.0003281 ***
|   Residuals         178 2857.9    16.1
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# The results depend on the order of the terms.
anova(update(fit, . ~ area * (propGovt + log(Health) + Health)))
```

```
|   Analysis of Variance Table
|
|   Response: Life
|                     Df Sum Sq Mean Sq  F value    Pr(>F)
|   area               2 9077.6  4538.8 282.6878 < 2.2e-16 ***
|   propGovt           1  718.7   718.7  44.7647 2.783e-10 ***
|   log(Health)        1 2566.2  2566.2 159.8331 < 2.2e-16 ***
|   Health             1    3.8     3.8   0.2359  0.627813
```

```
|   area:propGovt       2  160.9    80.4   5.0101  0.007642 **
|   area:log(Health)    2  225.3   112.7   7.0166  0.001166 **
|   area:Health         2    6.6     3.3   0.2065  0.813630
|   Residuals         178 2857.9    16.1
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Notes:

1. Each p-values asks whether there's evidence that each terms adds **to the previous terms** listed in the model.
2. Interpretation of tests respects marginality since interactions are listed after their included main effects and sub-interactions.
3. Factors with multiple degrees of freedom are tested jointly.

```
Anova(fit)
```

4.4.1.3 Type II Tests and Sums of Squares

```
|   Anova Table (Type II tests)
|
|   Response: Life
|                     Sum Sq  Df F value    Pr(>F)
|   Health             19.55   1  1.2175 0.2713384
|   log(Health)       942.29   1 58.6882 1.142e-12 ***
|   propGovt           24.05   1  1.4979 0.2226053
|   area             2187.86   2 68.1331 < 2.2e-16 ***
|   Health:area         6.63   2  0.2065 0.8136300
|   log(Health):area   50.69   2  1.5785 0.2091650
|   propGovt:area     269.57   2  8.3948 0.0003281 ***
|   Residuals        2857.93 178
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Notes:

1. Terms within a 'level' are each added last respecting marginality. e.g. each main effect is added last among main effects but not including higher-order interactions that contain the effect.

2. Main effects are interpretable as test of significance under assumption that there are no interactions.

4.4.1.4 Type III Tests and Sums of Squares

1. Popularized by SAS and SPSS
2. No universal definition so can be misleading
3. With interactions, main effects are averages over levels of interacting variables which can be misleading if groups sizes are unequal.
4. Loved by many researchers, deprecated by most statisticians ... like pie charts.

```
Anova(fit, type = 3)    # Type III Anova: Very popular but ....!
```

```
|   Anova Table (Type III tests)
|
|   Response: Life
|                     Sum Sq  Df F value    Pr(>F)
|   (Intercept)       831.13   1 51.7651 1.678e-11 ***
|   Health              7.75   1  0.4824 0.4882213
```

```
|   log(Health)         58.62   1  3.6511 0.0576404 .
|   propGovt           279.52   1 17.4090 4.706e-05 ***
|   area                18.52   2  0.5766 0.5628569
|   Health:area          6.63   2  0.2065 0.8136300
|   log(Health):area    50.69   2  1.5785 0.2091650
|   propGovt:area      269.57   2  8.3948 0.0003281 ***
|   Residuals         2857.93 178
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

4.4.1.5 Alternative – or supplementary – approaches

Answer specific questions.

e.g. None of the above are equivalent to Wald test for 'OVERALL SIGNIFICANCE':

```
wald(fit, "Health")
```

```
|           numDF denDF F.value p.value
|   Health      6   178 29.4052 <.00001
```

	Estimate	Std.Error	DF	t-value	p-v
Health	-0.002672	0.003847	178	-0.694585	0.4
log(Health)	2.368302	1.239444	178	1.910778	0.0
Health:areaOther	0.002218	0.003872	178	0.573005	0.5
Health:areaSouth Asia	0.006672	0.015873	178	0.420359	0.6
log(Health):areaOther	2.454985	1.393231	178	1.762080	0.0
log(Health):areaSouth Asia	1.029188	4.155759	178	0.247653	0.8

	Upper 0.95
Health	0.004919
log(Health)	4.814197
Health:areaOther	0.009858
Health:areaSouth Asia	0.037995
log(Health):areaOther	5.204360
log(Health):areaSouth Asia	9.230082

```
wald(fit, "area")
```

	numDF	denDF	F.value	p.value
area	8	178	20.09158	<.00001

	Estimate	Std.Error	DF	t-value	p-
areaOther	4.316456	6.528366	178	0.661185	0.
areaSouth Asia	15.629164	15.611788	178	1.001113	0.
Health:areaOther	0.002218	0.003872	178	0.573005	0.
Health:areaSouth Asia	0.006672	0.015873	178	0.420359	0.
log(Health):areaOther	2.454985	1.393231	178	1.762080	0.
log(Health):areaSouth Asia	1.029188	4.155759	178	0.247653	0.
propGovt:areaOther	-17.211452	4.343678	178	-3.962414	0.
propGovt:areaSouth Asia	-21.993937	8.781094	178	-2.504692	0.

	Lower 0.95	Upper 0.95
areaOther	-8.566498	17.199409
areaSouth Asia	-15.178841	46.437168
Health:areaOther	-0.005422	0.009858
Health:areaSouth Asia	-0.024651	0.037995
log(Health):areaOther	-0.294391	5.204360
log(Health):areaSouth Asia	-7.171707	9.230082
propGovt:areaOther	-25.783184	-8.639720
propGovt:areaSouth Asia	-39.322380	-4.665493

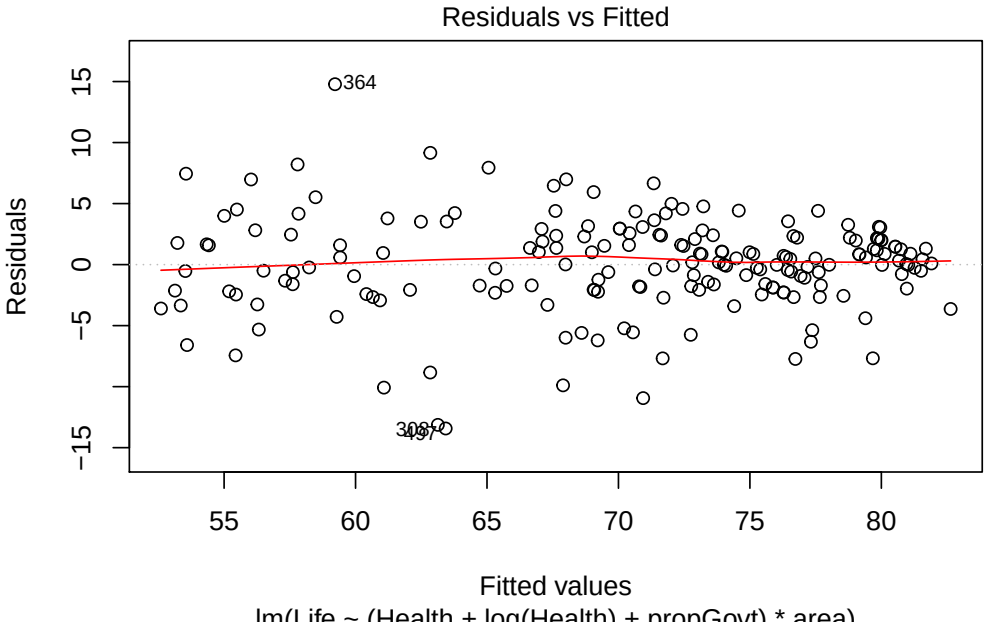
```
wald(fit, "propGovt")
```

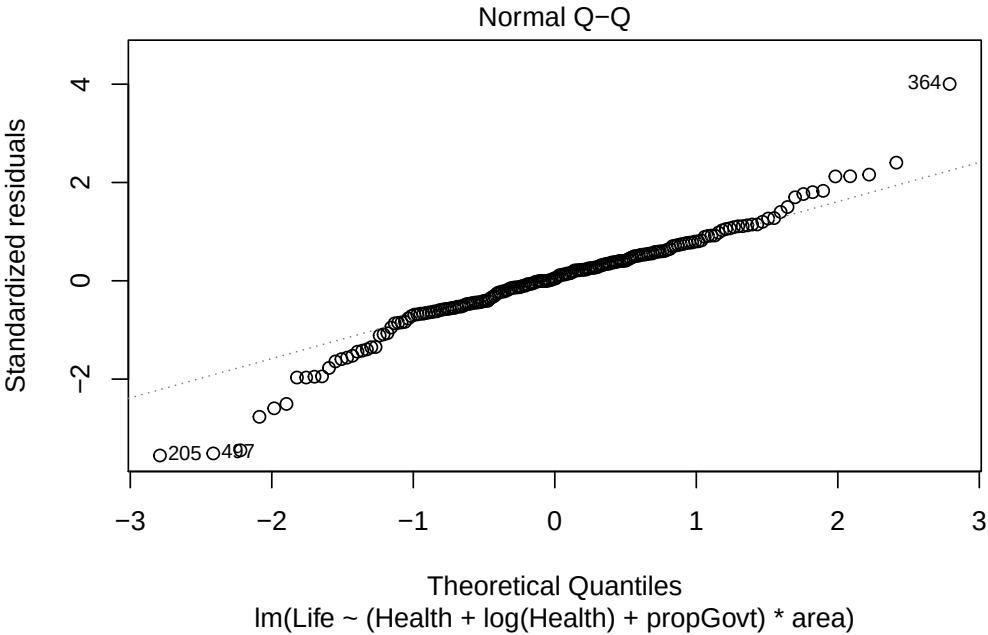
```
|            numDF denDF  F.value p.value
|   propGovt     3   178 6.095867 0.00057
|                           Estimate  Std.Error DF  t-value   p-valu
|   propGovt                 16.08345 3.854716  178  4.172409 0.0000
|   propGovt:areaOther      -17.21145 4.343678  178 -3.962414 0.0001
|   propGovt:areaSouth Asia -21.99394 8.781094  178 -2.504692 0.0131
|                           Upper 0.95
|   propGovt                23.690273
|   propGovt:areaOther      -8.639720
|   propGovt:areaSouth Asia -4.665493
```

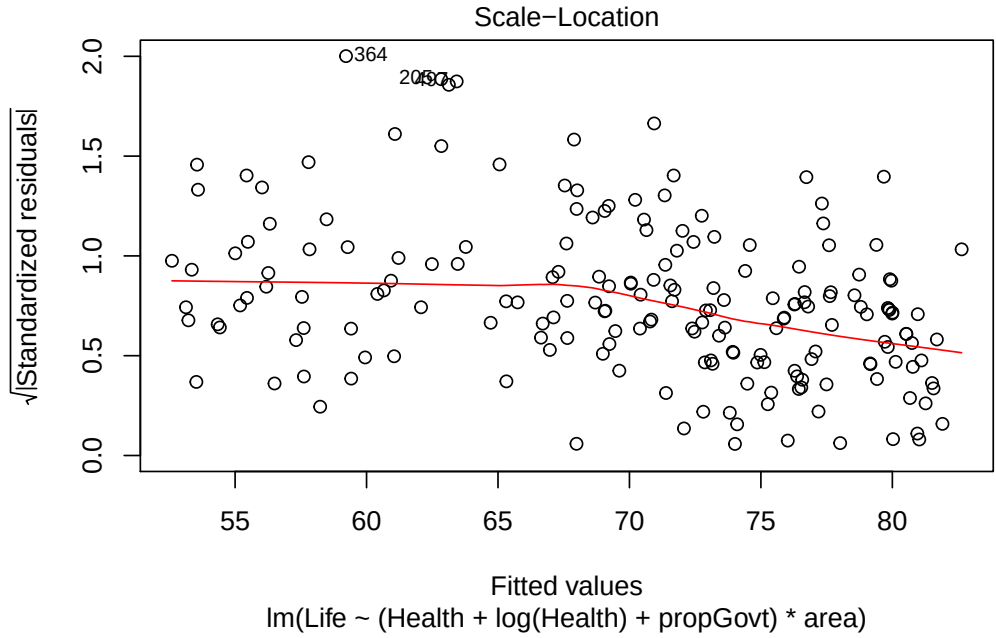
4.5 Regression diagnostics – quick

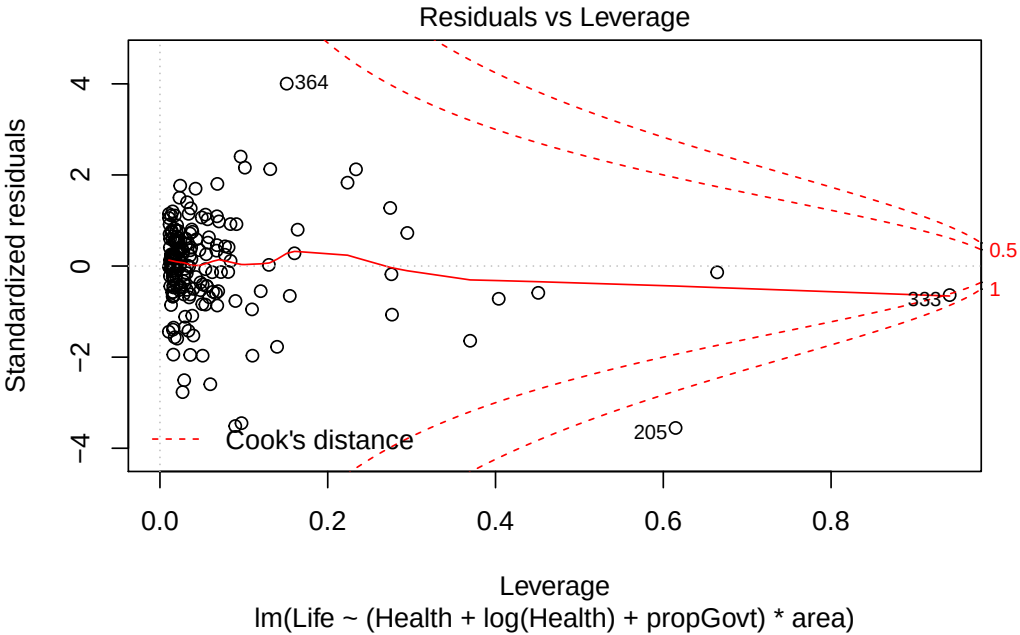
```
plot(fit)
```

4.5.0.1 Traditional




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(Life ~ (Health + log(Health) + propGovt) * area)
364
205
49
97
−3
−2
−1
0
1
2
3
−2
0
2
4


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
364
55
60
65
70
75
80
Fitted values
lm(Life ~ (Health + log(Health) + propGovt) * area)


Residuals vs Leverage
Standardized residuals
Leverage
lm(Life ~ (Health + log(Health) + propGovt) * area)
Cook's distance
0.5
1
364
333
205
4
2
0
−2
−4
0.0
0.2
0.4
0.6
0.8

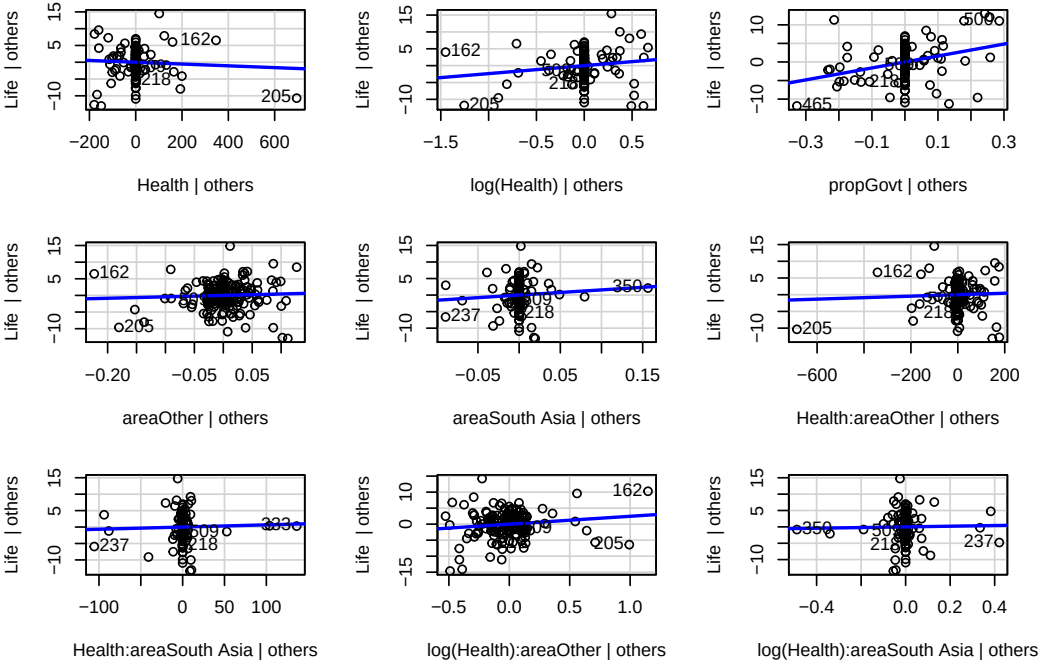
produces 4 plots

1) resid ~ fit
2) normal quantiles of residuals – Why would this matter??? GEQ. What can it mean if observed residuals are not normal? Clue: Why would you expect them to be normal anyways?
3) scale-location for heteroscedasticity
4) Residual vs leverage plot – will see deeper meaning of this plot

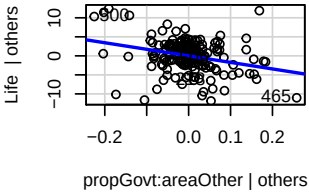
points with high Cook's distance might have strong influence on fitted vals

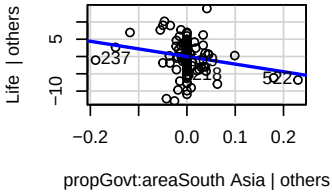
Note: added-variable plots = partial residual leverage plots

```
avPlots(fit) # look at these as if they
```

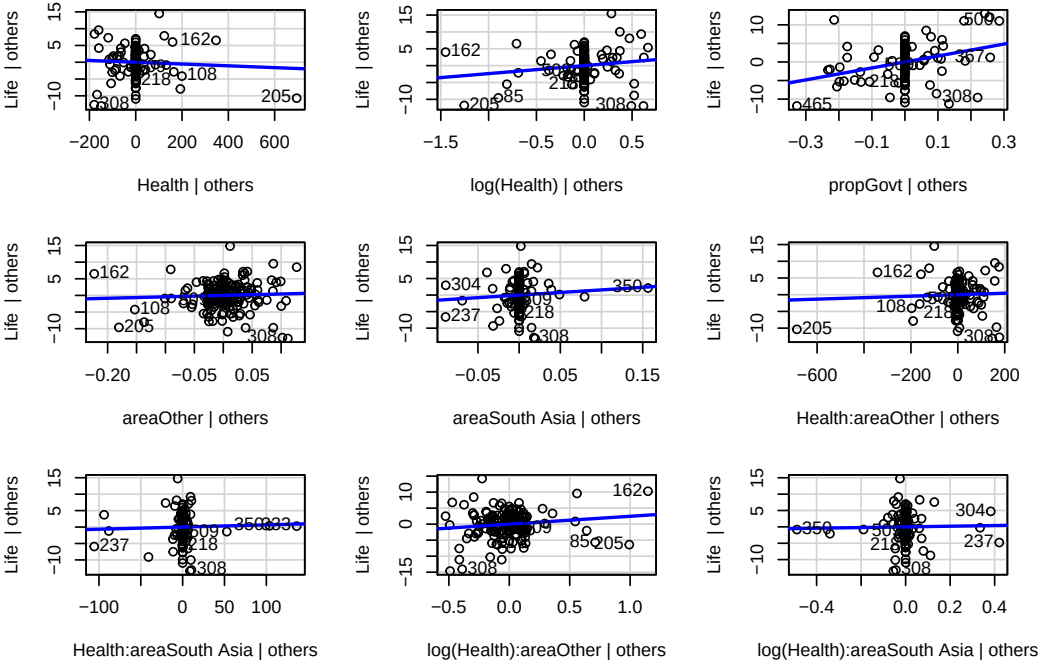

Life | others
Health | others
log(Health) | others
propGovt | others
areaOther | others
areaSouth Asia | others
Health:areaOther | others
Health:areaSouth Asia | others
log(Health):areaOther | others
log(Health):areaSouth Asia | others

Added−Variable Plots

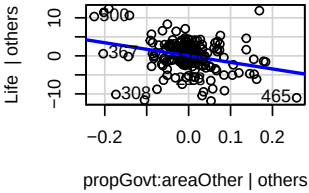


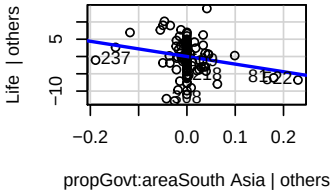


```r
            # were simple regression plots
avPlots(fit, id =list(n = 3))
```

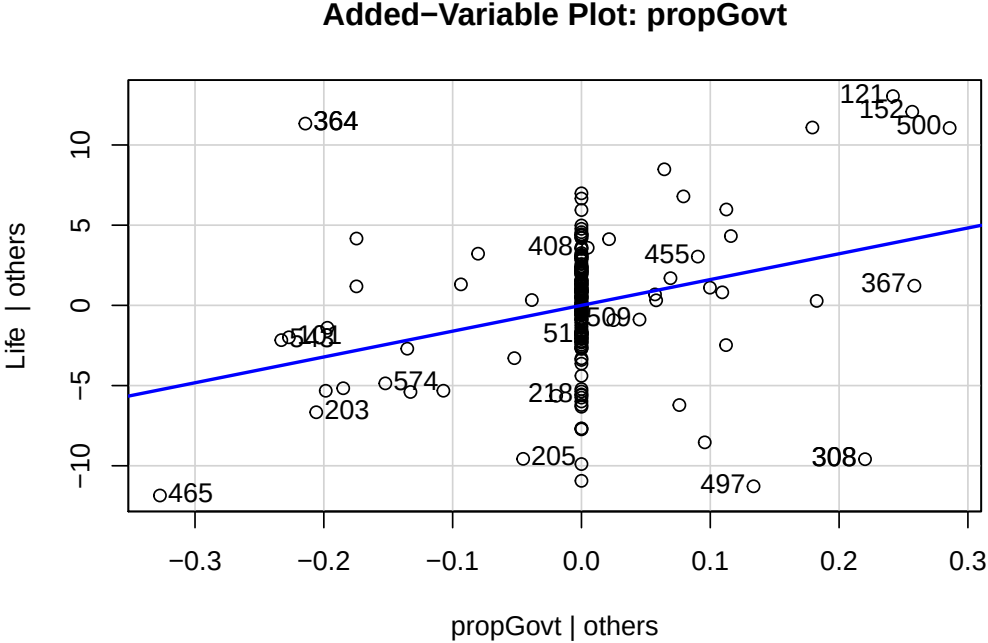

Life | others
Health | others
log(Health) | others
propGovt | others
areaOther | others
areaSouth Asia | others
Health:areaOther | others
Health:areaSouth Asia | others
log(Health):areaOther | others
log(Health):areaSouth Asia | others

Added−Variable Plots





```
avPlot(fit, 'propGovt', id =list(n = 10))
```


Added−Variable Plot: propGovt
Life | others
propGovt | others
364
121
152
500
408
455
367
509
574
203
205
308
497
465
10
5
0
−5
−10
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3

Interactively, you can use:

```
avPlot(fit, 'propGovt', id.method = "identify")
```

4.5.1 Visualize fit for diagnostics

In 3D

```
Plot3d( Life ~ Health + propGovt | area, ds)
```

```
|           area    col
|   1     Africa   blue
|   2      Other  green
|   3 South Asia orange
|
|   Use left mouse to rotate, middle mouse (or scroll) to zoom, righ
```

```
Fit3d( fit , resid = T)
```

```
|   Warning in log(Health): NaNs produced
```

```
#  Id3d()  # outliers?
```

2D

```
summary(ds)
```

```
|                    country         iso3      region   HealthExpPC.G
|    Afghanistan         :  1   AFG    :  1   AFR :46   Min.   :   2.
|    Albania             :  1   AGO    :  1   AMR :35   1st Qu.:  42.
|    Algeria             :  1   ALB    :  1   EMR :22   Median : 183.
|    Andorra             :  1   AND    :  1   EUR :53   Mean   : 782.
|    Angola              :  1   ARE    :  1   SEAR:11   3rd Qu.: 650.
|    Antigua and Barbuda:  1   ARG    :  1   WPR :27   Max.   :7696.
|    (Other)             :188   (Other):188             NA's   :3
|    HealthExpPC.Tot.ppp HealthExpPC.Govt.ppp HealthExpPC.Tot.exch
|    Min.   :  16.99     Min.   :   3.61      Min.   :  13.9
|    1st Qu.: 166.09     1st Qu.:  79.92      1st Qu.:  90.7
|    Median : 518.83     Median : 286.72      Median : 333.8
|    Mean   :1114.08     Mean   : 770.12      Mean   :1094.2
```

```
|    3rd Qu.:1333.26     3rd Qu.: 926.80      3rd Qu.: 971.4
|    Max.   :8607.88     Max.   :5794.45      Max.   :9120.8
|    NA's   :4           NA's   :4            NA's   :3
|        total              govt             private          sex
|    Min.   :  16.99   Min.   :   3.61   Min.   :   0.51   BTSX:194
|    1st Qu.: 166.09   1st Qu.:  79.92   1st Qu.:  57.10   FMLE:  0
|    Median : 518.83   Median : 286.72   Median : 187.50   MLE :  0
|    Mean   :1114.08   Mean   : 770.12   Mean   : 343.96
|    3rd Qu.:1333.26   3rd Qu.: 926.80   3rd Qu.: 475.42
|    Max.   :8607.88   Max.   :5794.45   Max.   :4653.69
|    NA's   :4         NA's   :4         NA's   :4
|    lifeexp.Birth    lifeexp.At60   smoking.tobacco.current smoking
|    Min.   :47.00   Min.   :11.00   Min.   : 4.00           Min.
|    1st Qu.:64.00   1st Qu.:17.00   1st Qu.:14.00           1st Qu.
|    Median :72.50   Median :19.00   Median :23.00           Median
|    Mean   :70.01   Mean   :19.36   Mean   :22.63           Mean
|    3rd Qu.:76.00   3rd Qu.:22.00   3rd Qu.:29.00           3rd Qu.
|    Max.   :83.00   Max.   :26.00   Max.   :57.00           Max.
|                                    NA's   :47              NA's
```

```
|    smoking.cig.current smoking.cig.daily   Pop.Total         Pop.M
|    Min.   : 4.0        Min.   : 2.00     Min.   :      1   Min.
|    1st Qu.:12.0        1st Qu.: 8.25     1st Qu.:   1696   1st Qu.
|    Median :21.0        Median :16.00     Median :   7790   Median
|    Mean   :21.2        Mean   :17.22     Mean   :  36360   Mean
|    3rd Qu.:29.0        3rd Qu.:23.75     3rd Qu.:  24535   3rd Qu.
|    Max.   :57.0        Max.   :55.00     Max.   :1390000   Max.
|    NA's   :46          NA's   :48                          NA's
|    Pop.pCntUnder15 Pop.pCntOver60  Pop.pCntAnnGrowth consumption.c
|    Min.   :13.12   Min.   : 0.81   Min.   :-9.100    Min.   :   9.
|    1st Qu.:18.72   1st Qu.: 5.20   1st Qu.:-2.300    1st Qu.: 179.
|    Median :28.65   Median : 8.53   Median :-1.300    Median : 529.
|    Mean   :28.73   Mean   :11.16   Mean   :-1.453    Mean   : 730.
|    3rd Qu.:37.75   3rd Qu.:16.69   3rd Qu.:-0.500    3rd Qu.:1039.
|    Max.   :49.99   Max.   :31.92   Max.   : 0.800    Max.   :2861.
|                                                      NA's   :9
|    hiv_prev15_49       LifeExp           LE             smoke
|    Min.   : 0.000   Min.   :47.00   Min.   :47.00   Min.   :   9.0
|    1st Qu.: 0.200   1st Qu.:64.00   1st Qu.:64.00   1st Qu.: 179.0
```

```
|    Median : 0.400   Median :72.50   Median :72.50   Median : 529.0
|    Mean   : 1.817   Mean   :70.01   Mean   :70.01   Mean   : 730.1
|    3rd Qu.: 1.200   3rd Qu.:76.00   3rd Qu.:76.00   3rd Qu.:1039.0
|    Max.   :26.000   Max.   :83.00   Max.   :83.00   Max.   :2861.0
|    NA's   :40                                       NA's   :9
|          HE               hiv             special            yq
|    Min.   :  16.99   Min.   : 0.000   Min.   :0.00000   Min.   :48
|    1st Qu.: 166.09   1st Qu.: 0.200   1st Qu.:0.00000   1st Qu.:60
|    Median : 518.83   Median : 0.400   Median :0.00000   Median :72
|    Mean   :1114.08   Mean   : 1.817   Mean   :0.01546   Mean   :69
|    3rd Qu.:1333.26   3rd Qu.: 1.200   3rd Qu.:0.00000   3rd Qu.:76
|    Max.   :8607.88   Max.   :26.000   Max.   :1.00000   Max.   :80
|    NA's   :4         NA's   :40                         NA's   :42
|         Life         Cigarettes         Health                area
|    Min.   :47.00   Min.   :   9.0   Min.   :  16.99   Africa    :
|    1st Qu.:64.00   1st Qu.: 179.0   1st Qu.: 166.09   Other     :1
|    Median :72.50   Median : 529.0   Median : 518.83   South Asia:
|    Mean   :70.01   Mean   : 730.1   Mean   :1114.08
|    3rd Qu.:76.00   3rd Qu.:1039.0   3rd Qu.:1333.26
```

```
|    Max.   :83.00   Max.   :2861.0   Max.   :8607.88
|                    NA's   :9        NA's   :4
|       propGovt
|    Min.   :0.1296
|    1st Qu.:0.4561
|    Median :0.6070
|    Mean   :0.5940
|    3rd Qu.:0.7478
|    Max.   :0.9989
|    NA's   :4
```

Create a prediction data frame with values for which you want to predict model with observed values for Health and area but controlling for propGovt

4.5.1.1 3 ways:

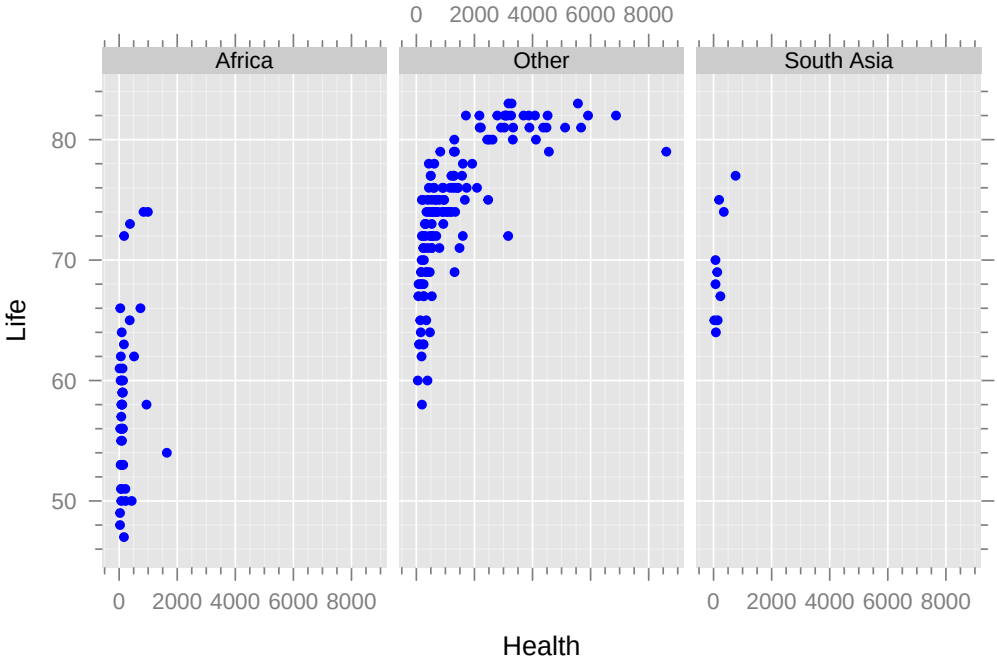
1. easiest: use data BUT need to control for propGovt
2. Generate cartesian product of values of predictors: but hard to generate conditional ranges
3. Create prediction data set with original data augmented by extra points

```
ds <- sortdf( ds, ~ Health)
pred1 <- rbind(ds,NA,ds,NA, ds,NA, ds,NA, ds, NA)
    # to set five values for predicted propGovt
pred1$propGovt <- rep(seq(.1,.9,by=.2), each = nrow(ds)+1)

pred1$Life.fit <- predict(fit, newdata = pred1)
```

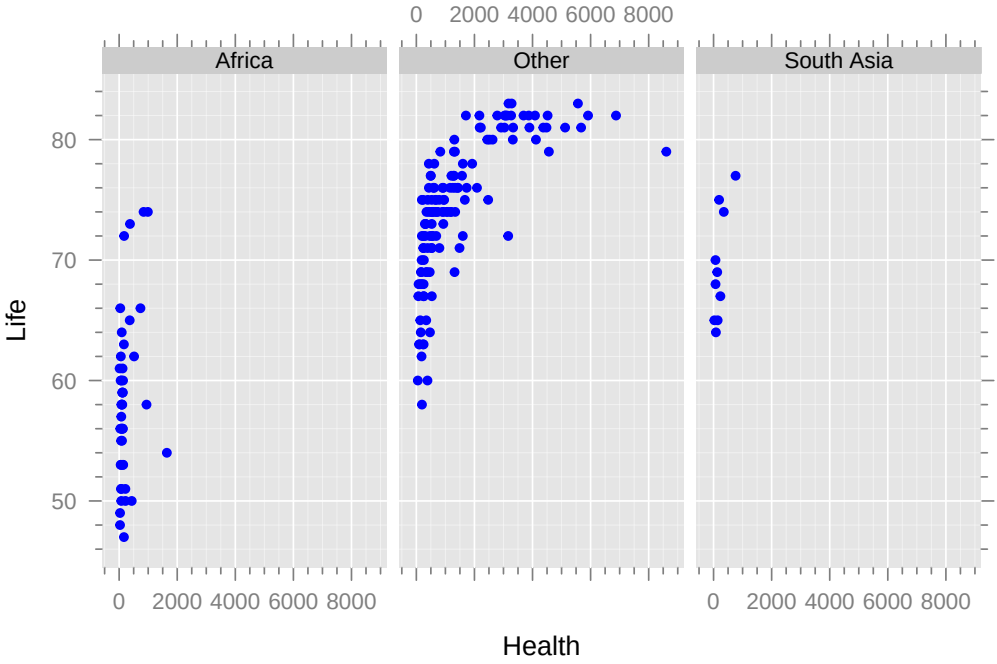
4.5.1.2 1. add predicted values to data frame

In 'panels':

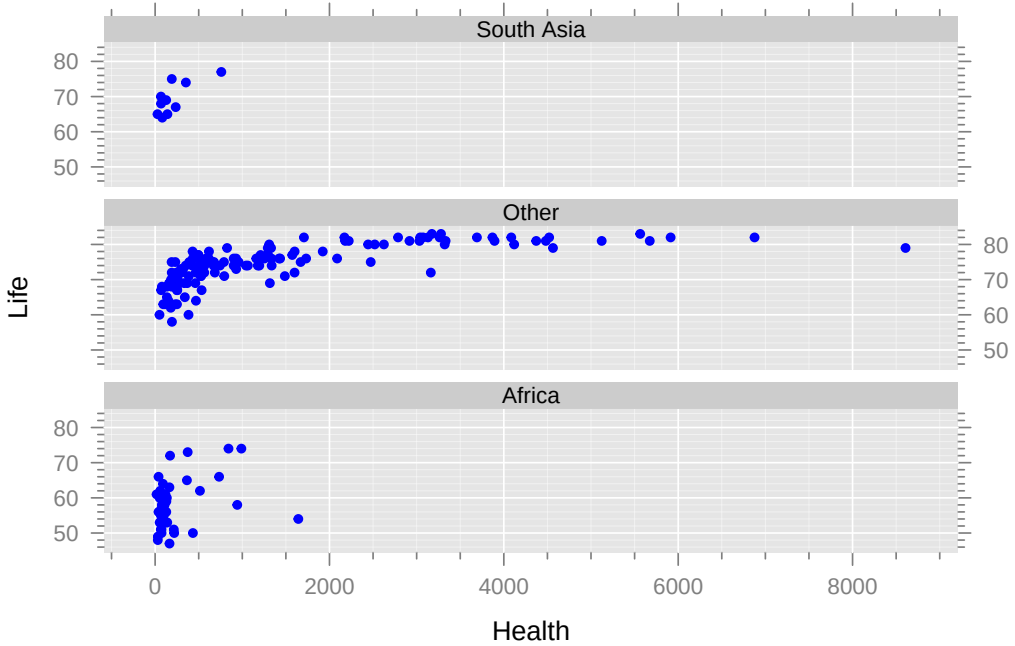
```
xyplot(Life ~ Health | area, ds)
```


Africa
Other
South Asia
Life
Health
80
70
60
50
0
2000
4000
6000
8000

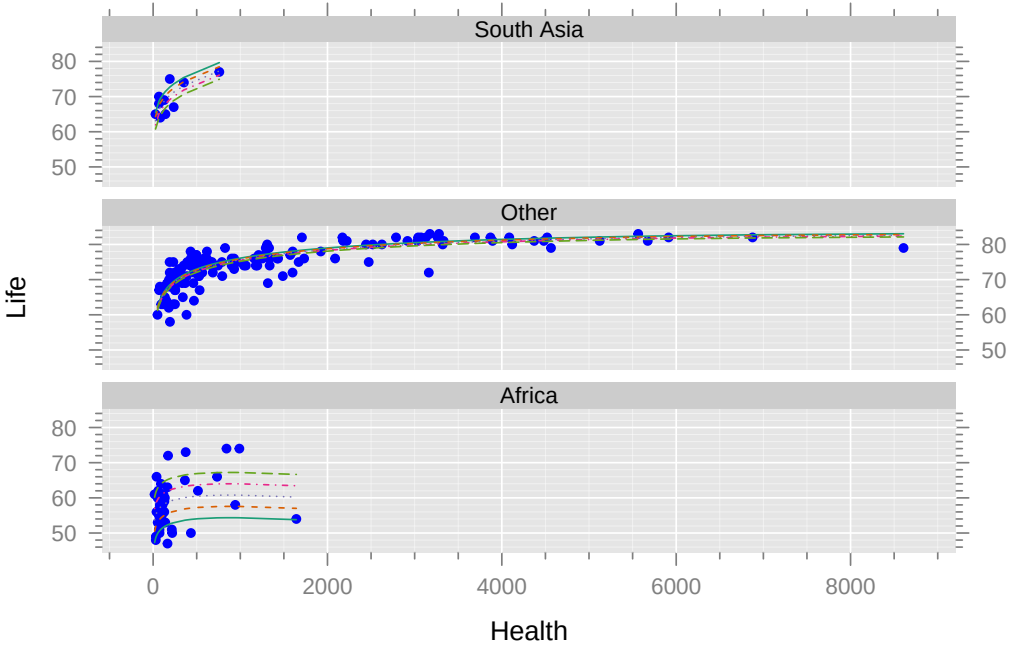
```
gd() # ggplot2 look-alike

xyplot(Life ~ Health | area, ds)
```


Africa
Other
South Asia
Life
Health
80
70
60
50
0
2000
4000
6000
8000

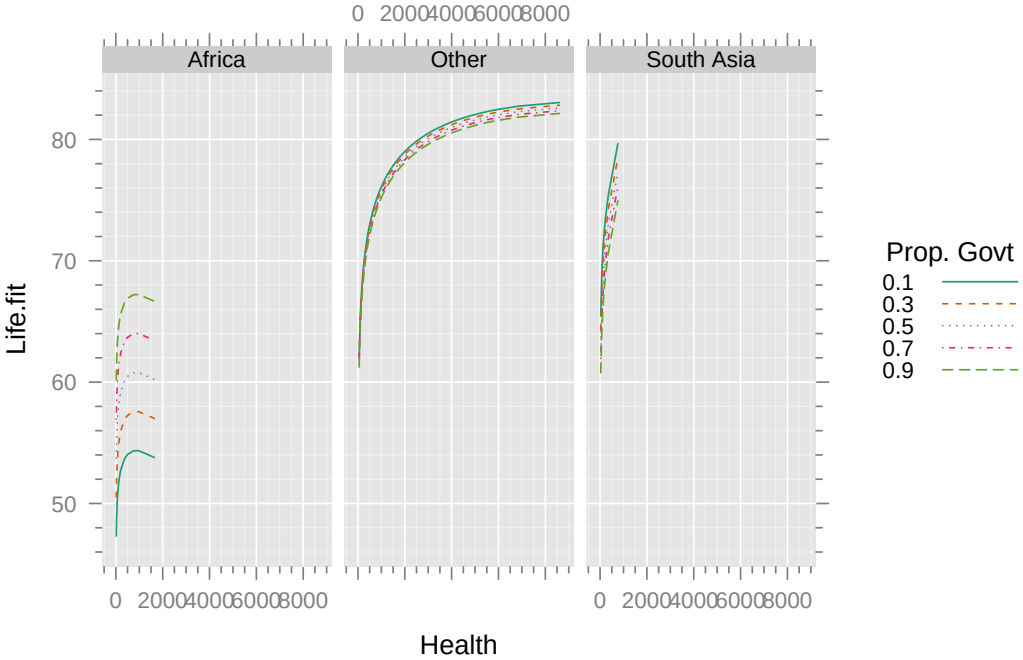
```
xyplot(Life ~ Health | area, ds, layout = c(1,3))
```


South Asia
Other
Africa
Life
Health
80
70
60
50
0
2000
4000
6000
8000

```
xyplot(Life ~ Health | area, ds, layout = c(1,3)) +
  xyplot( Life.fit ~ Health | area, pred1, groups = propGovt,
          type = 'l')
```

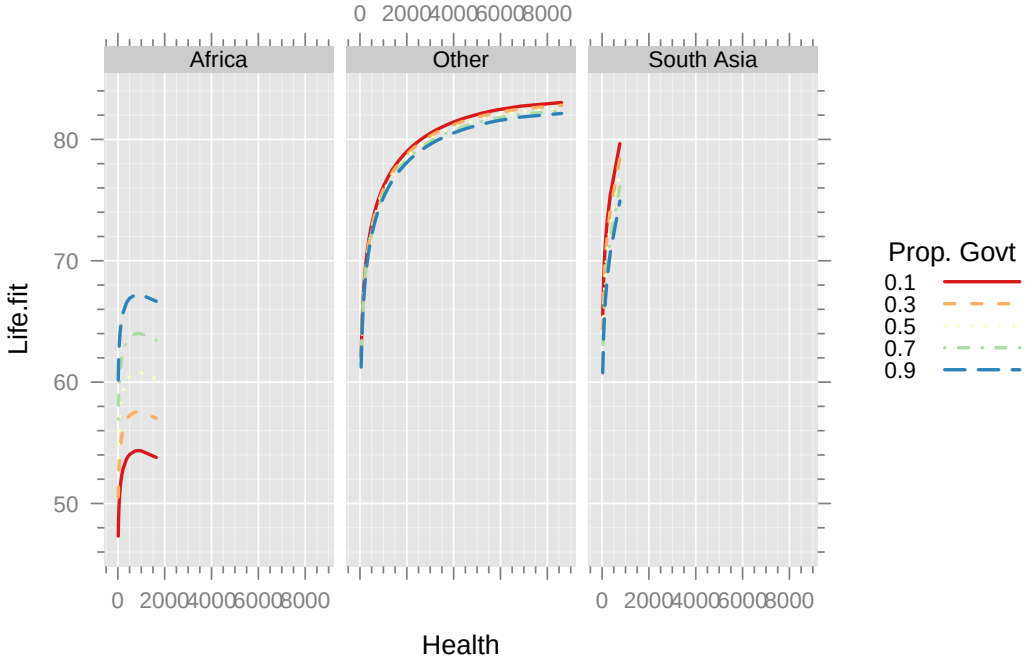

South Asia
Other
Africa
Life
Health
80
70
60
50
0
2000
4000
6000
8000

```
gd(lwd = 2)
(p <- xyplot(Life.fit ~ Health | area,
             pred1, groups = propGovt,
             type = 'l',
             auto.key = list(space='right',
                             lines = T, points = F,
                             title = 'Prop. Govt',
                             cex.title = 1)))
```

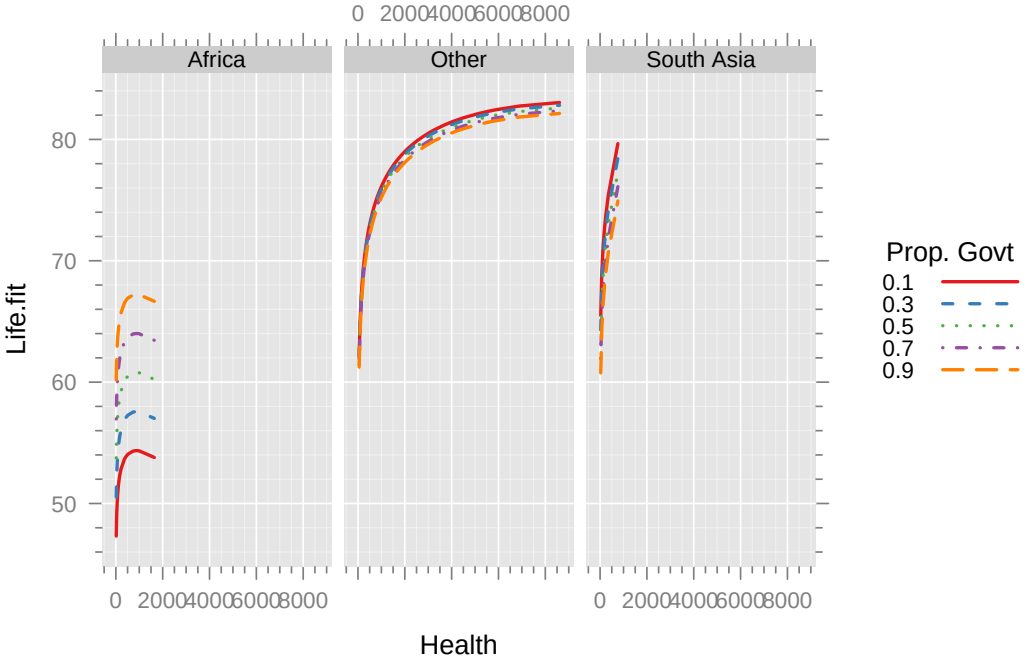

0
2000
4000
6000
8000
Africa
Other
South Asia
80
70
60
50
Life.fit
Health
Prop. Govt
0.1
0.3
0.5
0.7
0.9

Some colour palettes you can choose from. display.brewer.all() Note that the first group consists of 'progressive' palettes, the second group of categorical palettes (one is 'paired') and the third group of 'bipolar' palettes. Note that yellow often doesn't work for lines that blend into the background so you might have to avoid palettes that include yellow for some purposes.

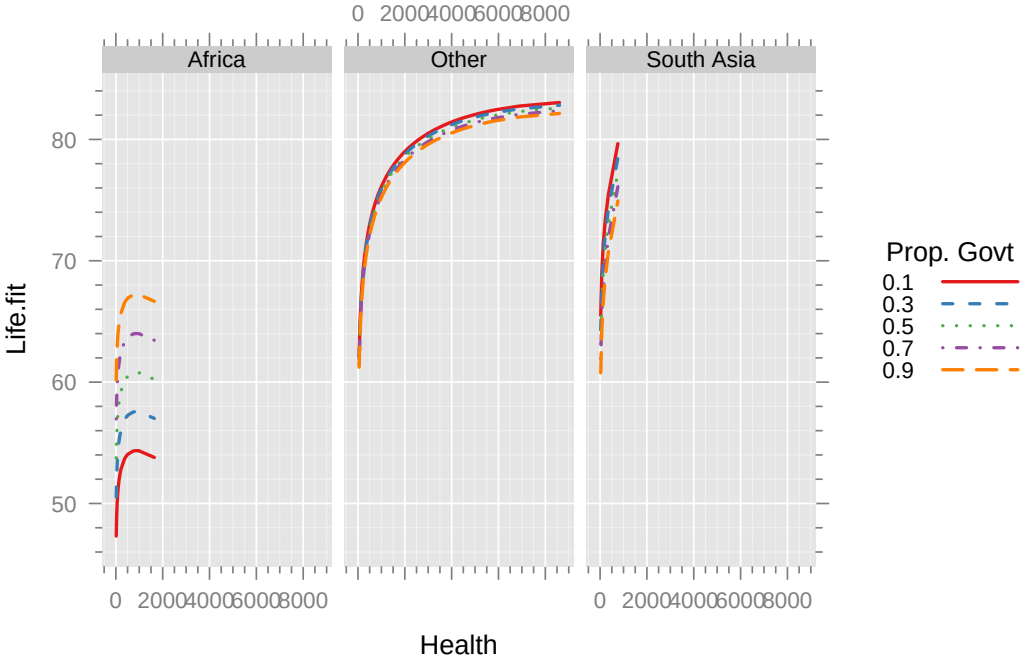
```
gd(lwd = 2, col = brewer.pal(5,"Spectral"))
p  # replots with new parameters
```


Africa
Other
South Asia
Life.fit
Health
Prop. Govt
0.1
0.3
0.5
0.7
0.9
0
2000
4000
6000
8000
50
60
70
80

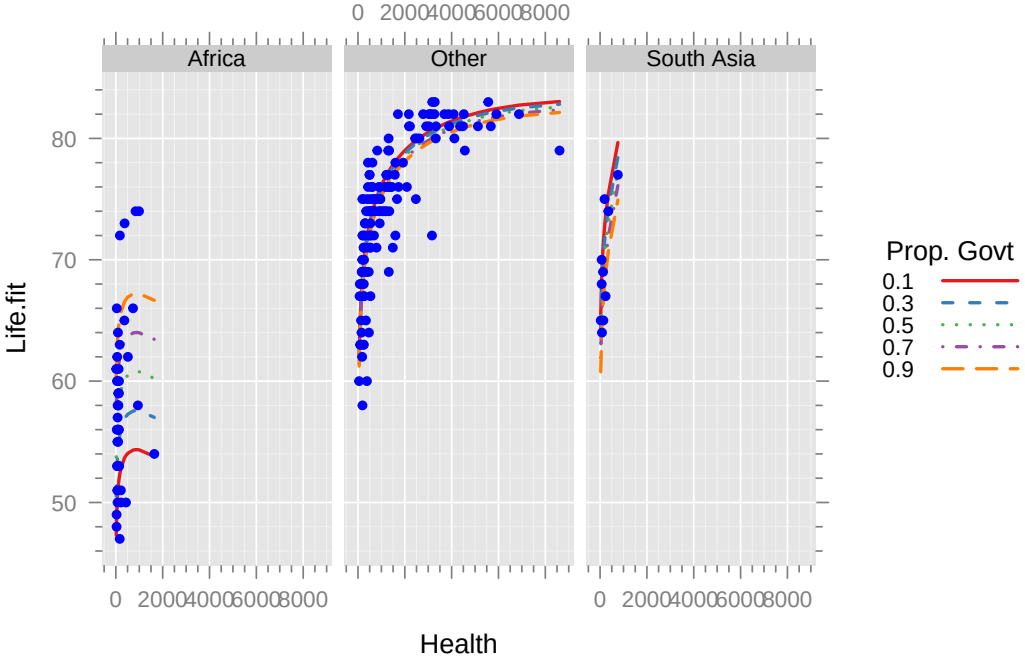
```
gd(lwd = 2, col = brewer.pal(5,"Set1"))
p
```


Africa
Other
South Asia
Life.fit
Health
0
2000
4000
6000
8000
50
60
70
80
Prop. Govt
0.1
0.3
0.5
0.7
0.9

```
gd(lty = 1)
p
```

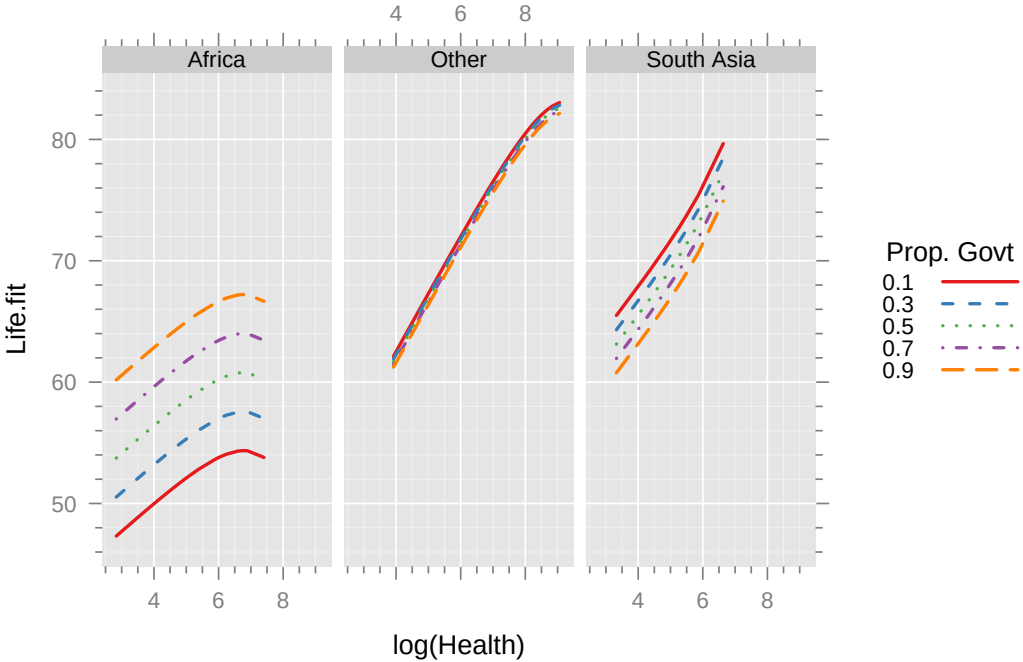

Africa
Other
South Asia
Life.fit
Health
0
2000
4000
6000
8000
50
60
70
80
Prop. Govt
0.1
0.3
0.5
0.7
0.9

```
p + xyplot(Life ~ Health | area, ds)
```

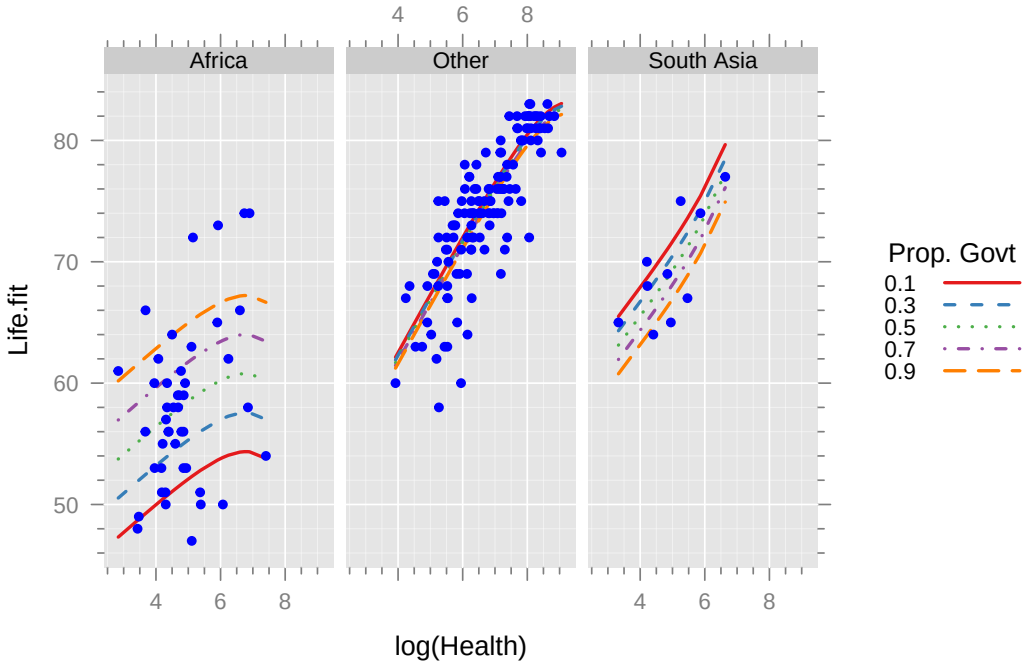

Africa
Other
South Asia
Life.fit
Health
Prop. Govt
0.1
0.3
0.5
0.7
0.9
0
2000
4000
6000
8000
50
60
70
80

Probably better with log(Health)

```
(p <- xyplot(Life.fit ~ log(Health) | area,
             pred1, groups = propGovt,
             type = 'l',
             auto.key = list(space = 'right',lines = T,
                             points = F, title = 'Prop. Govt',
                             cex.title = 1)))
```


Africa
Other
South Asia
Life.fit
log(Health)
Prop. Govt
0.1
0.3
0.5
0.7
0.9

```
p + xyplot(Life ~ log(Health) | area, ds)
```

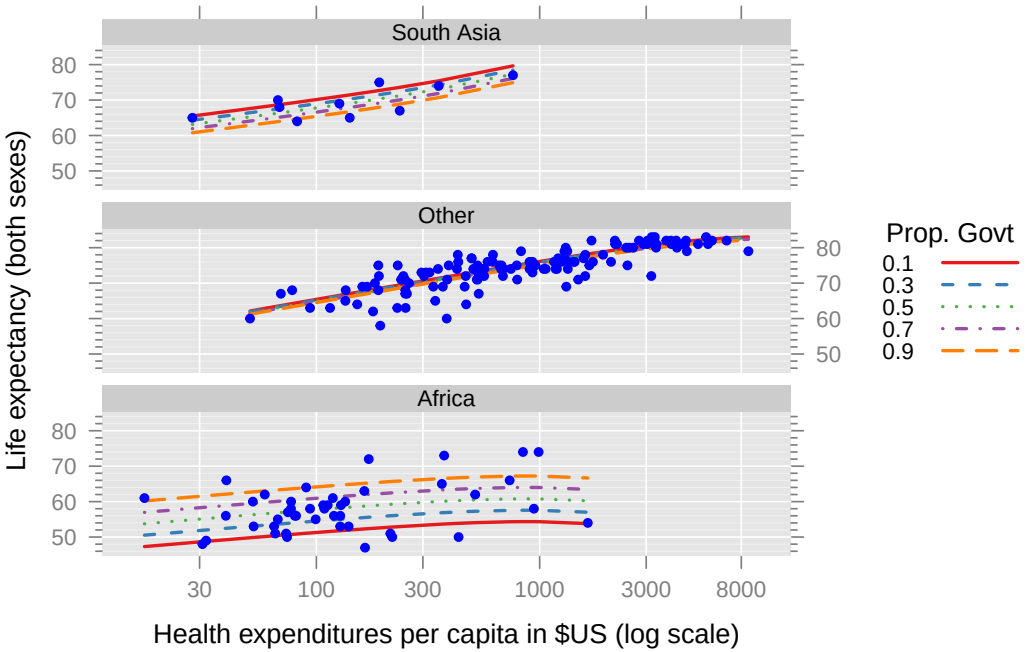

Africa
Other
South Asia
Life.fit
log(Health)
Prop. Govt
0.1
0.3
0.5
0.7
0.9

BUT NEVER NEVER USE axes that are not meaningful

Your work will be written off as incomprehensible!

Also: use meaningful labels

```
update(p,
       xlab = "Health expenditures per capita in $US (log scale)",
       ylab = "Life expectancy (both sexes)",
       layout = c(1,3),
       scales = list( x = list(
         at = log(c(30,100,300,1000,3000,8000)),
         labels = c(30,100,300,1000,3000,8000)))) +
  xyplot(Life ~ log(Health) | area, ds)
```


South Asia
Other
Africa
80
70
60
50
30
100
300
1000
3000
8000
Life expectancy (both sexes)
Health expenditures per capita in $US (log scale)
Prop. Govt
0.1
0.3
0.5
0.7
0.9

Dropping some observations: BEWARE

Two ways:

1) Drop from data set and refit
2) Add parameters for dummy variables for observations to drop and refit

Suppose we want to drop "Equatorial Guinea"

CAUTION: this needs good reflection BUT we often should approach this like a sensitivity analysis: e.g. "would it make a big difference if I dropped this point?"

```
ds$EqG <- 1*(ds$country == "Equatorial Guinea")

fit2 <- lm( Life ~
              (Health + log(Health) + propGovt) * area + EqG, ds,
            na.action = na.exclude)
```

OR

```
fit2 <- update(fit, . ~ . + EqG)
summary(fit2)
```

```
|
|   Call:
|   lm(formula = Life ~ Health + log(Health) + propGovt + area +
|       EqG + Health:area + log(Health):area + propGovt:area, data =
|       na.action = na.exclude)
|
|   Residuals:
|        Min       1Q   Median       3Q      Max
|   -13.3962  -1.8221   0.1869   2.1919  10.5391
|
|   Coefficients:
|                                 Estimate Std. Error t value Pr(>|t|
|   (Intercept)                  50.031832   6.036711   8.288 2.78e-1
|   Health                        0.012027   0.005459   2.203 0.02888
|   log(Health)                  -0.386600   1.412966  -0.274 0.78470
|   propGovt                     15.122468   3.735031   4.049 7.69e-0
|   areaOther                    -6.674012   6.982150  -0.956 0.34044
|   areaSouth Asia                4.638695  15.383108   0.302 0.76335
|   EqG                         -22.943658   6.239158  -3.677 0.00031
```

```
|   Health:areaOther             -0.012481   0.005475  -2.279 0.02384
|   Health:areaSouth Asia        -0.008027   0.015854  -0.506 0.61329
|   log(Health):areaOther         5.209887   1.541017   3.381 0.00088
|   log(Health):areaSouth Asia    3.784089   4.086121   0.926 0.35566
|   propGovt:areaOther          -16.250469   4.206623  -3.863 0.00015
|   propGovt:areaSouth Asia     -21.032953   8.491622  -2.477 0.01419
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
|
|   Residual standard error: 3.873 on 177 degrees of freedom
|     (4 observations deleted due to missingness)
|   Multiple R-squared:   0.83, Adjusted R-squared:  0.8185
|   F-statistic: 72.01 on 12 and 177 DF,  p-value: < 2.2e-16
```

```
Fit3d(fit2, other.vars = list( EqG = 0))
```

```
|   Warning in log(Health): NaNs produced
```

```
spin(-2,18,10)
par3d(windowRect=c(10,10,700,700))
```

```
rgl.snapshot('dropEqG.png')
```

Figure X: Note the change in the fitted surface for Africa when Equatorial Guinea is dropped from the model.

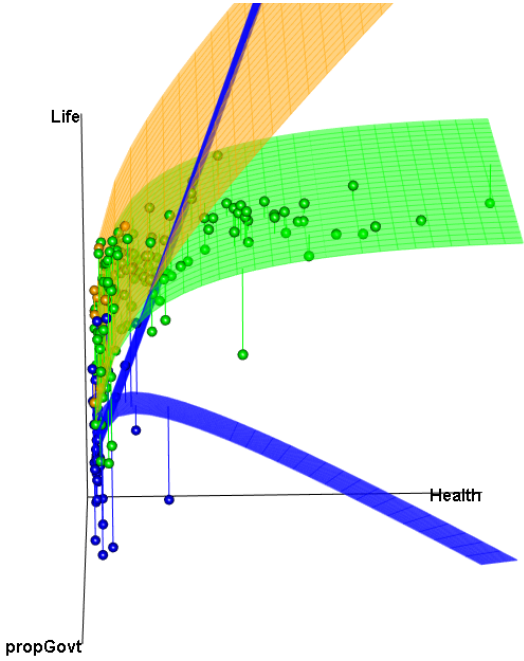
4.6 Asking questions:

4.6.1 Can we simplify the model?

Often this process focuses on the initial regression parameter and asks which ones 'can we drop'? Often starting with highest order interactions and working in.

BEWARE THE PRINCIPLE OF MARGINALITY

In general (i.e. 99.9% of the time) DO NOT eliminate a term without also eliminating all higher-order terms to which the term is *marginal*. e.g. 'Health' is marginal to 'Health:area', and 'Health:area' would be marginal to 'Health:propGovt:area'. Otherwise the resulting model loses invariance with respect to changes of origins of interacting variables.


Life
Health
propGovt

Here's a model that seems to tell a different story but it's perfectly equivalent. The individual terms estimate different aspects of the model, but the models as a whole are equivalent and produce the same fitted values.

```
fit2.eq <- lm( Life ~
                 area/(Health + log(Health) + propGovt)  + EqG -1,
               ds, na.action = na.exclude)
anova(fit2, fit2.eq)
```

```
|   Analysis of Variance Table
|
|   Model 1: Life ~ Health + log(Health) + propGovt + area + EqG + H
|       log(Health):area + propGovt:area
|   Model 2: Life ~ area/(Health + log(Health) + propGovt) + EqG - 1
|     Res.Df    RSS Df   Sum of Sq F Pr(>F)
|   1    177 2655.1
|   2    177 2655.1  0 -5.2751e-11
```

```
AIC( fit2, fit2.eq)
```

```
|           df      AIC
|   fit2    14 1068.266
|   fit2.eq 14 1068.266
```

```
summary(fit2)
```

```
|
|   Call:
|   lm(formula = Life ~ Health + log(Health) + propGovt + area +
|       EqG + Health:area + log(Health):area + propGovt:area, data =
|       na.action = na.exclude)
|
|   Residuals:
|        Min       1Q   Median       3Q      Max
|   -13.3962  -1.8221   0.1869   2.1919  10.5391
|
|   Coefficients:
```

```
|                                Estimate Std. Error t value Pr(>|t|
|   (Intercept)                 50.031832   6.036711   8.288 2.78e-1
|   Health                       0.012027   0.005459   2.203 0.02888
|   log(Health)                 -0.386600   1.412966  -0.274 0.78470
|   propGovt                    15.122468   3.735031   4.049 7.69e-0
|   areaOther                   -6.674012   6.982150  -0.956 0.34044
|   areaSouth Asia               4.638695  15.383108   0.302 0.76335
|   EqG                        -22.943658   6.239158  -3.677 0.00031
|   Health:areaOther            -0.012481   0.005475  -2.279 0.02384
|   Health:areaSouth Asia       -0.008027   0.015854  -0.506 0.61329
|   log(Health):areaOther        5.209887   1.541017   3.381 0.00088
|   log(Health):areaSouth Asia   3.784089   4.086121   0.926 0.35566
|   propGovt:areaOther         -16.250469   4.206623  -3.863 0.00015
|   propGovt:areaSouth Asia    -21.032953   8.491622  -2.477 0.01419
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
|
|   Residual standard error: 3.873 on 177 degrees of freedom
|     (4 observations deleted due to missingness)
```

```
|   Multiple R-squared:   0.83, Adjusted R-squared:  0.8185
|   F-statistic: 72.01 on 12 and 177 DF,  p-value: < 2.2e-16
```

```
summary(fit2.eq)
```

```
|
|   Call:
|   lm(formula = Life ~ area/(Health + log(Health) + propGovt) +
|       EqG - 1, data = ds, na.action = na.exclude)
|
|   Residuals:
|        Min       1Q   Median       3Q      Max
|   -13.3962  -1.8221   0.1869   2.1919  10.5391
|
|   Coefficients:
|                                 Estimate Std. Error t value Pr(>|t|
|   areaAfrica                   5.003e+01  6.037e+00   8.288 2.78e-1
|   areaOther                    4.336e+01  3.508e+00  12.358  < 2e-1
|   areaSouth Asia               5.467e+01  1.415e+01   3.864 0.00015
```

```
|   EqG                          -2.294e+01  6.239e+00  -3.677 0.00031
|   areaAfrica:Health             1.203e-02  5.459e-03   2.203 0.02888
|   areaOther:Health             -4.536e-04  4.217e-04  -1.075 0.28363
|   areaSouth Asia:Health         4.000e-03  1.488e-02   0.269 0.78843
|   areaAfrica:log(Health)       -3.866e-01  1.413e+00  -0.274 0.78470
|   areaOther:log(Health)         4.823e+00  6.150e-01   7.842 4.00e-1
|   areaSouth Asia:log(Health)    3.397e+00  3.834e+00   0.886 0.37674
|   areaAfrica:propGovt           1.512e+01  3.735e+00   4.049 7.69e-0
|   areaOther:propGovt           -1.128e+00  1.935e+00  -0.583 0.56072
|   areaSouth Asia:propGovt      -5.910e+00  7.626e+00  -0.775 0.43935
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
|
|   Residual standard error: 3.873 on 177 degrees of freedom
|     (4 observations deleted due to missingness)
|   Multiple R-squared:  0.9972,    Adjusted R-squared:  0.997
|   F-statistic:  4880 on 13 and 177 DF,  p-value: < 2.2e-16
```

```r
Anova(fit2)
```

```
|   Anova Table (Type II tests)
|
|   Response: Life
|                     Sum Sq  Df F value    Pr(>F)
|   Health             12.01   1  0.8003 0.3722085
|   log(Health)       763.67   1 50.9096  2.39e-11 ***
|   propGovt           19.56   1  1.3039 0.2550477
|   area             1845.55   2 61.5164 < 2.2e-16 ***
|   EqG               202.85   1 13.5230 0.0003124 ***
|   Health:area        79.23   2  2.6410 0.0741004 .
|   log(Health):area  171.81   2  5.7268 0.0038909 **
|   propGovt:area     240.45   2  8.0147 0.0004655 ***
|   Residuals        2655.08 177
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
Anova(fit2.eq)
```

```
|   Anova Table (Type II tests)
|
|   Response: Life
|                     Sum Sq  Df    F value    Pr(>F)
|   area              944894   3 20997.0169 < 2.2e-16 ***
|   EqG                  203   1    13.5230 0.0003124 ***
|   area:Health           91   3     2.0274 0.1117604
|   area:log(Health)     935   3    20.7877 1.385e-11 ***
|   area:propGovt        260   3     5.7778 0.0008611 ***
|   Residuals           2655 177
|   ---
|   Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
wald(fit2, "th:|th):")
```

```
|             numDF denDF  F.value p.value
|   th:|th):     4   177 3.143431 0.01582
```

```
|                                Estimate  Std.Error DF  t-value   p-v
|   Health:areaOther            -0.012481 0.005475  177 -2.279349 0.0
|   Health:areaSouth Asia       -0.008027 0.015854  177 -0.506274 0.6
|   log(Health):areaOther        5.209887 1.541017  177  3.380811 0.0
|   log(Health):areaSouth Asia   3.784089 4.086121  177  0.926083 0.3
|                                Upper 0.95
|   Health:areaOther            -0.001675
|   Health:areaSouth Asia        0.023261
|   log(Health):areaOther        8.251017
|   log(Health):areaSouth Asia 11.847874
```

Question: Under what conditions would two seemingly different models produce exactly the same fit (i.e. predicted values of Y)?

4.6.2 Asking specific questions

What question does each coefficient answer and how can we get answers to the questions we want?

PRINCIPLE

Except with very simple models, raw regression output generally answers few meaningful questions AND most important questions are rarely answered by raw regression output

Interpreting βs:

Each term involving 'area' is a comparison with the **REFERENCE LEVEL** when **ALL VARIABLES IN HIGHER ORDER INTERACTING TERMS are set to 0**.

Each term involving 'area' is a comparison with the **REFERENCE LEVEL** (Africa because it's the level that isn't showing) when **ALL VARIABLES IN HIGHER ORDER INTERACTING TERMS are set to 0**.

The model is:

$$
\begin{aligned}
Y = {} & \beta_0 + \beta_1 Health + \beta_2 ln(Health) + \beta_3 propGovt \\
& + \beta_4 area_{Other} + \beta_5 area_{SouthAsia} \\
& + \beta_6 EqG \\
& + \beta_7 Health \times area_{Other} + \beta_8 Health \times area_{SouthAsia} \\
& + \beta_9 ln(Health) \times area_{Other} + \beta_{10} ln(Health) \times area_{SouthAsia} \\
& + \beta_{11} propGovt \times area_{Other} + \beta_{12} propGovt \times area_{SouthAsia} \\
& + \varepsilon
\end{aligned}
$$

where $\varepsilon \sim N(0, \sigma^2)$ independently of predictors.

Note: Here we encounter the vital difference between assumptions that can be checked and assumptions that can't be checked. We are assuming that

1) errors are normal,
2) they have the same variance for each observation, and
3) they are independent of predictors.

The first two assumptions can be checked with diagnostics, the third, in general,

cannot. In econometrics it's recognized as a key assumption related to the 'exogeneity' of the predictors and the causal interpretation of the model. For a further treatment of this question see Murnane and Willett (2010) and Pearl and Mackenzie (2018).

4.7 Understanding coefficients

Q: What does β_1 mean?

A: It's the expected change in Y when you change $Health$ by one unit keeping all other terms constant — **BUT THAT'S IMPOSSIBLE**

We'll have more luck with β_3: It's the expected change in Y when you change *progGovt* by one unit keeping all other terms constant, i.e. when all variables that interact with *propGovt* are equal to 0.

i.e. when $area_{Other} = area_{SouthAsia} = 0$

i.e. **in Africa**

So we have a clear interpretation: it's the expected change in Life Expectancy

using our model to compare a hypothetical country where health expenditures are entirely supported by the government with a hypothetical country in which they are entirely private **in Africa**.

This is not obvious to a casual user of regression and most surely not to most clients and readers of academic journals.

4.7.1 How can we get answers to meaningful questions?

You're in luck. Calculus comes in handy.

What's the 'effect' (a very misused word and I'm still looking for a better one) of increasing Health expenditures by 1 dollar?

$$
\begin{aligned}
\frac{\partial E(Y)}{\partial Health} &= 0 \times \beta_0 + \beta_1 + \beta_2 \frac{1}{Health} + 0\beta_3 \\
&\quad + 0 \times \beta_4 + 0 \times \beta_5 \\
&\quad + 0 \times \beta_6 \\
&\quad + \beta_7 area_{Other} + \beta_8 area_{SouthAsia} \\
&\quad + \beta_9 \frac{area_{Other}}{Health} + \beta_{10} \frac{area_{SouthAsia}}{Health} \\
&\quad + 0 \times \beta_{11} + 0 \times \beta_{12} \\
&= L\beta
\end{aligned}
$$

where

$$
L = \begin{bmatrix} 0 & 1 & \frac{1}{Health} & 0 & 0 & 0 & 0 & area_{Other} & area_{SouthAsia} & \frac{area_{Other}}{Health} & \frac{area_{SouthAsia}}{Health} & 0
\end{bmatrix}
$$

and

$$\beta = \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \\ \beta_5 \\ \beta_6 \\ \beta_7 \\ \beta_8 \\ \beta_9 \\ \beta_{10} \\ \beta_{11} \\ \beta_{12} \end{bmatrix}$$

which we can estimate with

$$\hat{\eta} = L\hat{\beta}$$

NOTE: This does not seem to depend on *propGovt*! Is this reasonable? What should we do about that?

Exercise: Explore what could be done with *propGovt*.

$\hat{\beta}$ is obtained with:

```
coef(fit2)
```

```
|                    (Intercept)                         Health
|                   50.031832427                    0.012026938
|                    log(Health)                        propGovt
|                   -0.386599991                   15.122467573
|                      areaOther                 areaSouth Asia
|                   -6.674012321                    4.638695485
|                            EqG               Health:areaOther
|                 -22.943657897                   -0.012480505
|         Health:areaSouth Asia       log(Health):areaOther
|                   -0.008026658                    5.209886574
|    log(Health):areaSouth Asia          propGovt:areaOther
|                    3.784089239                  -16.250468580
|       propGovt:areaSouth Asia
|                  -21.032953424
```

To estimate the marginal effect of Health Expenditures in 'Other' when Health expenditures = 100:

```
Lmat <- cbind( 0,1,1/100,0,0,0,0,   1, 0 , 1/100, 0,0,0)
Lmat
```

```
|        [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,
|   [1,]    0    1 0.01    0    0    0    0    1    0  0.01     0
```

```
Lmat %*% coef(fit2)
```

```
|             [,1]
|   [1,] 0.0477793
```

We could write matrix expression to get the variance, F-test, p-values etc., but it's already been done with the 'wald' function in 'spida2'. Other packages also have functions that do this, e.g. 'lht' in the 'car' package.

```
wald(fit2, Lmat)
```

```
|     numDF denDF  F.value p.value
|   1     1   177 67.95056 <.00001
```

```
|         Estimate Std.Error DF  t-value  p-value Lower 0.95 Upper 0.
|    [1,] 0.047779 0.005796  177 8.243213 <.00001 0.036341   0.059218
```

How could we mass produce this?

```
ex <- expression( cbind( 0,1,1/Health,0,0,0,0,
                         area == "Other", area == "South Asia",
                         (area == "Other")/Health,
                         (area == "South Asia")/Health,
                         0,0))
ex
```

```
|   expression(cbind(0, 1, 1/Health, 0, 0, 0, 0, area == "Other",
|       area == "South Asia", (area == "Other")/Health, (area ==
|           "South Asia")/Health, 0, 0))
```

```
with( list(Health=100, area = "South Asia"), eval(ex))
```

```
|         [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,
|    [1,]    0    1 0.01    0    0    0    0    0    1     0  0.01
```

```r
pred <- expand.grid( Health = seq(30,4000,10),
                     area = levels(ds$area))
head(pred)
```

```
|     Health   area
|   1     30 Africa
|   2     40 Africa
|   3     50 Africa
|   4     60 Africa
|   5     70 Africa
|   6     80 Africa
```

```r
tail(pred)
```

```
|        Health       area
|   1189   3950 South Asia
|   1190   3960 South Asia
|   1191   3970 South Asia
|   1192   3980 South Asia
```

```
|   1193   3990 South Asia
|   1194   4000 South Asia
```

```
dim(pred)
```

```
|   [1] 1194    2
```

```
head(with(pred, eval(ex)))
```

```
|         [,1] [,2]       [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,
|   [1,]    0    1 0.03333333    0    0    0    0    0    0      0
|   [2,]    0    1 0.02500000    0    0    0    0    0    0      0
|   [3,]    0    1 0.02000000    0    0    0    0    0    0      0
|   [4,]    0    1 0.01666667    0    0    0    0    0    0      0
|   [5,]    0    1 0.01428571    0    0    0    0    0    0      0
|   [6,]    0    1 0.01250000    0    0    0    0    0    0      0
```

```
ww <- wald(fit2, with(pred, eval(ex)))
str(ww)
```

```
|   List of 1
```

```
|    $ :List of 7
|     ..$ anova   :List of 4
|     .. ..$ numDF  : int 6
|     .. ..$ denDF  : int 177
|     .. ..$ F-value: num [1, 1] 33.2
|     .. ..$ p-value: num [1, 1] 1.3e-26
|     ..$ estimate:Classes 'data.frame.lab' and 'data.frame':   1194
|     .. ..$ Estimate  : num [1:1194] -0.00086 0.00236 0.00429 0.005
|     .. ..$ Std.Error : num [1:1194] 0.0424 0.0306 0.0236 0.0189 0.
|     .. ..$ DF        : num [1:1194] 177 177 177 177 177 177 177 17
|     .. ..$ t-value   : num [1:1194] -0.0203 0.0772 0.1822 0.2954 0
|     .. ..$ p-value   : num [1:1194] 0.984 0.939 0.856 0.768 0.677
|     .. ..$ Lower 0.95: num [1:1194] -0.0844 -0.058 -0.0422 -0.0317
|     .. ..$ Upper 0.95: num [1:1194] 0.0827 0.0628 0.0508 0.0429 0.
|     .. ..- attr(*, "labs")= chr [1:2] "" ""
|     ..$ coef    : num [1:1194] -0.00086 0.00236 0.00429 0.00558 0.
|     ..$ L       : num [1:1194, 1:13] 0 0 0 0 0 0 0 0 0 0 ...
|     ..$ se      : num [1:1194] 0.0424 0.0306 0.0236 0.0189 0.0156
|     ..$ L.full  : num [1:6, 1:13] 0 0 0 0 0 ...
```

```
|     ..$ L.rank  : int 6
|    - attr(*, "class")= chr "wald"
```

```
head(as.data.frame(ww))
```

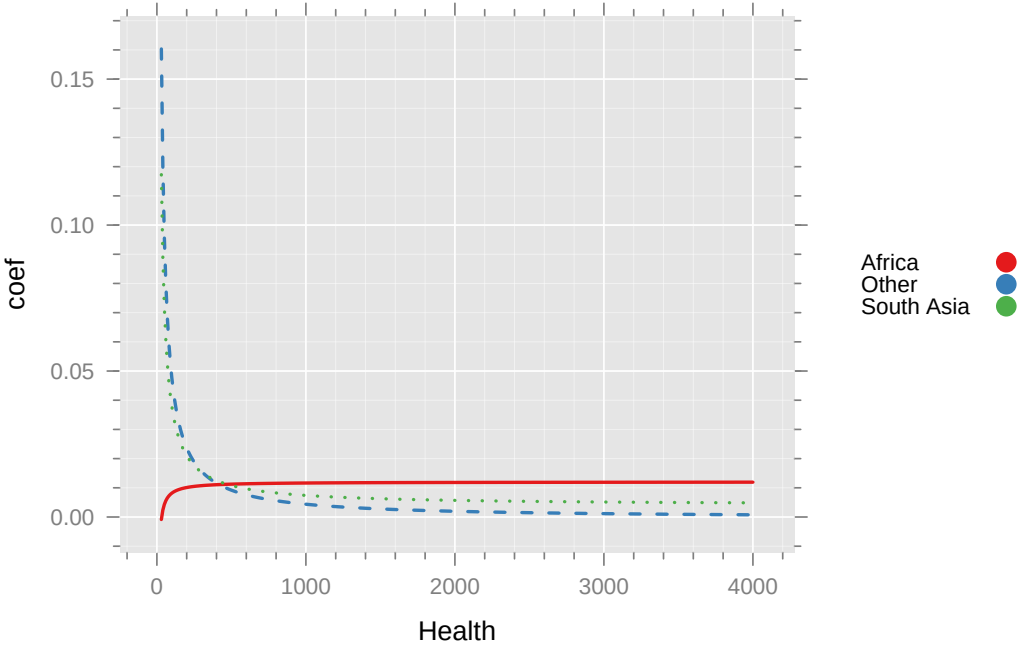
```
|              coef         se         U2          L2
|   1 -0.0008597286 0.04235567 0.08385161 -0.08557107
|   2  0.0023619380 0.03061080 0.06358353 -0.05885966
|   3  0.0042949379 0.02357817 0.05145127 -0.04286140
|   4  0.0055836046 0.01890309 0.04338979 -0.03222258
|   5  0.0065040807 0.01557660 0.03765728 -0.02464912
|   6  0.0071944378 0.01309439 0.03338321 -0.01899433
```

```
pred <- cbind( pred, as.data.frame(ww))
head(pred)
```

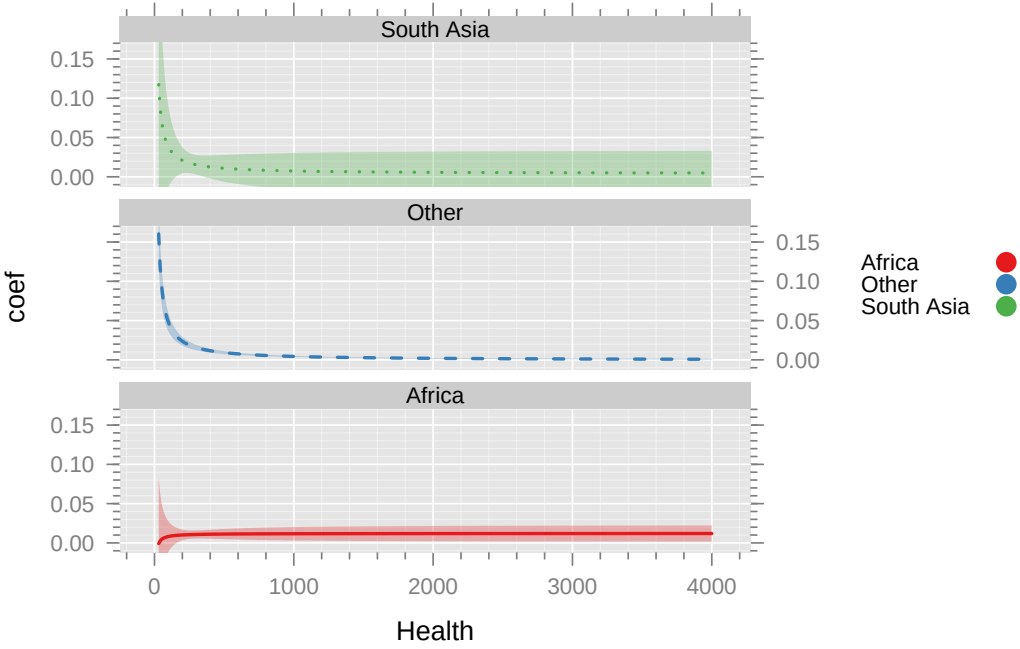
```
|     Health   area          coef         se         U2          L2
|   1     30 Africa -0.0008597286 0.04235567 0.08385161 -0.08557107
|   2     40 Africa  0.0023619380 0.03061080 0.06358353 -0.05885966
|   3     50 Africa  0.0042949379 0.02357817 0.05145127 -0.04286140
```

```
|   4     60 Africa  0.0055836046 0.01890309 0.04338979 -0.03222258
|   5     70 Africa  0.0065040807 0.01557660 0.03765728 -0.02464912
|   6     80 Africa  0.0071944378 0.01309439 0.03338321 -0.01899433
```

```
gd(lwd=2)
xyplot(coef ~ Health, pred, groups = area,type= 'l',
       auto.key=list(space = 'right'))
```

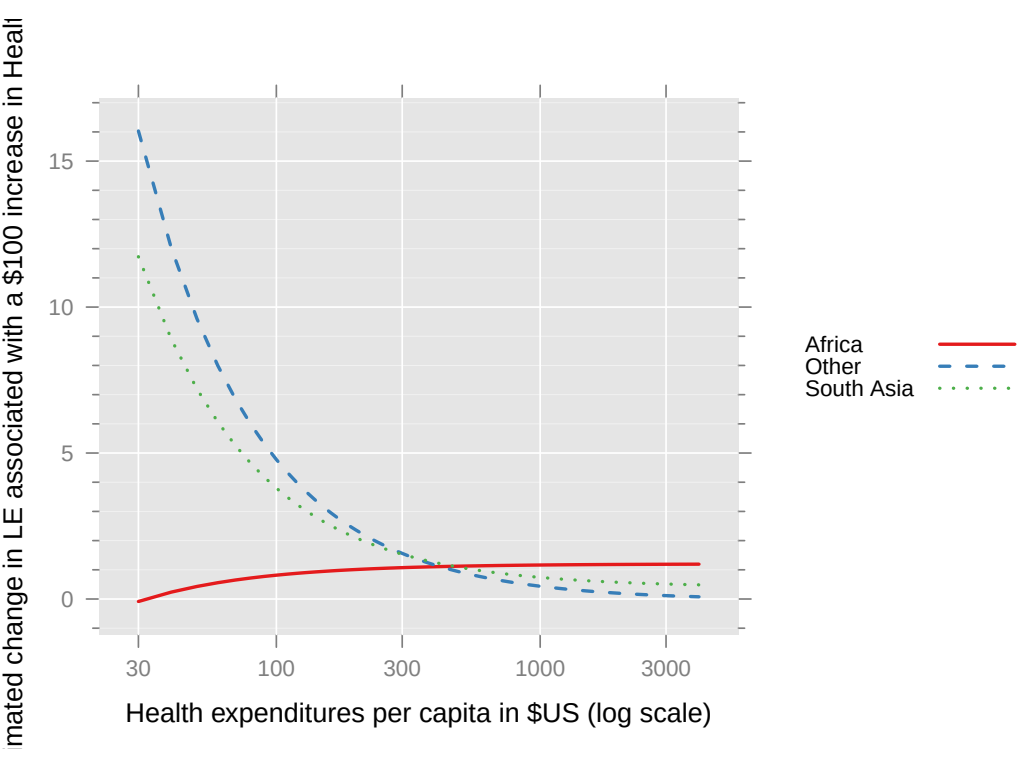

0.15
0.10
0.05
0.00
coef
0
1000
2000
3000
4000
Health
Africa
Other
South Asia

```
xyplot(coef ~ Health|area, pred, groups = area, type = 'l',
       auto.key = list(space='right'),
       subscripts = T,
       lower = pred$coef - 2* pred$se,
       upper = pred$coef + 2* pred$se,
       layout = c(1,3)) +
  glayer( gpanel.fit(...))
```


South Asia
Other
Africa
0.15
0.10
0.05
0.00
coef
0
1000
2000
3000
4000
Health
Africa
Other
South Asia

Make labels and axes interprettable for presentation:

```
xyplot(I(100*coef) ~ log(Health), pred, groups = area,type= 'l',
       auto.key=list(space='right', lines = T, points = F),
       ylab =
"Estimated change in LE associated with a $100 increase in Health Ex
       xlab = "Health expenditures per capita in $US (log scale)",
       scales = list( x = list(
         at = log(c(30,100,300,1000,3000,8000)),
         labels = c(30,100,300,1000,3000,8000))))
```


mated change in LE associated with a $100 increase in Healt
15
10
5
0
30
100
300
1000
3000
Health expenditures per capita in $US (log scale)
Africa
Other
South Asia

Limit plot to ranges in each Area:

1. create a small data set with ranges

```
dsr <- ds
```

1. max within each area

```
dsr$max <- with(dsr, capply( Health, area, max, na.rm = T))
```

2. min withing each area

```
dsr$min <- with(dsr, capply( Health, area, min, na.rm = T))
```

3. summary data frame with variables that are 'area invariant'

```
dsr <- up(dsr, ~ area)  # keeps 'area' invariant variables only
dsr
```

```
|               sex       area     max   min
|   Africa     BTSX     Africa 1642.71 16.99
|   Other      BTSX      Other 8607.88 50.47
|   South Asia BTSX South Asia  759.66 27.86
```

4. merge back into pred

```
predr <- merge(pred, dsr[,c('area','max','min')], all.x = T)
```

5. keep values of Health that are within range

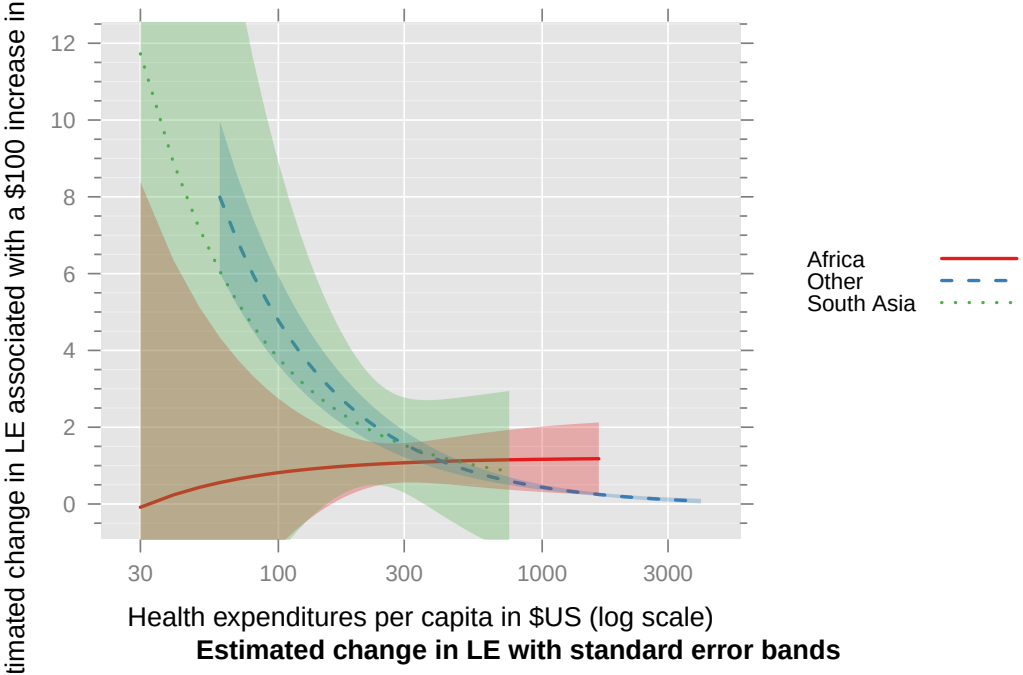
```
predr <- subset( predr, (Health <= max) & ( Health >= min))
head(predr)
```

	area	Health	coef	se	U2	L2
1	Africa	30	-0.0008597286	0.04235567	0.08385161	-0.08557107
2	Africa	40	0.0023619380	0.03061080	0.06358353	-0.05885966
3	Africa	50	0.0042949379	0.02357817	0.05145127	-0.04286140
4	Africa	60	0.0055836046	0.01890309	0.04338979	-0.03222258
5	Africa	70	0.0065040807	0.01557660	0.03765728	-0.02464912
6	Africa	80	0.0071944378	0.01309439	0.03338321	-0.01899433

4.7.2 Plotting fitted values and bands

If you define arguments **fit**, **lower** and **upper** in 'xyplot', they will be available to 'gpanel.fit' to draw the fitted line and confidence or predictions bands.

```
xyplot(I(100*coef) ~ log(Health), predr, groups = area,type= 'l',
       auto.key=list(space='right', lines = T, points = F),
       ylab =
"Estimated change in LE associated with a $100 increase in Health Ex
       lower = 100*(predr$coef - 2* predr$se),
       upper = 100*(predr$coef + 2* predr$se),
       sub= "Estimated change in LE with standard error bands",
       xlab = "Health expenditures per capita in $US (log scale)",
       scales = list( x = list(
         at = log(c(30,100,300,1000,3000,8000)),
         labels = c(30,100,300,1000,3000,8000)))) +
    glayer(gpanel.fit(...))
```


timated change in LE associated with a $100 increase in Hea
12
10
8
6
4
2
0
30
100
300
1000
3000
Health expenditures per capita in $US (log scale)
Africa
Other
South Asia
Estimated change in LE with standard error bands

```r
xyplot(I(100*coef) ~ log(Health)|area, predr, groups = area,
       type = 'l',
       auto.key=list(space='right', lines = T, points = F),
       ylab = "Additional years of Life Expectancy",
       xlim = c(-5,15),
       layout = c(1,3),
       lower = 100*(predr$coef - predr$se),
       upper = 100*(predr$coef + predr$se),
       xlab = "Health expenditures per capita in $US (log scale)",
       scales = list( x = list(
         at = log(c(30,100,300,1000,3000,8000)),
         labels = c(30,100,300,1000,3000,8000)))) +
  glayer(gpanel.fit(...))
```

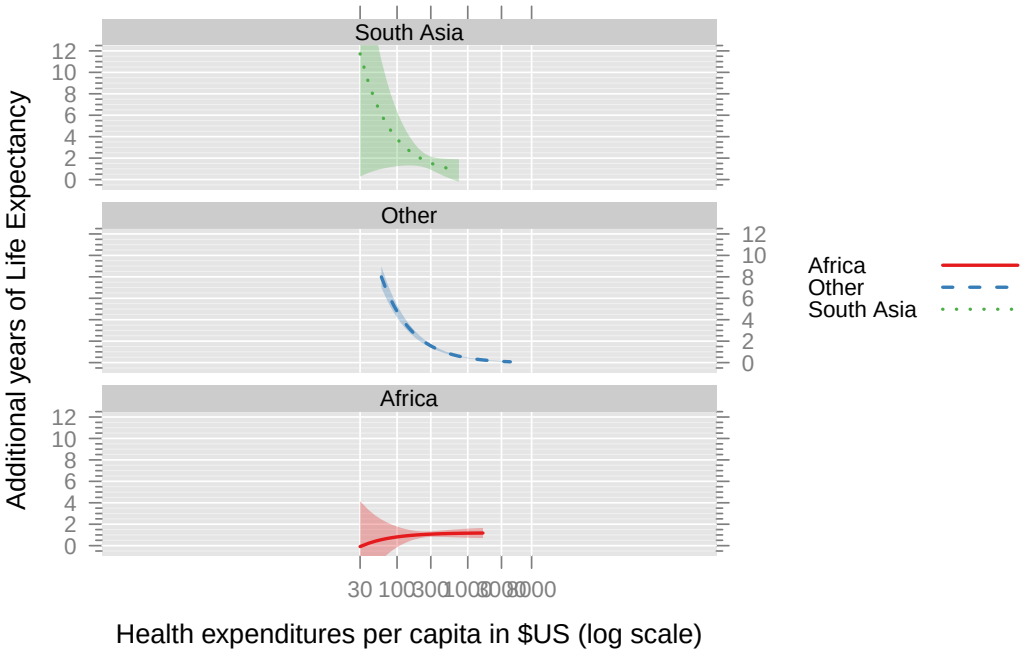

South Asia
Other
Africa
Additional years of Life Expectancy
0
2
4
6
8
10
12
30
100
300
1000
3000
8000
Health expenditures per capita in $US (log scale)
Africa
Other
South Asia

Figure: Additional years of life expectancy associated with a $100 (U.S.) increase in health expenditures per capita per year, in three world regions as a function of the current level of health expenditures. The bands show the standard error of estimation.

4.7.2.1 Exercises

1. Redo the above plot showing relationship of LE with a 1% increase in Health Expenditures.
2. What happens if you introduce the possibility of interaction between health expenditures and *propGovt*?
3. How do things change if we do a regression that gives more weight to larger countries? How should we do this?
4. Compare Africa and South Asia: Is there evidence of a difference between LE adjusted for Health Expenditures and propGovt. Prepare an appropriate plot.
5. Same for South Asia and "Other".

4.7.3 In the future:

We will explore the 'Lfx' function in 'spida2' and the 'sc' function for generalized splines generated by the 'gsp' function.

5 Appendices

5.1 Notes on the Principle of Marginality

The principle of marginality (POM) came up in class recently and it occurred to me that there might be confusion arising from the distinction between the requirements for:

1) A model to satisfy the POM, and

2) A null hypothesis specified by setting a set of terms to zero to satisfy the POM.

In a linear model with main effects and interactions of various orders, a model satisfies the POM if, for any interaction in the model, all included lower-order

interactions and main effects are also in the model. In other words, the model is closed under taking *margins* where we think of A:B as a margin of A:B:C, for example. Note that the intercept is considered a 0-th order effect and must be included if anything else is included.

A hypothesis that sets some parameters to 0 in a model that satisfies the POM, itself satisfies the POM provided the resulting H_0 model also satisfies the POM.

That's why the requirements for the set of terms that are set to 0 seems to be the reverse of the requirements for a model. For any given term set to zero in the hypothesis, all higher-order terms in the model that include the given term must also be set to zero. Otherwise, the null model would include those higher-order terms without including the given term which would result in a null hypothesis that violates the POM.

Thus the requirement for the set of terms set to zero is the 'reverse' of the requirement for models. The set of terms set to zero must be closed under taking interactions that are in the full model.

Note that the main significance of the POM is that predicted values ($\hat{Y}$) for a model that satisfies the POM are invariant under location-scale transformations

of numerical variables and non-singular recodings of categorical variables.

Hypotheses that satisfy the POM result in a test statistic, hence p-value, that is invariant under those same transformations and recodings.

So, if you stick to models and hypotheses that satisfy the POM, you don't need to worry that your conclusions would have been different if you had measured temperature in degrees Celsius instead of Farenheit or whether you had used a different category as a reference level for a factor.

Wald tests performed by specifying a regular expression to be matched in a model's terms will usually satisfy the POM because, if a regular expression matches a term, it will also match higher-order terms that contain that term.

This is true for tests of interactions with e.g. `wald(fit, ":")` for all interactions or `wald(fit, ':.*:")` for two-way and higher-order interactions, and for test of any particular effect, e.g. `wald(fit, 'X')` provided one checks that the regular expression does not match unintended terms.

Of course, you will be interested in estimating many parameters whose values do depend on the units used and the reference level. But those will be parameters

addressing specific questions. For example in a model Y ~ X*G where X is continuous and G has three levels: A, B and C, you may want to estimate the difference in the rate of change with respect to X (which does depend on the units of X) comparing levels C and B. The exact hypothesis to test whether this is a particular value will not satisfy the POM because the estimate will depend on the units of X and the coding of the factor G. This is okay because it obeys the principle of "you know what you are doing". The general rule is:

Never violate the POM unless you know what you are doing!

References

Fox, John, and Jangman Hong. 2009. "Effect Displays in R for Multinomial and Proportional-Odds Logit Models: Extensions to the effects Package." *Journal of Statistical Software* 32 (1): 1–24. http://www.jstatsoft.org/v32/i01/.

Monette, Georges, John Fox, Michael Friendly, and Heather Krause. 2018. *Spida2: Collection of Tools Developed for the Summer Programme in Data Analysis 2000-2012*.

Murnane, Richard J, and John B Willett. 2010. *Methods Matter: Improving Causal Inference in Educational and Social Science Research.* Oxford University Press.

Pearl, Judea, and Dana Mackenzie. 2018. *The Book of Why: The New Science of Cause and Effect.* Basic Books.