Bivariate Regression Analysis

Yalin Yang

2020-04-28

Contents

Scatter Plot & Effect Plots

Initialize Data

Relevant variables:

- price: median home price in community
- crime: crime rate
- nox: nitrogen oxide in the air
- dist: weighted distance to five employment centers
- rooms: average number of in houses in the community
- stratio: Student-teacher ratio of schools in the community
- proptax: property tax in community per $1000 home value

```
library(car); library(effects)
hprice2 <- foreign::read.dta("http://fmwww.bc.edu/ec-p/data/wooldridge/hprice2.dta")
summary(hprice2)
```

```
##      price           crime              nox           rooms
##  Min.   : 5000   Min.   : 0.0060   Min.   :3.85   Min.   :3.560
##  1st Qu.:16850   1st Qu.: 0.0820   1st Qu.:4.49   1st Qu.:5.883
##  Median :21200   Median : 0.2565   Median :5.38   Median :6.210
##  Mean   :22512   Mean   : 3.6115   Mean   :5.55   Mean   :6.284
##  3rd Qu.:24999   3rd Qu.: 3.6770   3rd Qu.:6.24   3rd Qu.:6.620
##  Max.   :50001   Max.   :88.9760   Max.   :8.71   Max.   :8.780
##       dist            radial          proptax         stratio
##  Min.   : 1.130   Min.   : 1.000   Min.   :18.70   Min.   :12.60
##  1st Qu.: 2.100   1st Qu.: 4.000   1st Qu.:27.90   1st Qu.:17.40
##  Median : 3.210   Median : 5.000   Median :33.00   Median :19.10
##  Mean   : 3.796   Mean   : 9.549   Mean   :40.82   Mean   :18.46
##  3rd Qu.: 5.188   3rd Qu.:24.000   3rd Qu.:66.60   3rd Qu.:20.20
##  Max.   :12.130   Max.   :24.000   Max.   :71.10   Max.   :22.00
##     lowstat          lprice             lnox          lproptax
##  Min.   : 1.730   Min.   : 8.517   Min.   :1.348   Min.   :5.231
##  1st Qu.: 6.923   1st Qu.: 9.732   1st Qu.:1.502   1st Qu.:5.631
##  Median :11.360   Median : 9.962   Median :1.683   Median :5.799
##  Mean   :12.701   Mean   : 9.941   Mean   :1.693   Mean   :5.931
##  3rd Qu.:17.058   3rd Qu.:10.127   3rd Qu.:1.831   3rd Qu.:6.501
##  Max.   :39.070   Max.   :10.820   Max.   :2.164   Max.   :6.567
```

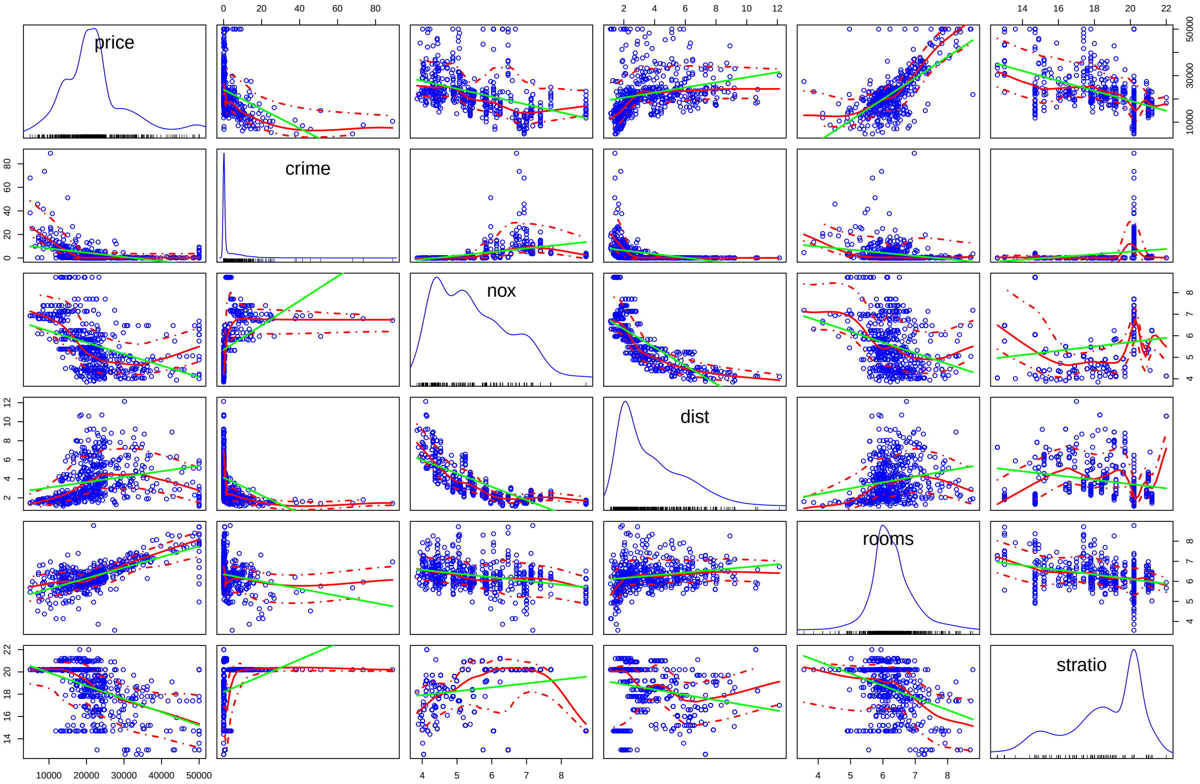
Check normality

```r
summary(powerTransform(cbind(price,crime,nox,dist,rooms,stratio)~1, data=hprice2))
```

```
## bcPower Transformations to Multinormality
##         Est Power Rounded Pwr Wald Lwr Bnd Wald Upr Bnd
## price      0.3971        0.50       0.2595       0.5348
## crime     -0.1025       -0.10      -0.1378      -0.0672
## nox       -1.2281       -1.00      -1.5282      -0.9280
## dist      -0.0798        0.00      -0.2024       0.0427
## rooms      1.1678        1.00       0.7340       1.6017
## stratio    4.5471        4.55       3.7763       5.3179
##
## Likelihood ratio test that transformation parameters are equal to 0
##  (all log transformations)
##                                       LRT df       pval
## LR test, lambda = (0 0 0 0 0 0) 298.4363  6 < 2.22e-16
##
## Likelihood ratio test that no transformations are needed
##                                       LRT df       pval
## LR test, lambda = (1 1 1 1 1 1) 3375.716  6 < 2.22e-16
```

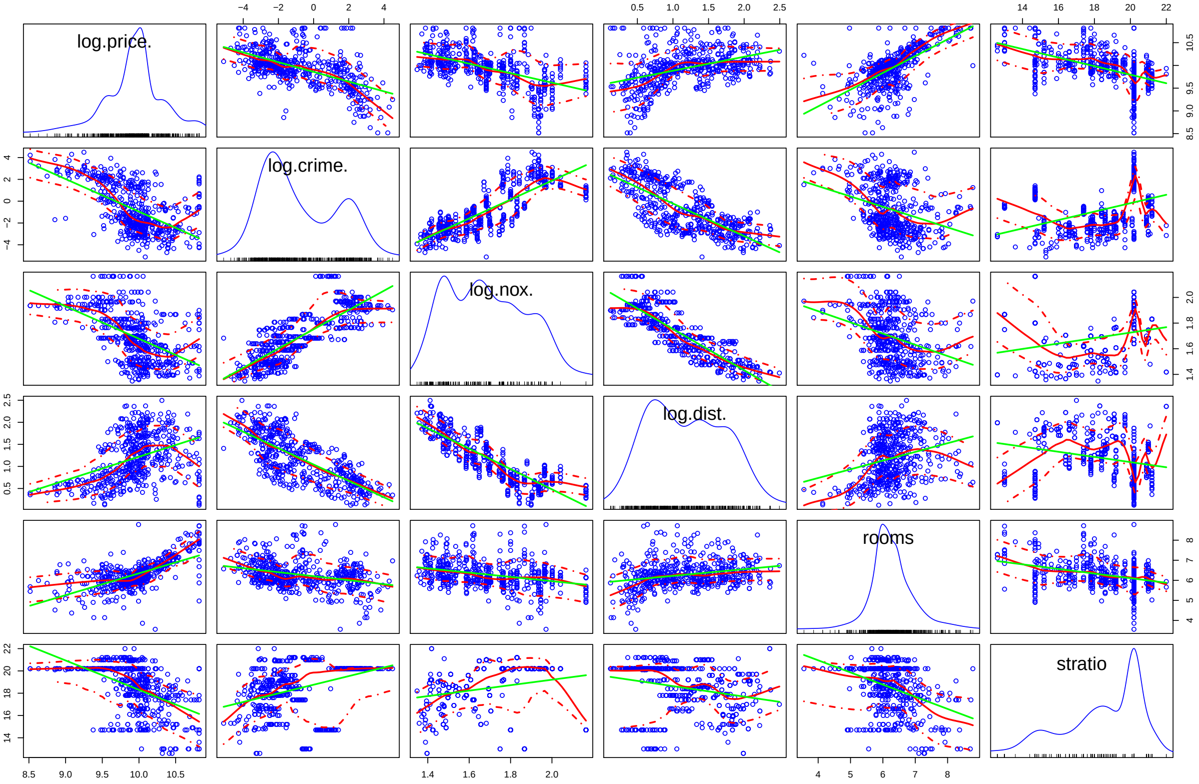
Scatter Plot

```r
scatterplotMatrix(~price+crime+nox+dist+rooms+stratio, data=hprice2, pch=1,
                  smooth=list(span = 0.35,lty.smooth=1, col.smooth="red", col.var="red"),
                  regLine=list(col="green"))
```



Applied log-transformation on highly positive skewed variables

```
scatterplotMatrix(~log(price)+log(crime)+log(nox)+log(dist)+rooms+stratio,
                  data=hprice2, pch=1,
                  smooth=list(span = 0.35,lty.smooth=1, col.smooth="red", col.var="red"),
                  regLine=list(col="green"))
```



Regression modeling on untransformed data

Plain Model

```
mod0 <- lm(price~nox+dist+rooms+stratio, data=hprice2)
summary(mod0)
```

```
##
## Call:
## lm(formula = price ~ nox + dist + rooms + stratio, data = hprice2)
##
## Residuals:
##    Min      1Q Median      3Q     Max
## -14310  -3124   -546   2181  38580
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  23716.2     5120.6   4.632 4.63e-06 ***
## nox          -3044.9      353.7  -8.609  < 2e-16 ***
## dist          -965.5      191.5  -5.042 6.45e-07 ***
## rooms         6808.8      401.4  16.964  < 2e-16 ***
## stratio      -1269.2      127.4  -9.965  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 5701 on 501 degrees of freedom
## Multiple R-squared:  0.6198, Adjusted R-squared:  0.6168
## F-statistic: 204.2 on 4 and 501 DF,  p-value: < 2.2e-16
```

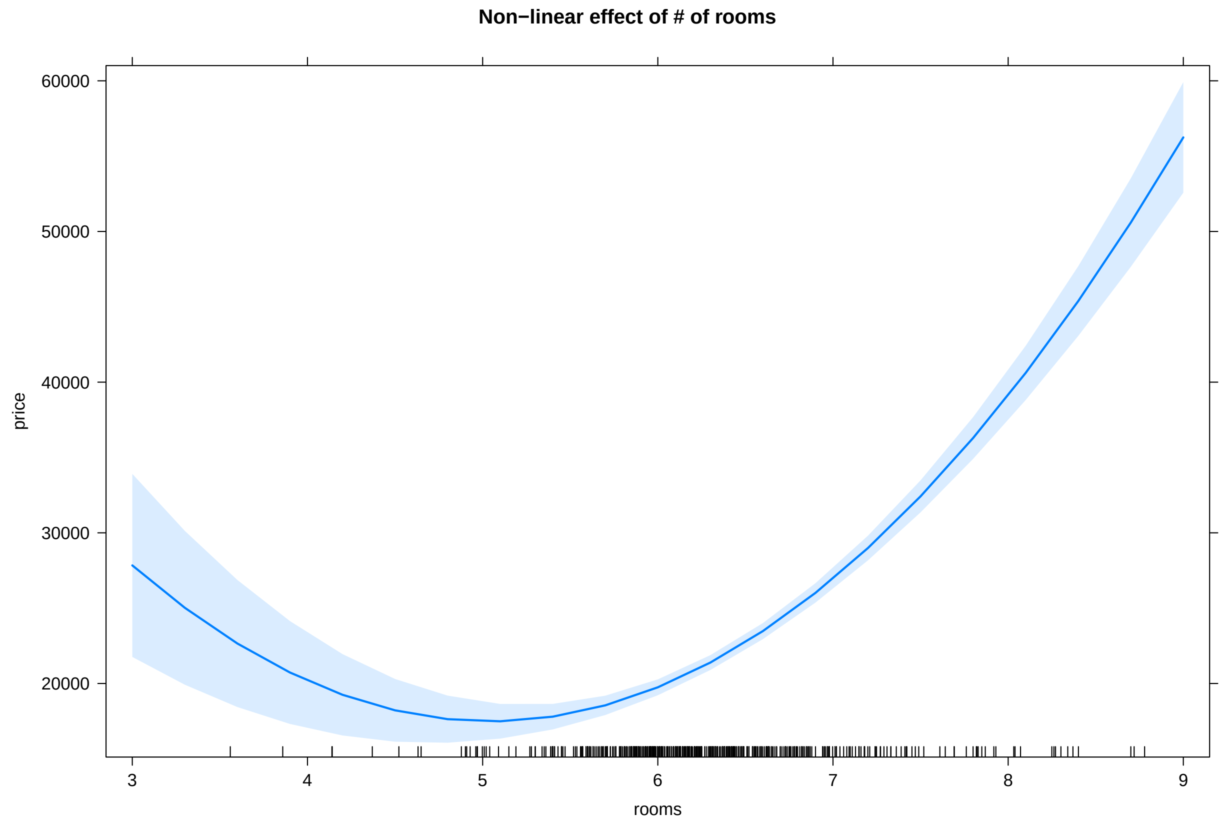
Quadratic term (Could use turkey test)

```
mod0 <- lm(price~nox+dist+rooms+I(rooms^2)+stratio, data=hprice2)
summary(mod0)
```

```
##
## Call:
## lm(formula = price ~ nox + dist + rooms + I(rooms^2) + stratio,
##     data = hprice2)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -24609  -2831   -225   2167  34950
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 120386.8    10964.2  10.980  < 2e-16 ***
## nox          -3086.5      324.5  -9.511  < 2e-16 ***
## dist          -723.5      177.4  -4.078 5.29e-05 ***
## rooms       -24993.1     3279.8  -7.620 1.28e-13 ***
## I(rooms^2)    2477.3      253.9   9.758  < 2e-16 ***
## stratio      -1082.9      118.4  -9.146  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5230 on 500 degrees of freedom
## Multiple R-squared:  0.6806, Adjusted R-squared:  0.6774
## F-statistic: 213.1 on 5 and 500 DF,  p-value: < 2.2e-16
```

Effect Plots

```
mod0.eff <- allEffects(mod0, xlevels=list(rooms=3:9))
plot(mod0.eff, "rooms", main="Non-linear effect of # of rooms")
```



Regression modeling on log-form data

Base model

```
mod1 <- lm(log(price)~log(crime)+log(nox)+log(dist)+rooms+stratio, data=hprice2)
summary(mod1)
```

```
##
## Call:
## lm(formula = log(price) ~ log(crime) + log(nox) + log(dist) +
##     rooms + stratio, data = hprice2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.93773 -0.12747  0.00152  0.11892  1.34566
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 10.352387   0.343160  30.168  < 2e-16 ***
## log(crime)  -0.048871   0.009734  -5.021 7.18e-07 ***
## log(nox)    -0.629195   0.131050  -4.801 2.09e-06 ***
## log(dist)   -0.166265   0.042576  -3.905 0.000107 ***
## rooms        0.253968   0.018099  14.032  < 2e-16 ***
## stratio     -0.042393   0.006098  -6.952 1.13e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2588 on 500 degrees of freedom
## Multiple R-squared:  0.604,  Adjusted R-squared:    0.6
## F-statistic: 152.5 on 5 and 500 DF,  p-value: < 2.2e-16
```

Quadratic effects

```
mod2 <- lm(log(price)~log(crime)+log(nox)+log(dist)+rooms+I(rooms^2)+stratio, data=hprice2)
summary(mod2)
```

```
##
## Call:
## lm(formula = log(price) ~ log(crime) + log(nox) + log(dist) +
##     rooms + I(rooms^2) + stratio, data = hprice2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.9092 -0.1188 -0.0013  0.1227  1.2647
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 12.907027   0.554213  23.289  < 2e-16 ***
## log(crime)  -0.056155   0.009519  -5.899 6.75e-09 ***
## log(nox)    -0.520840   0.128407  -4.056 5.79e-05 ***
## log(dist)   -0.115984   0.042180  -2.750  0.00618 **
## rooms       -0.671539   0.161558  -4.157 3.80e-05 ***
## I(rooms^2)   0.072055   0.012504   5.763 1.45e-08 ***
## stratio     -0.035268   0.006039  -5.841 9.39e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2509 on 499 degrees of freedom
## Multiple R-squared:  0.6287, Adjusted R-squared:  0.6242
## F-statistic: 140.8 on 6 and 499 DF,  p-value: < 2.2e-16
```

anova test

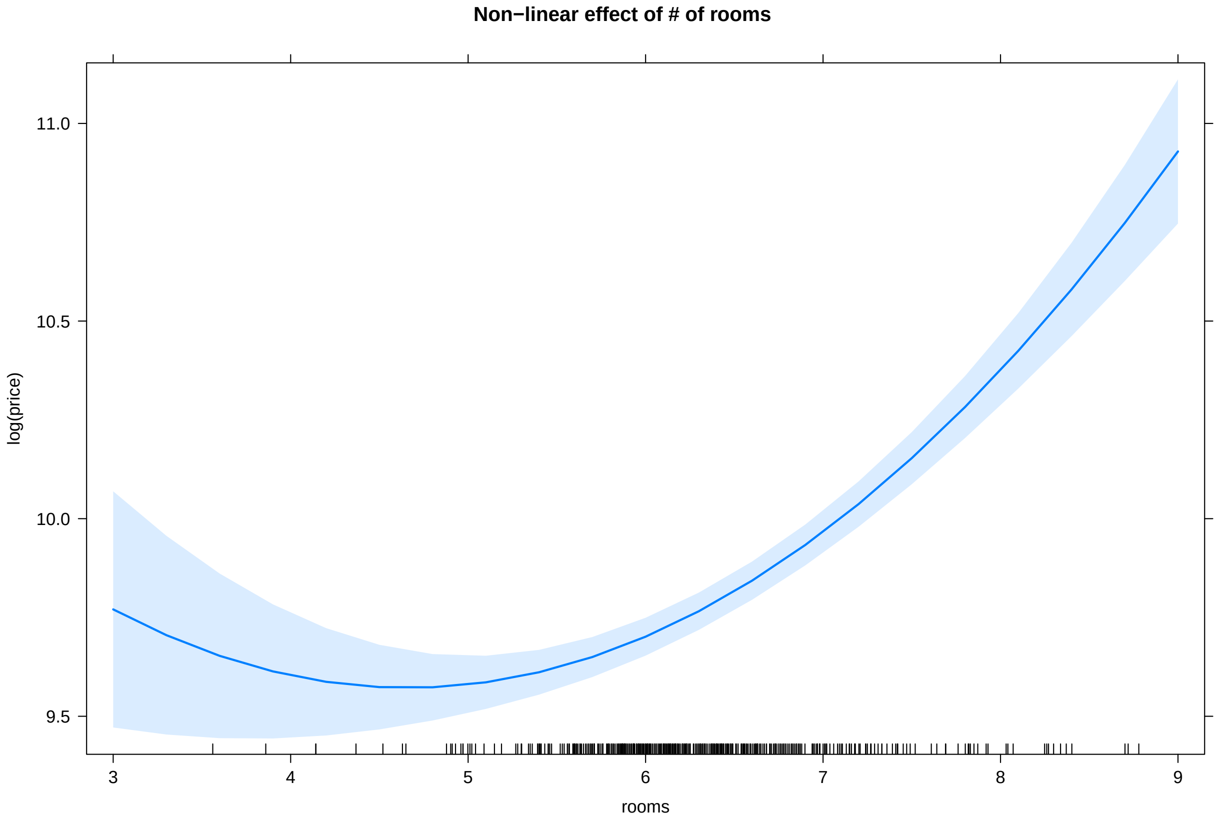
```
anova(mod1,mod2)
```

```
## Analysis of Variance Table
##
## Model 1: log(price) ~ log(crime) + log(nox) + log(dist) + rooms + stratio
## Model 2: log(price) ~ log(crime) + log(nox) + log(dist) + rooms + I(rooms^2) +
##     stratio
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    500 33.495
## 2    499 31.405  1      2.09 33.209 1.451e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Explore non-linear effect

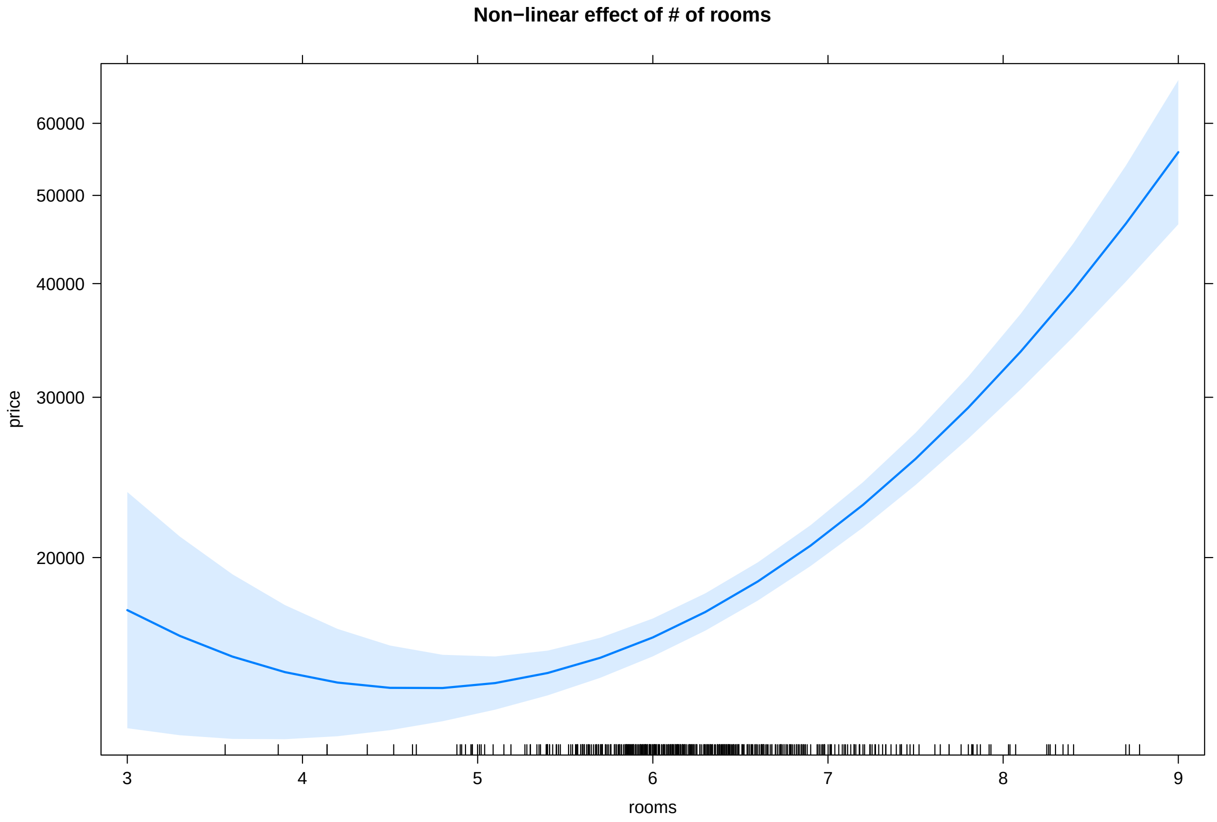
Turning point: $-\frac{b}{2a} => 0.672/(2 * 0.072) = 4.667$

```
mod2a.eff <- allEffects(mod2, xlevels=list(rooms=3:9))
plot(mod2a.eff, "rooms", ylab="log(price)", main="Non-linear effect of # of rooms")
```



Transfer dependent variable to the original scale

```
mod2b.eff <- allEffects(mod2, xlevels=list(rooms=3:9),
                        transformation=list(link=log, inverse=exp))
plot(mod2b.eff, "rooms", ylab="price", main="Non-linear effect of # of rooms")
```



Conditional Effect Plot

Initialize Data

Key variables: * stndfnl: Standardized outcome on final exam * atndrte: Percentage of class attendence * priGPA: Prior college grade point average * ACT: American College Testing score

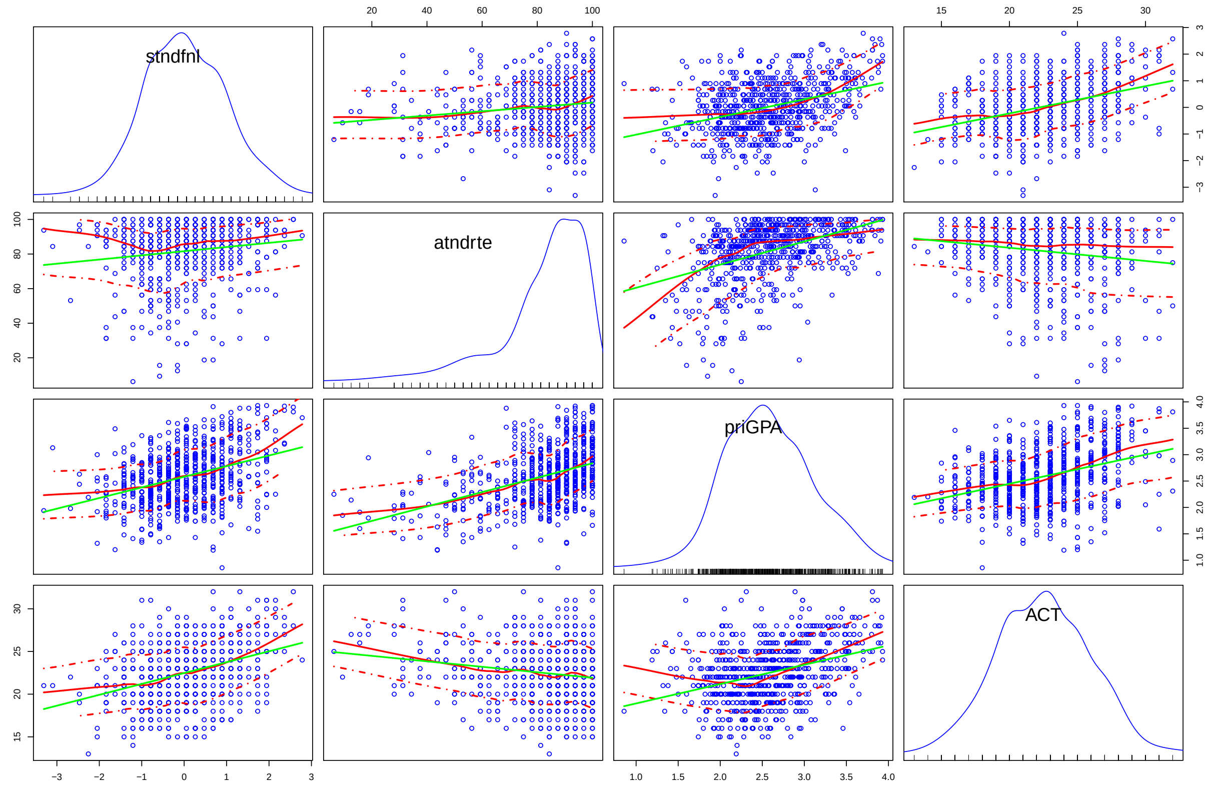
```
library(car); library(effects)
attend <- foreign::read.dta("http://fmwww.bc.edu/ec-p/data/wooldridge/attend.dta")
summary(attend)
```

```
##      attend         termgpa          priGPA           ACT
##  Min.   : 2.00   Min.   :0.000   Min.   :0.857   Min.   :13.00
##  1st Qu.:24.00   1st Qu.:2.138   1st Qu.:2.190   1st Qu.:20.00
##  Median :28.00   Median :2.670   Median :2.560   Median :22.00
##  Mean   :26.15   Mean   :2.601   Mean   :2.587   Mean   :22.51
##  3rd Qu.:30.00   3rd Qu.:3.120   3rd Qu.:2.942   3rd Qu.:25.00
##  Max.   :32.00   Max.   :4.000   Max.   :3.930   Max.   :32.00
##
##      final          atndrte          hwrte            frosh
##  Min.   :10.00   Min.   :  6.25   Min.   : 12.50   Min.   :0.0000
##  1st Qu.:22.00   1st Qu.: 75.00   1st Qu.: 87.50   1st Qu.:0.0000
##  Median :26.00   Median : 87.50   Median :100.00   Median :0.0000
##  Mean   :25.89   Mean   : 81.71   Mean   : 87.91   Mean   :0.2324
##  3rd Qu.:29.00   3rd Qu.: 93.75   3rd Qu.:100.00   3rd Qu.:0.0000
##  Max.   :39.00   Max.   :100.00   Max.   :100.00   Max.   :1.0000
##                                   NA's   :6
##       soph           skipped          stndfnl
##  Min.   :0.0000   Min.   : 0.000   Min.   :-3.30882
##  1st Qu.:0.0000   1st Qu.: 2.000   1st Qu.:-0.78782
##  Median :1.0000   Median : 4.000   Median : 0.05252
##  Mean   :0.5765   Mean   : 5.853   Mean   : 0.02966
##  3rd Qu.:1.0000   3rd Qu.: 8.000   3rd Qu.: 0.68277
##  Max.   :1.0000   Max.   :30.000   Max.   : 2.78361
##
```

Regression Modeling

Scatter Plot

```
scatterplotMatrix(~stndfnl+atndrte+priGPA+ACT, data=attend, pch=1,
                  smooth=list(span = 0.35,lty.smooth=1, col.smooth="red", col.var="red"),
                  regLine=list(col="green"))
```



Base model

```
mod1 <- lm(stndfnl~atndrte+priGPA+ACT, data=attend)
summary(mod1)
```

```
##
```

```
## Call:
## lm(formula = stndfnl ~ atndrte + priGPA + ACT, data = attend)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2339 -0.5528 -0.0329  0.5884  2.3303
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.343655   0.299098 -11.179  < 2e-16 ***
## atndrte      0.005334   0.002369   2.252   0.0247 *
## priGPA       0.402373   0.078280   5.140 3.60e-07 ***
## ACT          0.084257   0.011182   7.535 1.57e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8862 on 676 degrees of freedom
## Multiple R-squared:  0.2013, Adjusted R-squared:  0.1978
## F-statistic: 56.79 on 3 and 676 DF,  p-value: < 2.2e-16
```

With interaction. Notice the "*" in the formula

```
mod2 <- lm(stndfnl~atndrte*priGPA+ACT, data=attend)
summary(mod2)
```

```
##
## Call:
## lm(formula = stndfnl ~ atndrte * priGPA + ACT, data = attend)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2071 -0.5380 -0.0297  0.5852  2.3765
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -1.135889   0.793175  -1.432  0.15258
## atndrte        -0.020893   0.009047  -2.309  0.02122 *
## priGPA         -0.554498   0.328065  -1.690  0.09145 .
## ACT             0.081698   0.011149   7.328  6.7e-13 ***
## atndrte:priGPA  0.011462   0.003818   3.002  0.00278 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.881 on 675 degrees of freedom
## Multiple R-squared:  0.2118, Adjusted R-squared:  0.2072
## F-statistic: 45.35 on 4 and 675 DF,  p-value: < 2.2e-16
```

Partial F-test here equal to the t-test for the interaction term

```
anova(mod1,mod2)
```

```
## Analysis of Variance Table
```

```
##
## Model 1: stndfnl ~ atndrte + priGPA + ACT
## Model 2: stndfnl ~ atndrte * priGPA + ACT
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1    676 530.94
## 2    675 523.94  1    6.9971 9.0144 0.002778 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Conditional Effect Plot

Effect of attendence rate at **average priGPA**

```
(b <- coef(mod2))
```

```
##    (Intercept)        atndrte         priGPA            ACT atndrte:priGPA
##    -1.13588880    -0.02089258    -0.55449794     0.08169791     0.01146168
```

```
(mean(attend$priGPA))
```

```
## [1] 2.586775
```

```
cat("Partial effect of atndrte for priGPA=2.59:", b["atndrte"]+mean(attend$priGPA)*b["atndrte:priGPA"])
```

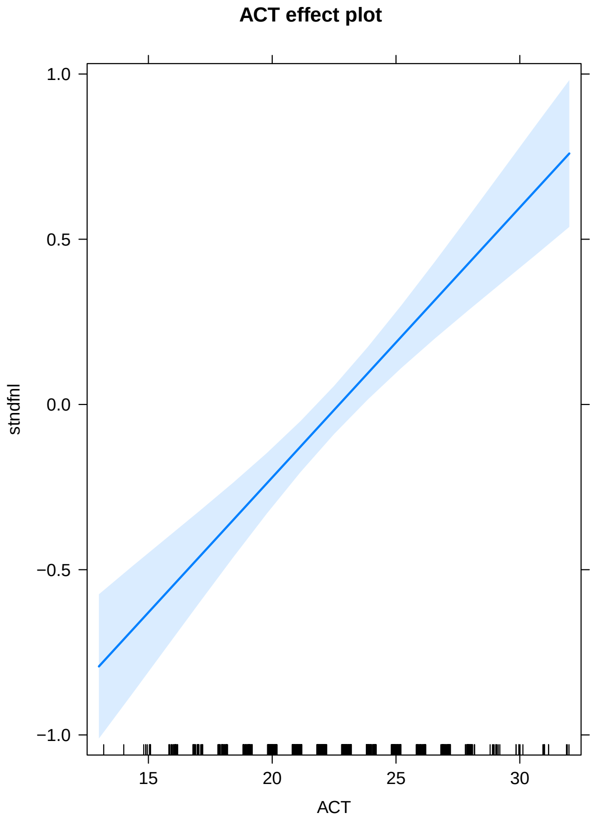
```
## Partial effect of atndrte for priGPA=2.59: 0.008756212
```

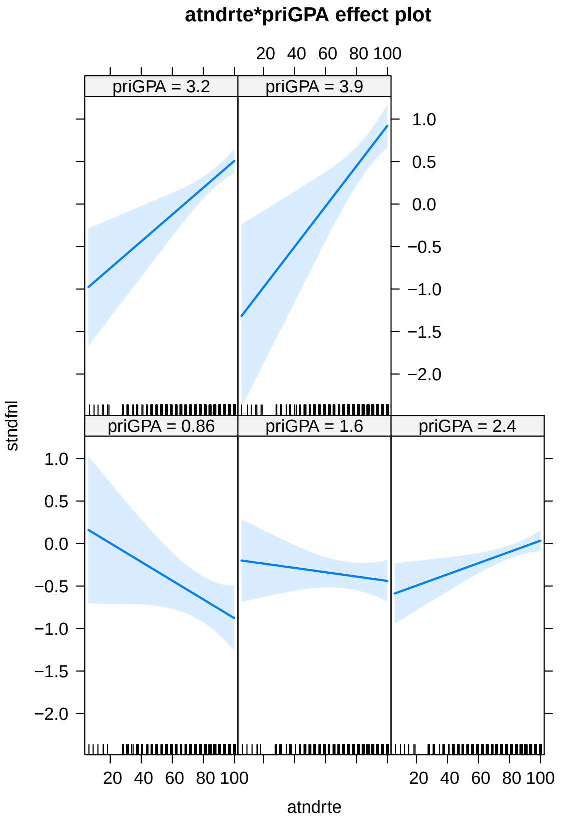
Test partial effect at priGPA=mean(attend$priGPA)

```
linearHypothesis(mod2, c("atndrte+2.59*atndrte:priGPA"))
```

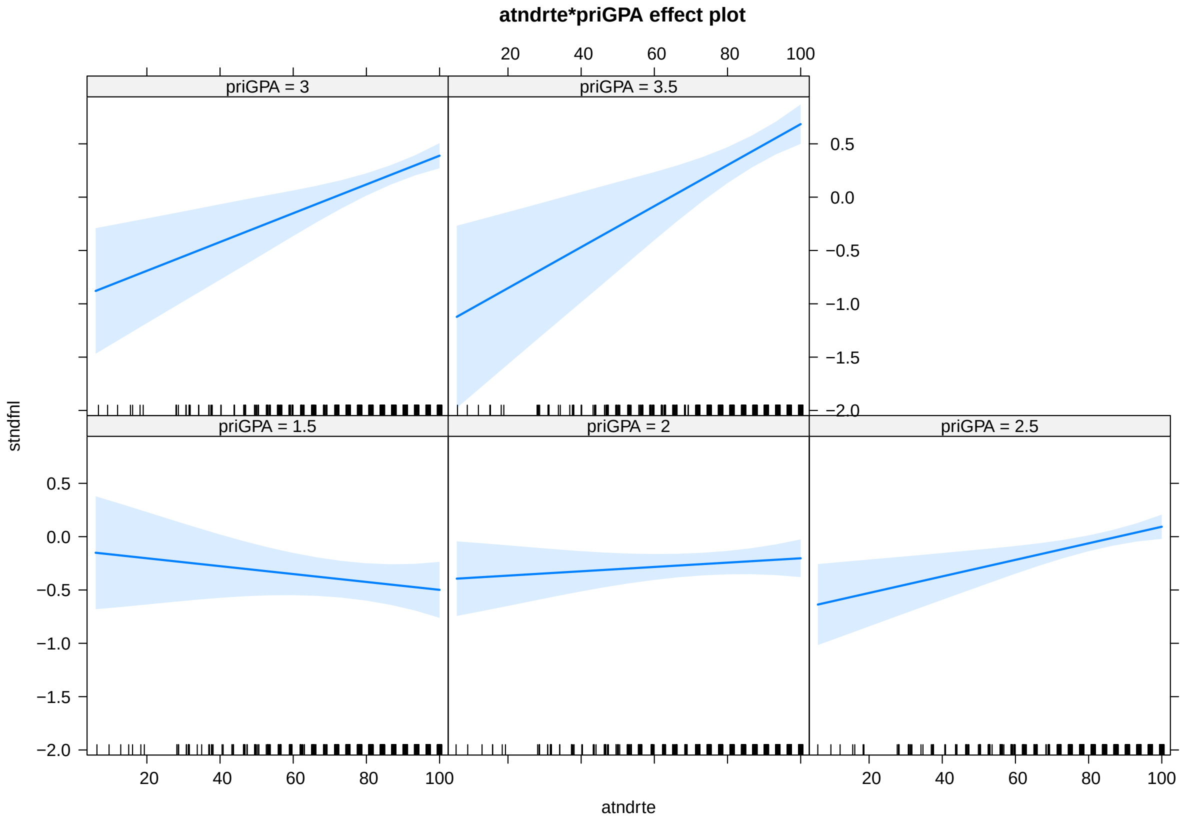
```
## Linear hypothesis test
##
## Hypothesis:
## atndrte  + 2.59 atndrte:priGPA = 0
##
## Model 1: restricted model
## Model 2: stndfnl ~ atndrte * priGPA + ACT
##
##   Res.Df    RSS Df Sum of Sq     F    Pr(>F)
## 1    676 532.68
## 2    675 523.94  1    8.7325 11.25 0.0008407 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
plot(allEffects(mod2))
```





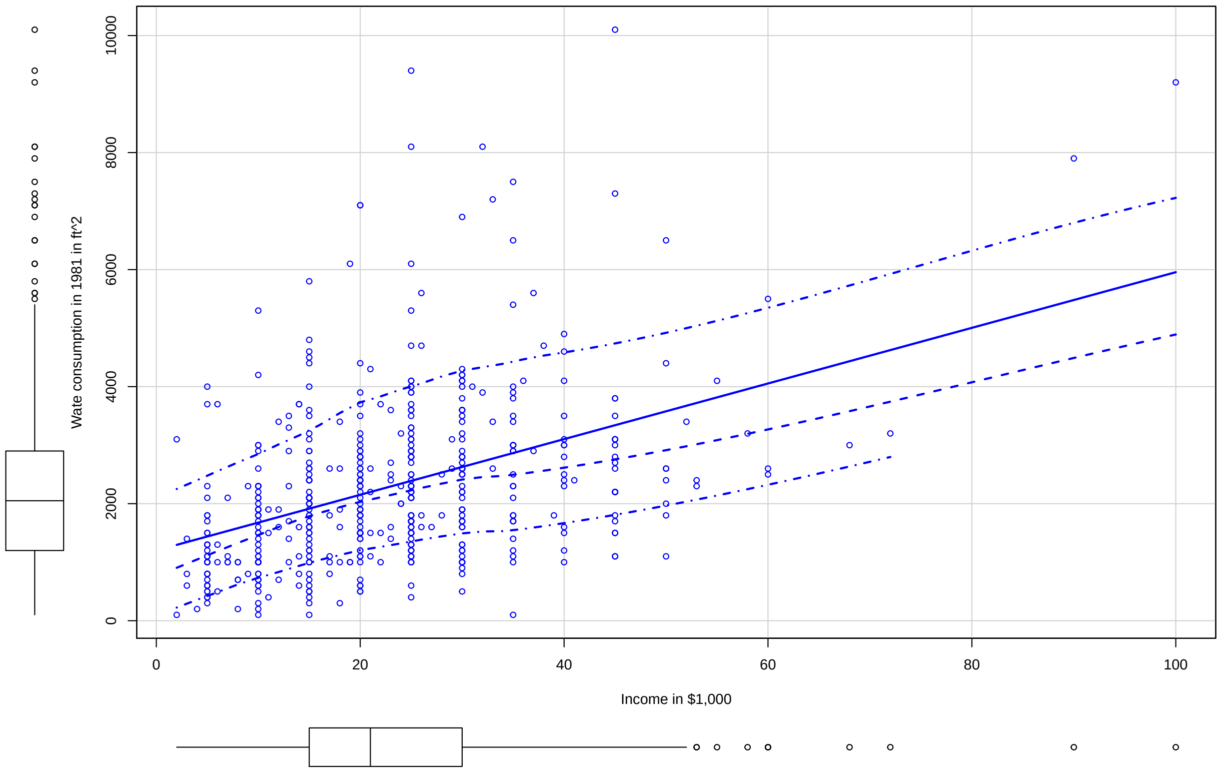
```r
plot(Effect(c("atndrte","priGPA"), mod2,
            xlevels=list(priGPA=seq(1.5, 3.5, by=0.5))))
```



Partial Effects Demo

Initialize Data

```
concord <- foreign::read.spss("Concord1.sav",to.data.frame=T)
car::scatterplot(water81~income, data=concord,
                 xlab="Income in $1,000",ylab="Wate consumption in 1981 in ft^2")
```



Demonstration: Partial Effects

```
mod1.lm <- lm(water81 ~ income, data=concord)
summary(mod1.lm)
```

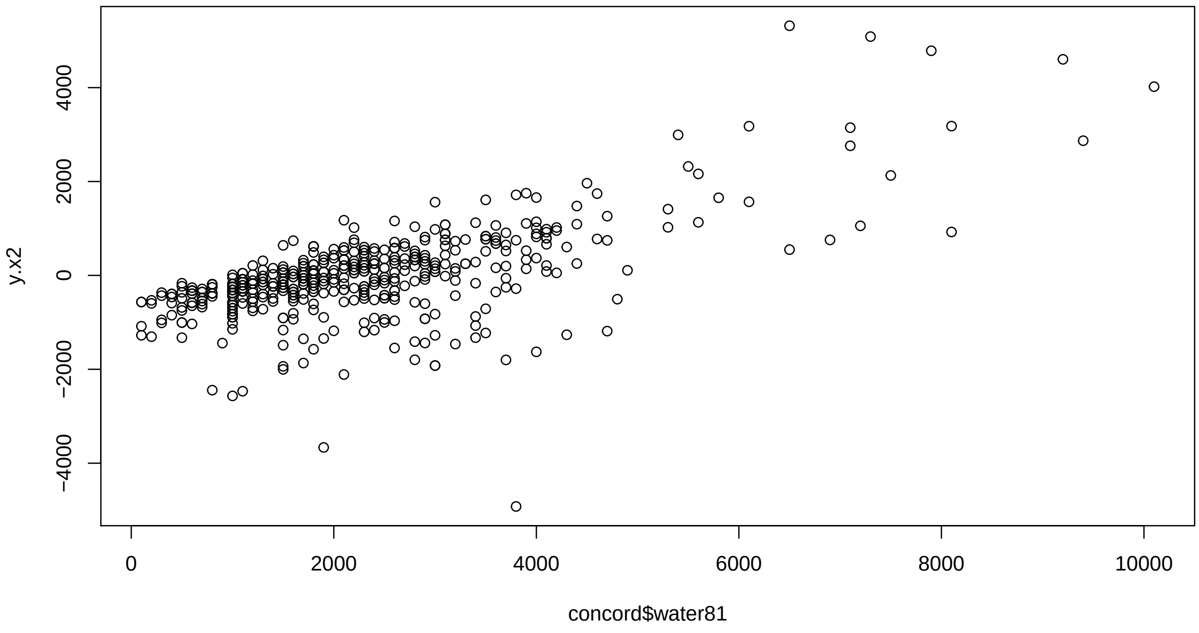
```
##
## Call:
## lm(formula = water81 ~ income, data = concord)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2765.3  -889.8  -239.8   536.8  7010.2
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1201.124    123.325    9.74   <2e-16 ***
## income        47.549      4.652   10.22   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1352 on 494 degrees of freedom
## Multiple R-squared:  0.1745, Adjusted R-squared:  0.1729
## F-statistic: 104.5 on 1 and 494 DF,  p-value: < 2.2e-16
```

```
mod2.lm <- lm(water81 ~ income + water80, data=concord)
summary(mod2.lm)
```

```
##
## Call:
## lm(formula = water81 ~ income + water80, data = concord)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4861.1  -439.5   -67.5   382.5  4984.0
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 203.82169   94.36129   2.160   0.0313 *
## income       20.54504    3.38341   6.072 2.52e-09 ***
## water80       0.59313    0.02505  23.679  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 925.4 on 493 degrees of freedom
## Multiple R-squared:  0.6138, Adjusted R-squared:  0.6122
## F-statistic: 391.8 on 2 and 493 DF,  p-value: < 2.2e-16
```

Controling for water80

```
y.x2 <- residuals(lm(water81 ~ water80, data=concord))
plot(concord$water81,y.x2)
```



```
cor(concord$water81,y.x2)
```

```
## [1] 0.6442743
```

```
cor(concord$water80,y.x2)
```

```
## [1] 2.637186e-16
```

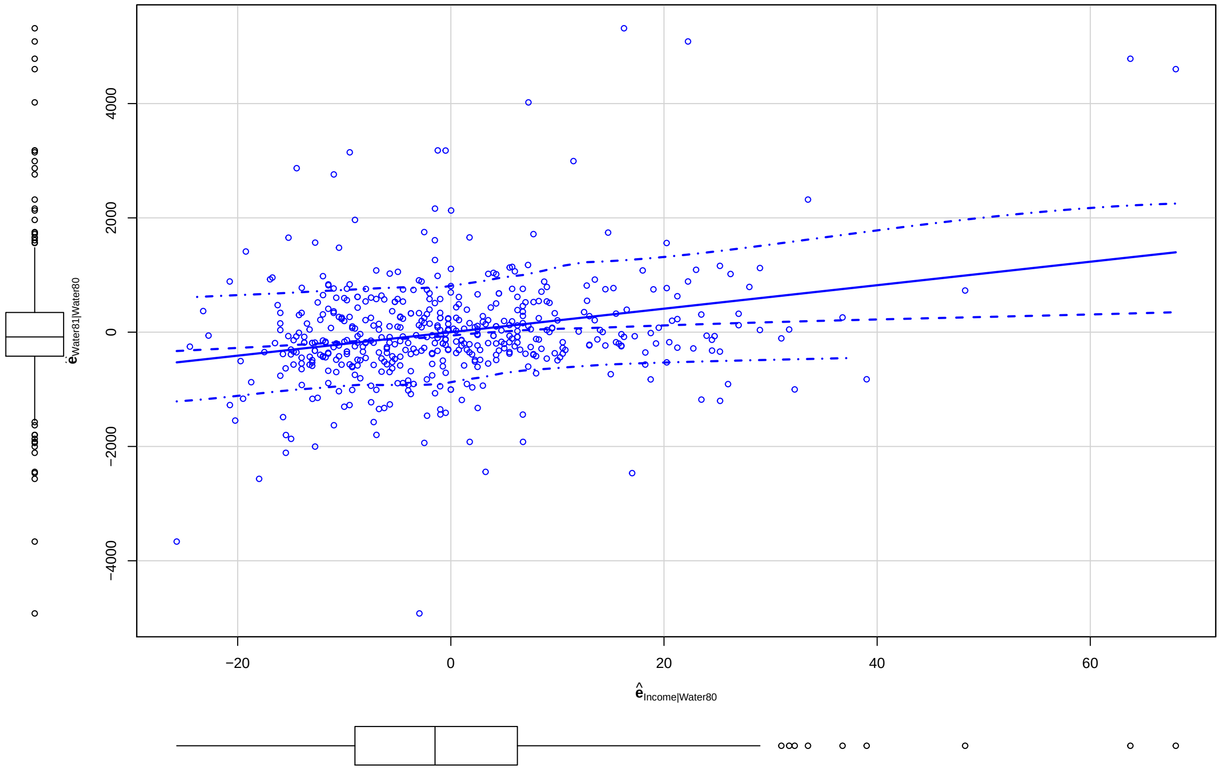
```
x1.x2 <- residuals(lm(income ~ water80, data=concord))
cat("Sum of residuals for Water81|Water80:",sum(y.x2),"\n")                    #sum of residuals = 0
```

```
## Sum of residuals for Water81|Water80: 5.421441e-12
```

```
cat("Sum of residuals for Income|Water80: ",sum(x1.x2),"\n")
```

```
## Sum of residuals for Income|Water80:  -4.060086e-13
```

```
scatterplot(y.x2~x1.x2, xlab=bquote(hat(bold(e))["Income|Water80"]),
            ylab=bquote(hat(bold(e))["Water81|Water80"]))
```



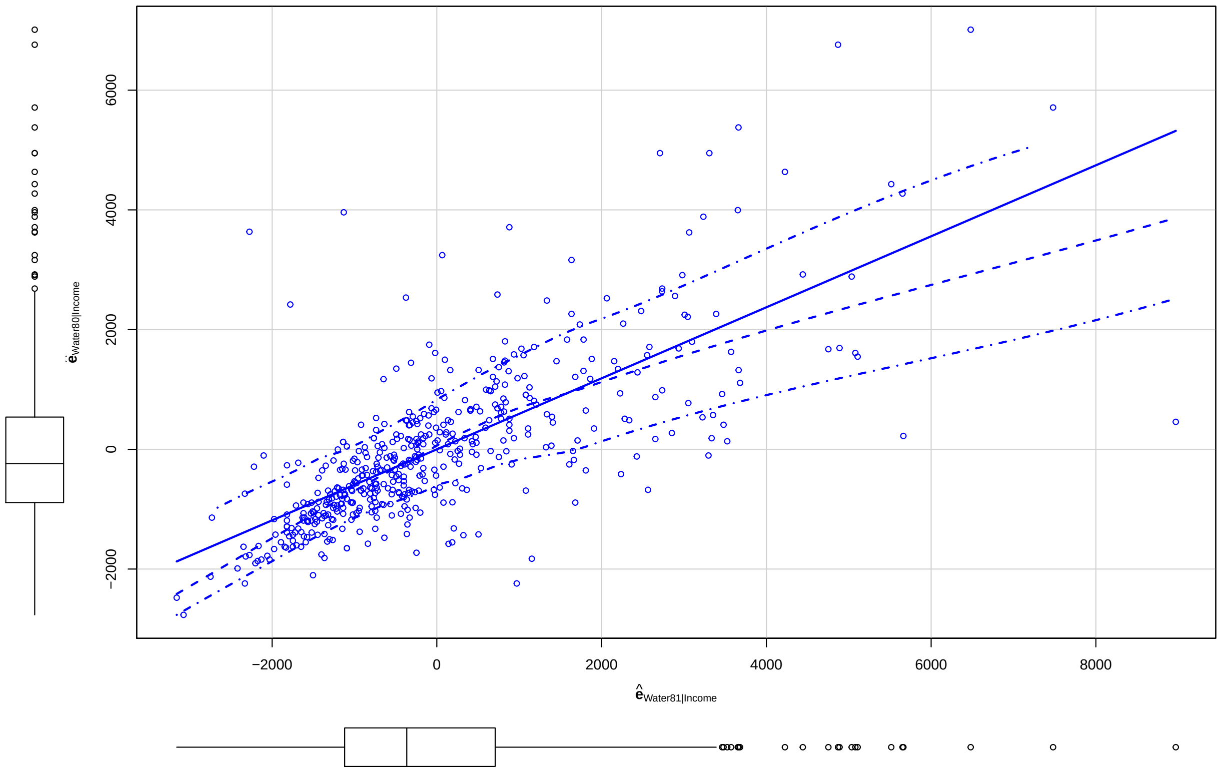
compare with coefficients of mod2.lm

```
summary(lm(y.x2 ~ x1.x2-1))
```

```
##
## Call:
## lm(formula = y.x2 ~ x1.x2 - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4861.1  -439.5   -67.5   382.5  4984.0
##
## Coefficients:
##       Estimate Std. Error t value Pr(>|t|)
## x1.x2   20.545      3.377   6.085 2.34e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 923.6 on 495 degrees of freedom
## Multiple R-squared:  0.06959,    Adjusted R-squared:  0.06771
## F-statistic: 37.02 on 1 and 495 DF,  p-value: 2.341e-09
```

Controling for income

```
y.x1 <- residuals(lm(water81 ~ income, data=concord))
x2.x1 <- residuals(lm(water80 ~ income, data=concord))
scatterplot(y.x1~x2.x1, xlab=bquote(hat(bold(e))["Water81|Income"]),
            ylab=bquote(hat(bold(e))["Water80|Income"]))
```



compare with coefficients of mod2.lm

```r
summary(lm(y.x1 ~ x2.x1-1))
```

```
##
## Call:
## lm(formula = y.x1 ~ x2.x1 - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4861.1  -439.5   -67.5   382.5  4984.0
##
## Coefficients:
##       Estimate Std. Error t value Pr(>|t|)
## x2.x1   0.5931     0.0250   23.73   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 923.6 on 495 degrees of freedom
## Multiple R-squared:  0.5321, Adjusted R-squared:  0.5312
## F-statistic:   563 on 1 and 495 DF,  p-value: < 2.2e-16
```

Beta Coefficients

Linear Modeling

full model

```
mod3.lm <- lm(water81 ~ income+water80+educat+retire+peop81+cpeop, data=concord)
summary(mod3.lm)
```

```
##
## Call:
## lm(formula = water81 ~ income + water80 + educat + retire + peop81 +
##     cpeop, data = concord)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4037.0  -447.6   -69.5   365.4  5038.0
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 242.22043  206.86382   1.171  0.24220
## income       20.96699    3.46372   6.053 2.83e-09 ***
## water80       0.49194    0.02635  18.671  < 2e-16 ***
## educat      -41.86552   13.22031  -3.167  0.00164 **
## retireyes   189.18433   95.02142   1.991  0.04704 *
## peop81      248.19702   28.72480   8.641  < 2e-16 ***
## cpeop        96.45360   80.51903   1.198  0.23154
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 849.3 on 489 degrees of freedom
## Multiple R-squared:  0.6773, Adjusted R-squared:  0.6734
## F-statistic: 171.1 on 6 and 489 DF,  p-value: < 2.2e-16
```

convert factor to metric. retired:“yes”=1,“no”=0

```
concord$retireDummy <- as.numeric(concord$retire)-1
```

Transfer all variables to scaled form

```
concordNew <- concord[ ,sapply(concord,is.numeric)]  # Remove non-numeric variables
concordScale <- as.data.frame(scale(concordNew))     # apply z-transformation with scale function
                                                     # dataframe concordScale holds the transformed val
mod4.lm <- lm(water81 ~ -1+income+water80+educat+retireDummy+peop81+cpeop, data=concordScale)
summary(mod4.lm)
```

```
##
## Call:
## lm(formula = water81 ~ -1 + income + water80 + educat + retireDummy +
##     peop81 + cpeop, data = concordScale)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7165 -0.3012 -0.0468  0.2459  3.3900
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## income       0.18423    0.03040   6.060 2.73e-09 ***
## water80      0.58386    0.03124  18.690  < 2e-16 ***
## educat      -0.08706    0.02746  -3.170  0.00162 **
## retireDummy  0.05808    0.02914   1.993  0.04682 *
## peop81       0.27676    0.03200   8.649  < 2e-16 ***
## cpeop        0.03146    0.02623   1.199  0.23106
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5709 on 490 degrees of freedom
## Multiple R-squared:  0.6773, Adjusted R-squared:  0.6734
## F-statistic: 171.4 on 6 and 490 DF,  p-value: < 2.2e-16
```

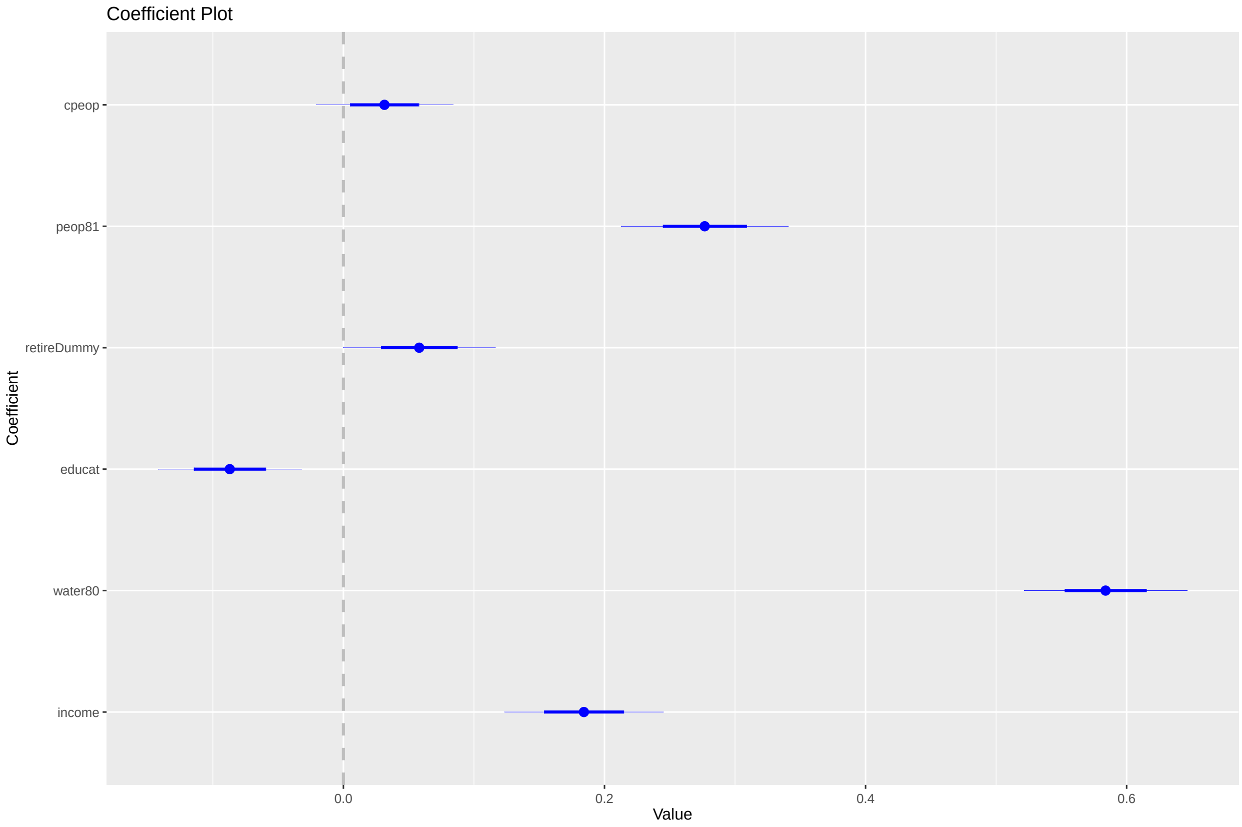
coefficent plot

coefficent plot useful for beta weights because the parameters are on the same scale

```
library (coefplot)
```

```
## Loading required package: ggplot2
```

```
coefplot(mod4.lm)
```



Partial F-test

H0: $\beta_{RETIRE} = \beta_{CPEOP} = 0$

```
mod5.lm <- lm(water81 ~ water80+income+educat+peop81, data=concord)
summary(mod5.lm)
```

```
##
## Call:
## lm(formula = water81 ~ water80 + income + educat + peop81, data = concord)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3956.4  -472.7   -65.3   365.7  4976.8
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 399.64803  188.85176   2.116 0.034830 *
## water80       0.48440    0.02613  18.538  < 2e-16 ***
## income       19.59823    3.35785   5.837 9.69e-09 ***
## educat      -43.98044   13.23258  -3.324 0.000955 ***
## peop81      240.50194   27.58814   8.718  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 852.3 on 491 degrees of freedom
## Multiple R-squared:  0.6738, Adjusted R-squared:  0.6711
## F-statistic: 253.5 on 4 and 491 DF,  p-value: < 2.2e-16
```

Compare both models

```
anova(mod5.lm,mod3.lm)
```

```
## Analysis of Variance Table
##
## Model 1: water81 ~ water80 + income + educat + peop81
## Model 2: water81 ~ income + water80 + educat + retire + peop81 + cpeop
##   Res.Df       RSS Df Sum of Sq     F  Pr(>F)
## 1    491 356658211
## 2    489 352761188  2   3897023 2.701 0.06814 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Nested F test Equal to t test

```
Anova(mod3.lm)
```

```
## Anova Table (Type II tests)
##
## Response: water81
##              Sum Sq  Df  F value    Pr(>F)
## income     26433751   1  36.6426 2.828e-09 ***
## water80   251481466   1 348.6053 < 2.2e-16 ***
## educat      7234388   1  10.0284  0.001638 **
## retire      2859560   1   3.9639  0.047041 *
## peop81     53858103   1  74.6585 < 2.2e-16 ***
## cpeop       1035170   1   1.4350  0.231537
## Residuals 352761188 489
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Stepwise Model

```
null.lm <- lm(water81~1, data=concord)
mod6.step <- step(null.lm,
                  scope= ~income+water80+educat+retire+peop81+cpeop,
                  direction="forward")
```

```
## Start:  AIC=7246.49
## water81 ~ 1
##
##           Df Sum of Sq       RSS    AIC
## + water80  1 639446987  453791723 6812.4
## + peop81   1 417946603  675292106 7009.5
## + income   1 190820566  902418143 7153.4
## + retire   1  81557382 1011681328 7210.0
## + cpeop    1   4777708 1088461002 7246.3
## <none>                 1093238710 7246.5
## + educat   1   1782539 1091456170 7247.7
##
## Step:  AIC=6812.38
## water81 ~ water80
##
##          Df Sum of Sq       RSS    AIC
## + peop81  1  70887612 382904111 6730.1
## + income  1  31578364 422213359 6778.6
## + cpeop   1   8853815 444937908 6804.6
## + retire  1   2971449 450820274 6811.1
## <none>                453791723 6812.4
## + educat  1   1332325 452459398 6812.9
##
## Step:  AIC=6730.13
## water81 ~ water80 + peop81
##
##          Df Sum of Sq       RSS    AIC
## + income  1  18221721 364682390 6707.9
## + cpeop   1   1989154 380914957 6729.5
## <none>                382904111 6730.1
## + educat  1   1501286 381402825 6730.2
## + retire  1    359352 382544758 6731.7
##
## Step:  AIC=6707.95
## water81 ~ water80 + peop81 + income
##
##          Df Sum of Sq       RSS    AIC
## + educat  1   8024179 356658211 6698.9
## + retire  1   3534967 361147423 6705.1
## <none>                364682390 6707.9
## + cpeop   1   1160227 363522163 6708.4
##
## Step:  AIC=6698.91
## water81 ~ water80 + peop81 + income + educat
##
##          Df Sum of Sq       RSS    AIC
## + retire  1   2861853 353796358 6696.9
## <none>                356658211 6698.9
## + cpeop   1   1037463 355620748 6699.5
##
## Step:  AIC=6696.92
## water81 ~ water80 + peop81 + income + educat + retire
##
##         Df Sum of Sq       RSS    AIC
## <none>               353796358 6696.9
## + cpeop  1   1035170 352761188 6697.5
```

```
summary(mod6.step)
```

```
##
## Call:
```

```
## lm(formula = water81 ~ water80 + peop81 + income + educat + retire,
##     data = concord)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4005.5  -462.9   -75.5   376.6  5035.7
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 229.8612   206.6980   1.112  0.26666
## water80       0.4875     0.0261  18.680  < 2e-16 ***
## peop81      253.9900    28.3273   8.966  < 2e-16 ***
## income       21.2943     3.4545   6.164 1.48e-09 ***
## educat      -42.1928    13.2233  -3.191  0.00151 **
## retireyes   189.2601    95.0636   1.991  0.04705 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 849.7 on 490 degrees of freedom
## Multiple R-squared:  0.6764, Adjusted R-squared:  0.6731
## F-statistic: 204.8 on 5 and 490 DF,  p-value: < 2.2e-16
```

Alternative stepwise specification

```
mod7.step <- step(mod3.lm, scope=list(lower=null.lm, upper=mod3.lm), direction="backward")
```

```
## Start:  AIC=6697.46
## water81 ~ income + water80 + educat + retire + peop81 + cpeop
##
##           Df Sum of Sq       RSS    AIC
## - cpeop    1   1035170 353796358 6696.9
## <none>                 352761188 6697.5
## - retire   1   2859560 355620748 6699.5
## - educat   1   7234388 359995575 6705.5
## - income   1  26433751 379194939 6731.3
## - peop81   1  53858103 406619291 6765.9
## - water80  1 251481466 604242653 6962.4
##
## Step:  AIC=6696.92
## water81 ~ income + water80 + educat + retire + peop81
##
##           Df Sum of Sq       RSS    AIC
## <none>                 353796358 6696.9
## - retire   1   2861853 356658211 6698.9
## - educat   1   7351065 361147423 6705.1
## - income   1  27436136 381232494 6732.0
## - peop81   1  58046844 411843202 6770.3
## - water80  1 251947001 605743359 6961.6
```

```
summary(mod7.step)
```

```
##
```

```
## Call:
## lm(formula = water81 ~ income + water80 + educat + retire + peop81,
##     data = concord)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4005.5  -462.9   -75.5   376.6  5035.7
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 229.8612   206.6980   1.112  0.26666
## income       21.2943     3.4545   6.164 1.48e-09 ***
## water80       0.4875     0.0261  18.680  < 2e-16 ***
## educat      -42.1928    13.2233  -3.191  0.00151 **
## retireyes   189.2601    95.0636   1.991  0.04705 *
## peop81      253.9900    28.3273   8.966  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 849.7 on 490 degrees of freedom
## Multiple R-squared:  0.6764, Adjusted R-squared:  0.6731
## F-statistic: 204.8 on 5 and 490 DF,  p-value: < 2.2e-16
```

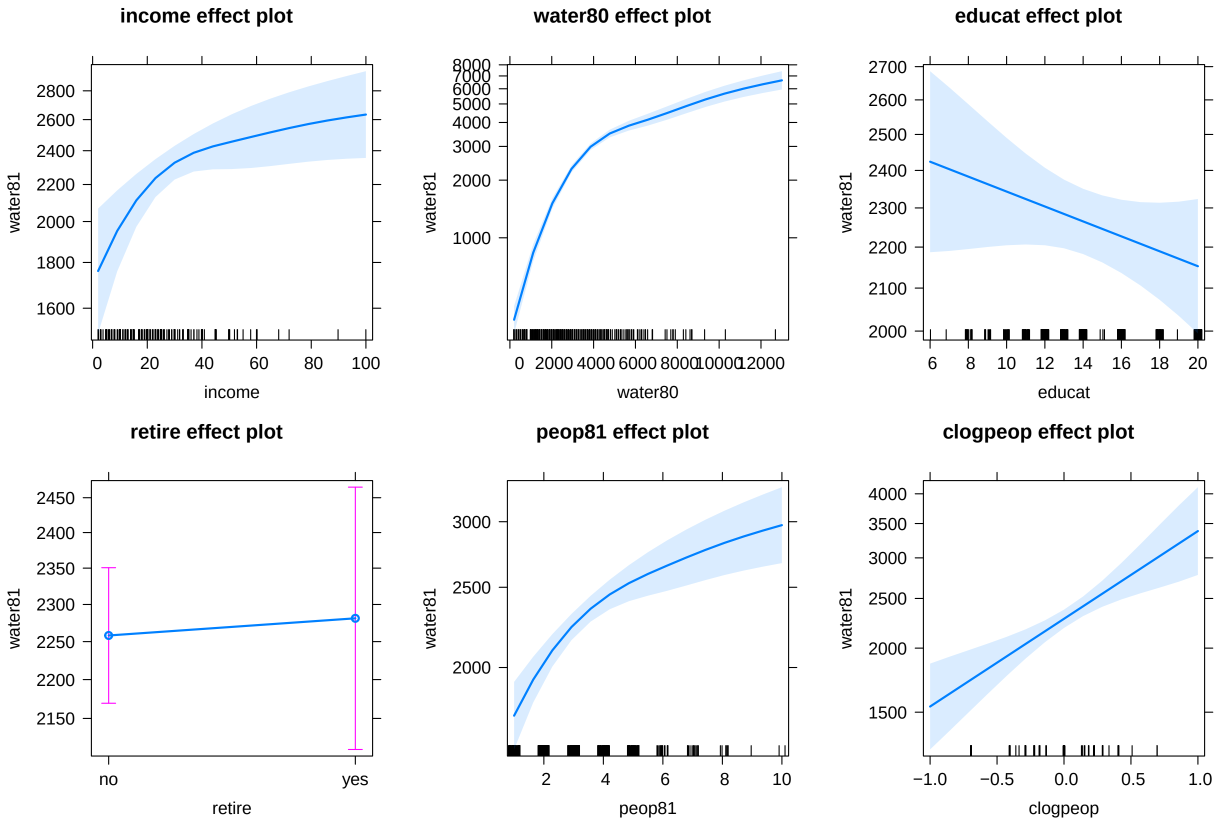
Conditional Effects

```
concord$clogpeop <- log(concord$peop81 / concord$peop80)
mod08.lm <- lm(log(water81)~log(income)+log(water80)+educat+retire+log(peop81)+clogpeop, data=concord)
summary(mod08.lm)
```

```
##
## Call:
## lm(formula = log(water81) ~ log(income) + log(water80) + educat +
##     retire + log(peop81) + clogpeop, data = concord)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.48013 -0.16567  0.02531  0.21069  1.78175
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.811259   0.240687   7.525 2.55e-13 ***
## log(income)   0.102958   0.034182   3.012  0.00273 **
## log(water80)  0.689469   0.032449  21.248  < 2e-16 ***
## educat       -0.008474   0.005939  -1.427  0.15425
## retireyes     0.010156   0.044540   0.228  0.81972
## log(peop81)   0.230115   0.041144   5.593 3.72e-08 ***
## clogpeop      0.394038   0.097559   4.039 6.23e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3788 on 489 degrees of freedom
## Multiple R-squared:  0.7102, Adjusted R-squared:  0.7066
## F-statistic: 199.7 on 6 and 489 DF,  p-value: < 2.2e-16
```

All effects at mean level of remaing variavles

```
plot(allEffects(mod08.lm, transformation=list(link=log, inverse=exp)), ylab="water81")
```



get value ranges

```
summary(concord)
```

```
##       case           water81          water80          water79
##  Min.   :  5.0   Min.   :  100   Min.   :  200   Min.   :  200
##  1st Qu.:133.8   1st Qu.: 1200   1st Qu.: 1500   1st Qu.: 1700
##  Median :259.5   Median : 2050   Median : 2300   Median : 2500
##  Mean   :260.4   Mean   : 2298   Mean   : 2732   Mean   : 2974
##  3rd Qu.:386.2   3rd Qu.: 2900   3rd Qu.: 3700   3rd Qu.: 3800
##  Max.   :516.0   Max.   :10100   Max.   :12700   Max.   :14500
##                                                  NA's   :47
##      income           educat     retire        peop81           cpeop
##  Min.   :  2.00   Min.   : 6   no :350   Min.   : 1.000   Min.   :-3.00000
##  1st Qu.: 15.00   1st Qu.:12   yes:146   1st Qu.: 2.000   1st Qu.: 0.00000
##  Median : 21.00   Median :13             Median : 3.000   Median : 0.00000
##  Mean   : 23.08   Mean   :14             Mean   : 3.073   Mean   :-0.03831
##  3rd Qu.: 30.00   3rd Qu.:16             3rd Qu.: 4.000   3rd Qu.: 0.00000
##  Max.   :100.00   Max.   :20             Max.   :10.000   Max.   : 3.00000
##
##      peop80        retireDummy        clogpeop
##  Min.   : 1.000   Min.   :0.0000   Min.   :-1.09861
##  1st Qu.: 2.000   1st Qu.:0.0000   1st Qu.: 0.00000
##  Median : 3.000   Median :0.0000   Median : 0.00000
##  Mean   : 3.111   Mean   :0.2944   Mean   :-0.01874
##  3rd Qu.: 4.000   3rd Qu.:1.0000   3rd Qu.: 0.00000
##  Max.   :10.000   Max.   :1.0000   Max.   : 1.38629
##
```

```
summary(log(concord$water80))
```

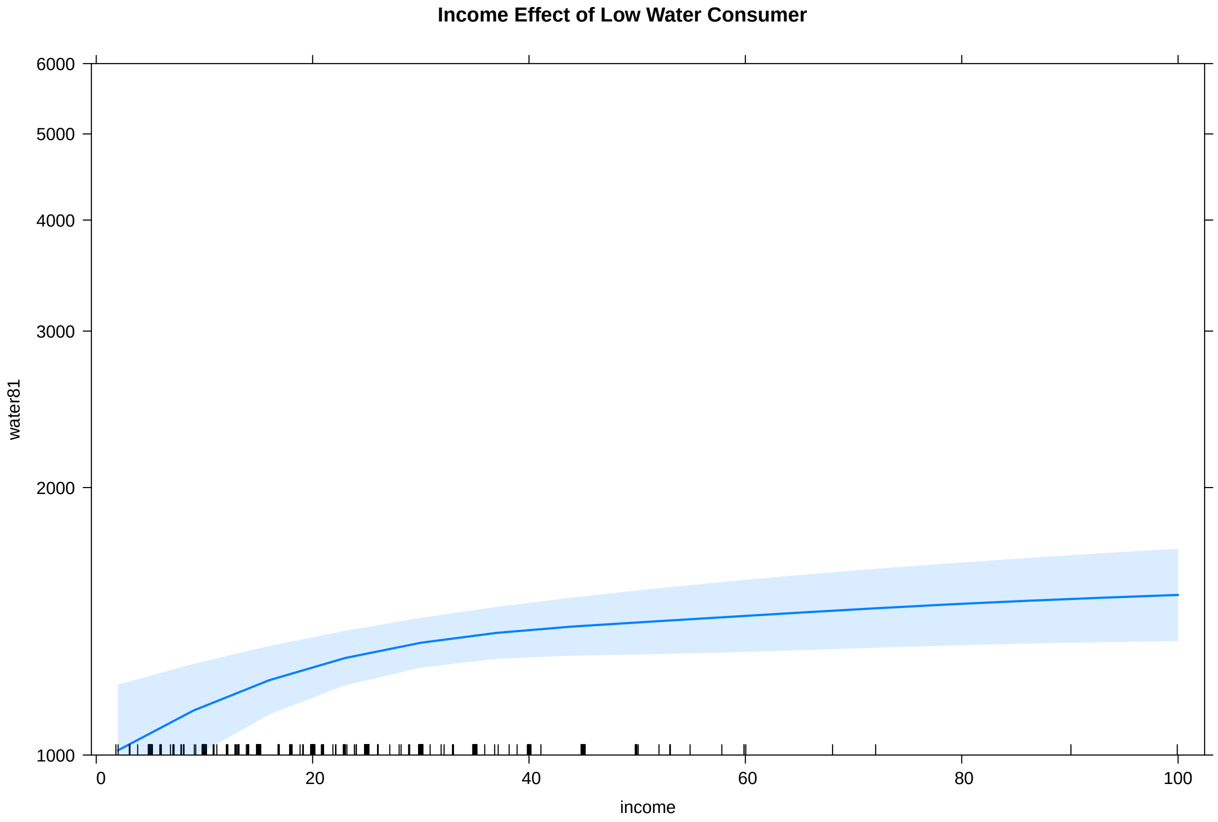
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   5.298   7.313   7.741   7.707   8.216   9.449
```

```
summary(log(concord$peop81))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.6931  1.0986  0.9751  1.3863  2.3026
```

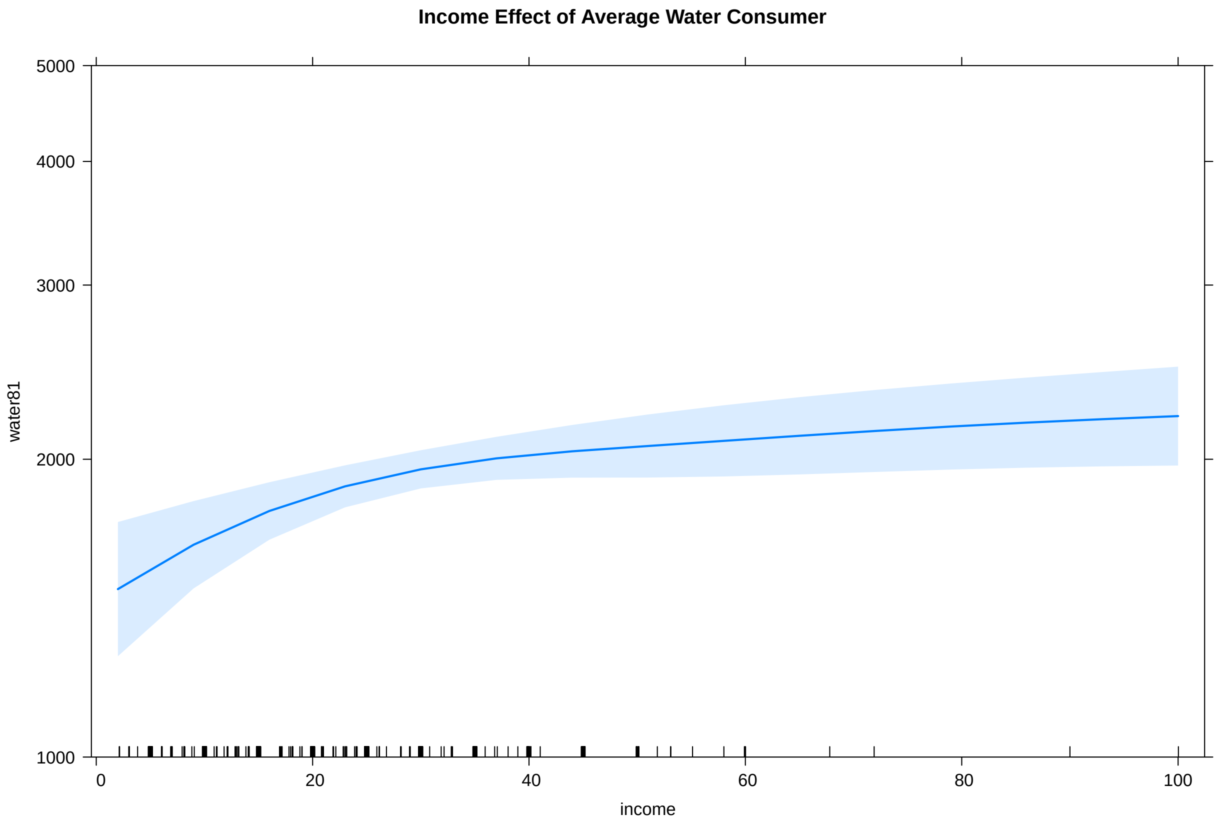
Income effect for a low consumer profile

```
plot(effect("log(income)", mod08.lm, given.values=c("log(water80)"=7.3, "educat"=18, "log(peop81)"=0.7)
            transformation=list(link=log, inverse=exp) ),
     ylim=c(log(1000),log(6000)), ylab="water81",
     main="Income Effect of Low Water Consumer")
```



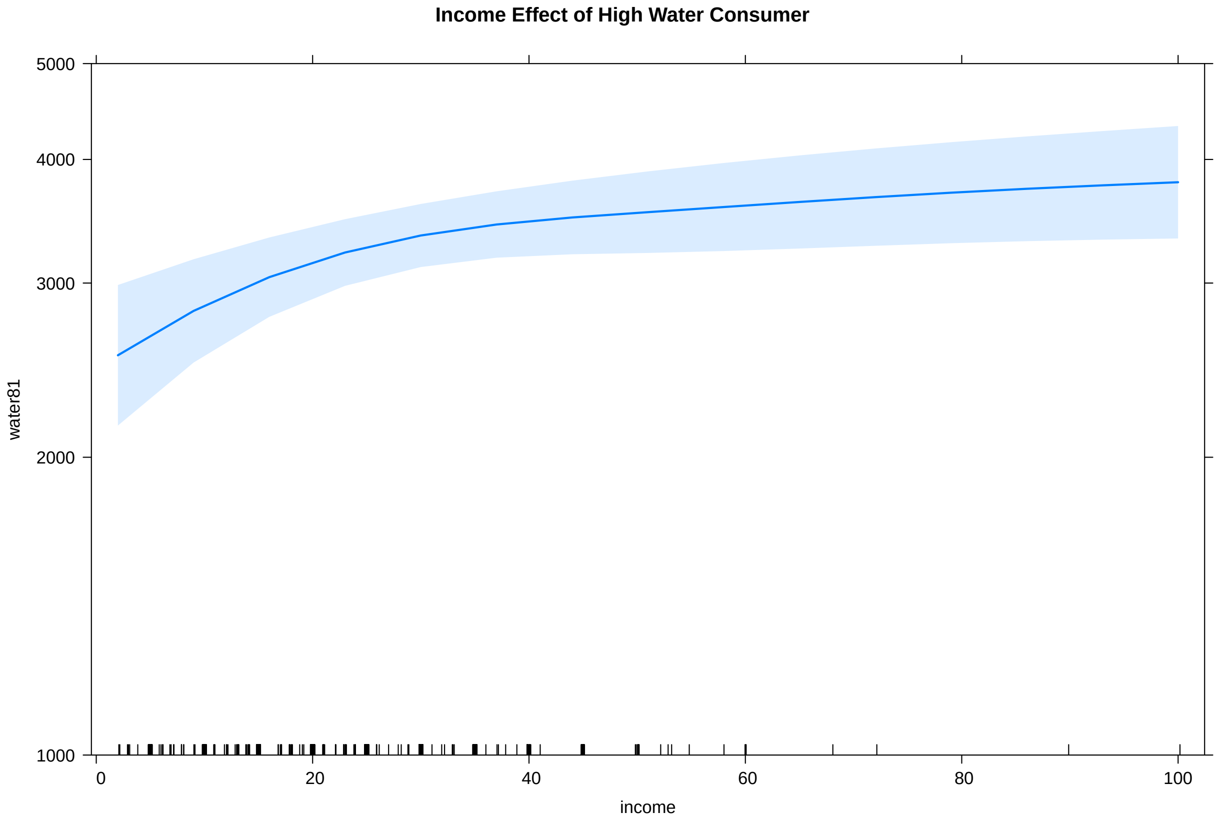
Income effect for an average consumer profile

```
plot(effect("log(income)", mod08.lm, given.values=c("log(water80)"=7.7, "educat"=14, "log(peop81)"=1.0)
            transformation=list(link=log, inverse=exp) ),
     ylim=c(log(1000),log(5000)), ylab="water81",
     main="Income Effect of Average Water Consumer")
```



Income effect for a high consumer profile

```
plot(effect("log(income)", mod08.lm, given.values=c("log(water80)"=8.3, "educat"=10, "log(peop81)"=1.4)
            transformation=list(link=log, inverse=exp) ),
     ylim=c(log(1000),log(5000)), ylab="water81", main="Income Effect of High Water Consumer")
```



Factor Variable Analysis

Initialize Data

```
wells <- foreign::read.spss("wells.sav",to.data.frame=TRUE)
summary(wells)
```

```
##        deep        droad             chlor
##  shallow :10   Min.   :  20.0   Min.   :  3.0
##  deep wel:43   1st Qu.:  60.0   1st Qu.: 10.0
##                Median : 100.0   Median : 10.5
##                Mean   : 251.2   Mean   : 81.5
##                3rd Qu.: 300.0   3rd Qu.: 43.5
##                Max.   :2640.0   Max.   :760.0
##                                 NA's   :1
```

Explore Coding Scheme of Factors

See coding of factor

```
class(wells$deep)
```

```
## [1] "factor"
```

```
contrasts(wells$deep)
```

```
##          deep wel
## shallow         0
## deep wel        1
```

Change to 1,0,-1 coding

```
contrasts(wells$deep) <- "contr.sum"
contrasts(wells$deep)
```

```
##          [,1]
## shallow     1
## deep wel   -1
```

Change back to 0,1 coding

```
contrasts(wells$deep) <- "contr.treatment"
contrasts(wells$deep)
```

```
##          deep wel
## shallow         0
## deep wel        1
```

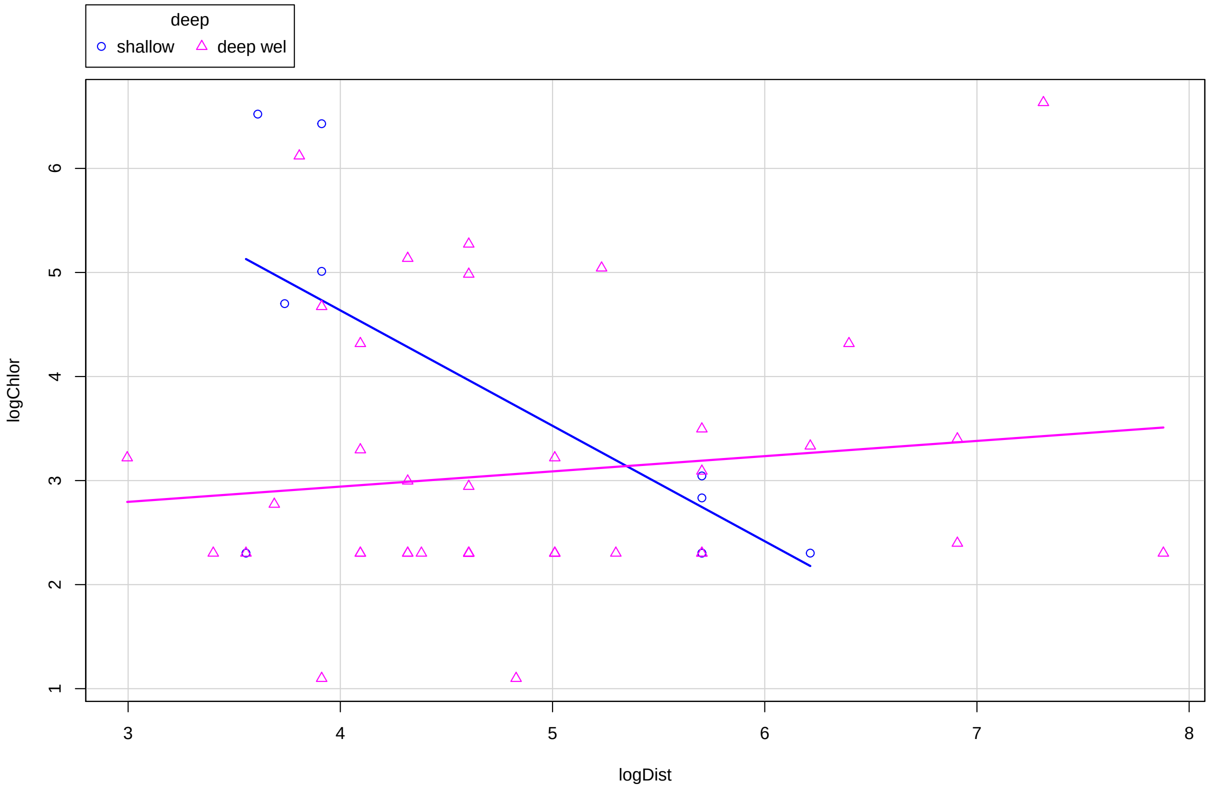
Prepare data for analysis

```
wells1 <- na.omit(wells)                               # drop observation 18 with missing values for Ch
wells1$logChlor <- log(wells1$chlor)                   # Transform to natural logarithm
wells1$logDist <- log(wells1$droad)                    # Transform to natural logarithm
(wells1$deepdum <- as.numeric(unclass(wells1$deep))-1)   # Convert factor to numeric 0/1 dummy
```

```
##  [1] 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
## [39] 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

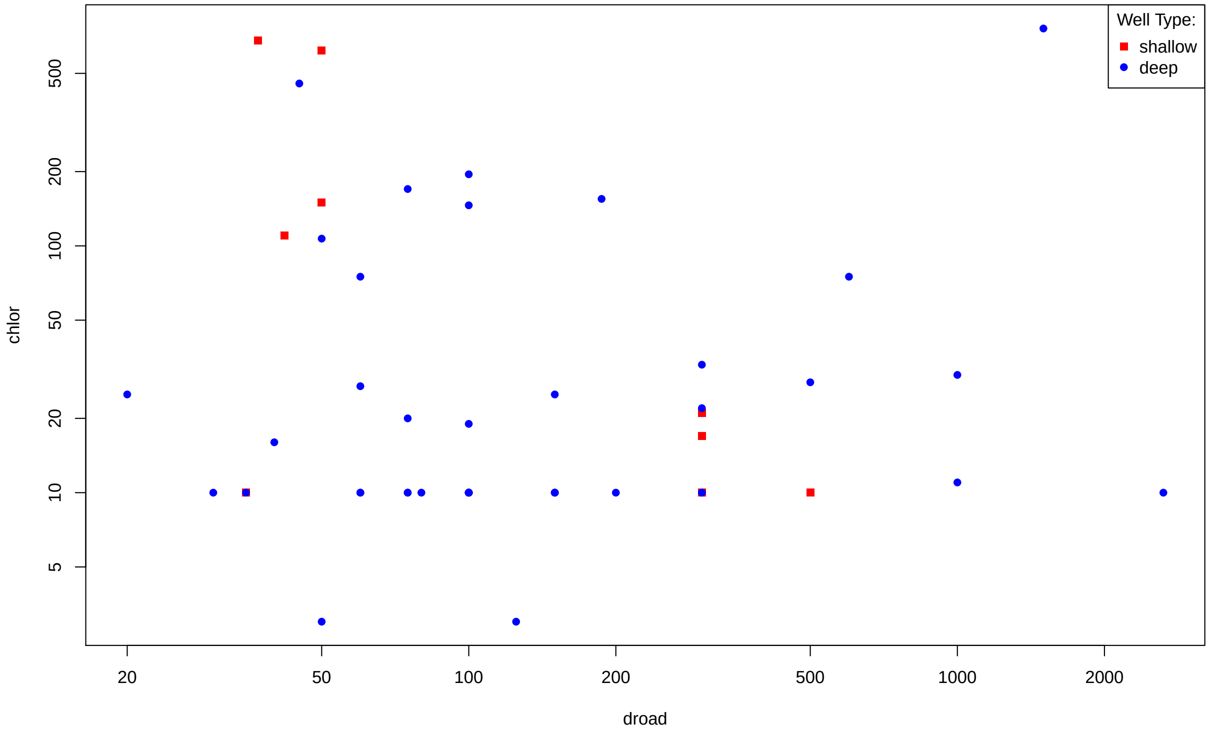
```
attach(wells1)
```

```
scatterplot(logChlor~logDist|deep, smooth=F, data=wells1)
```



Symbols & colors for well type

```
wellSymbol <- ifelse(deep==levels(deep)[1],15,16)
wellCol <- ifelse(deep==levels(deep)[1],"red","blue")
plot(droad,chlor, log="xy", pch=wellSymbol, col=wellCol)
legend("topright",legend=c("shallow","deep"),
       title="Well Type:", col=c("red","blue"),pch=c(15,16))
```



Linear Model for factor variables

No intercept Model

```
mod0 <- lm(logChlor ~ deep -1) # One way analysis of variance
summary(mod0)                 # Suppressing intercept gives mean levels
```

```
##
## Call:
## lm(formula = logChlor ~ deep - 1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9707 -0.7667 -0.7667  0.5517  3.5640
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## deepshallow    3.7751     0.4289   8.801 9.85e-12 ***
## deepdeep wel   3.0693     0.2093  14.664  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.356 on 50 degrees of freedom
## Multiple R-squared:  0.854,  Adjusted R-squared:  0.8482
## F-statistic: 146.2 on 2 and 50 DF,  p-value: < 2.2e-16
```

intercept model

```
mod1 <- lm(logChlor ~ deep)   # One-way analysis of variance
summary(mod1)
```

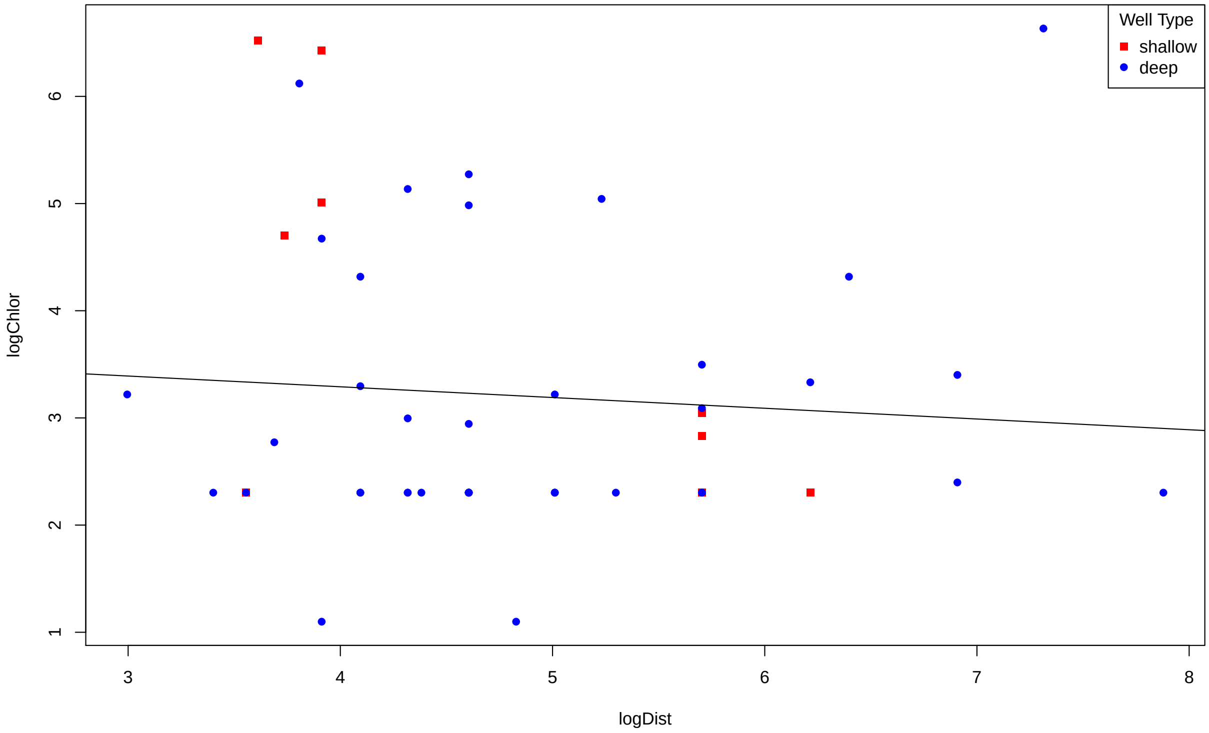
```
##
## Call:
## lm(formula = logChlor ~ deep)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9707 -0.7667 -0.7667  0.5517  3.5640
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.7751     0.4289   8.801 9.85e-12 ***
## deepdeep wel  -0.7058     0.4773  -1.479    0.145
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.356 on 50 degrees of freedom
## Multiple R-squared:  0.0419, Adjusted R-squared:  0.02274
## F-statistic: 2.187 on 1 and 50 DF,  p-value: 0.1455
```

Standard Regression

```
mod2 <- lm(logChlor ~ logDist)
summary(mod2)
```

```
##
## Call:
## lm(formula = logChlor ~ logDist)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2007 -0.9273 -0.6002  0.5607  3.6749
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.6914     0.9017   4.094 0.000155 ***
## logDist      -0.1002     0.1816  -0.552 0.583397
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.382 on 50 degrees of freedom
## Multiple R-squared:  0.006057,   Adjusted R-squared:  -0.01382
## F-statistic: 0.3047 on 1 and 50 DF,  p-value: 0.5834
```

```
plot(logDist,logChlor,pch=wellSymbol,col=wellCol)      # just distance not interaction
abline(mod2)
legend("topright",legend=c("shallow","deep"), title="Well Type", col=c("red","blue"),pch=c(15,16))
```

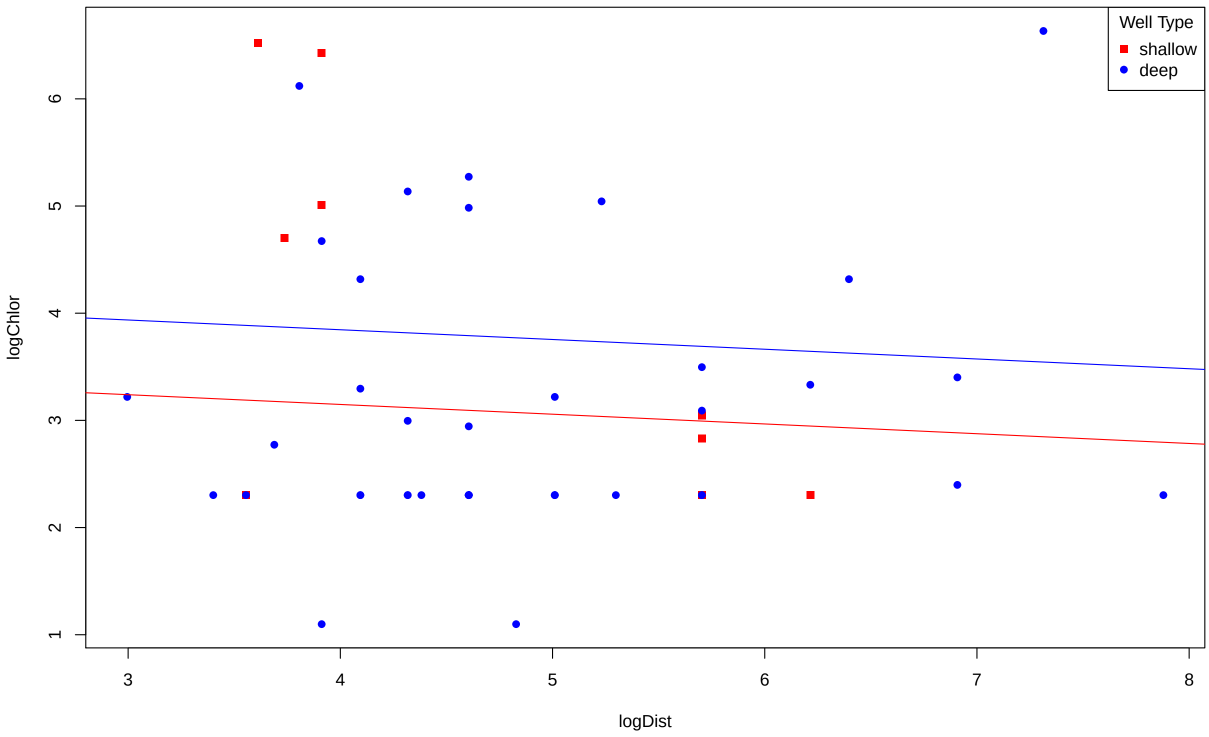


Regression with intercept dummy

```
mod3 <- lm(logChlor ~ deep + logDist)
summary(mod3)
```

```
##
## Call:
## lm(formula = logChlor ~ deep + logDist)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0579 -0.8127 -0.6686  0.5956  3.7862
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.20954    0.96096   4.381 6.22e-05 ***
## deepdeep wel -0.69712    0.48119  -1.449    0.154
## logDist      -0.09097    0.17972  -0.506    0.615
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.367 on 49 degrees of freedom
## Multiple R-squared:  0.04688,    Adjusted R-squared:  0.007981
## F-statistic: 1.205 on 2 and 49 DF,  p-value: 0.3084
```

```
plot(logDist,logChlor,pch=wellSymbol,col=wellCol)      # intercept dummy
abline(mod3$coef[1],mod3$coef[3],col="blue")
abline(mod3$coef[1]+mod3$coef[2],mod3$coef[3],col="red")
legend("topright",legend=c("shallow","deep"), title="Well Type", col=c("red","blue"),pch=c(15,16))
```

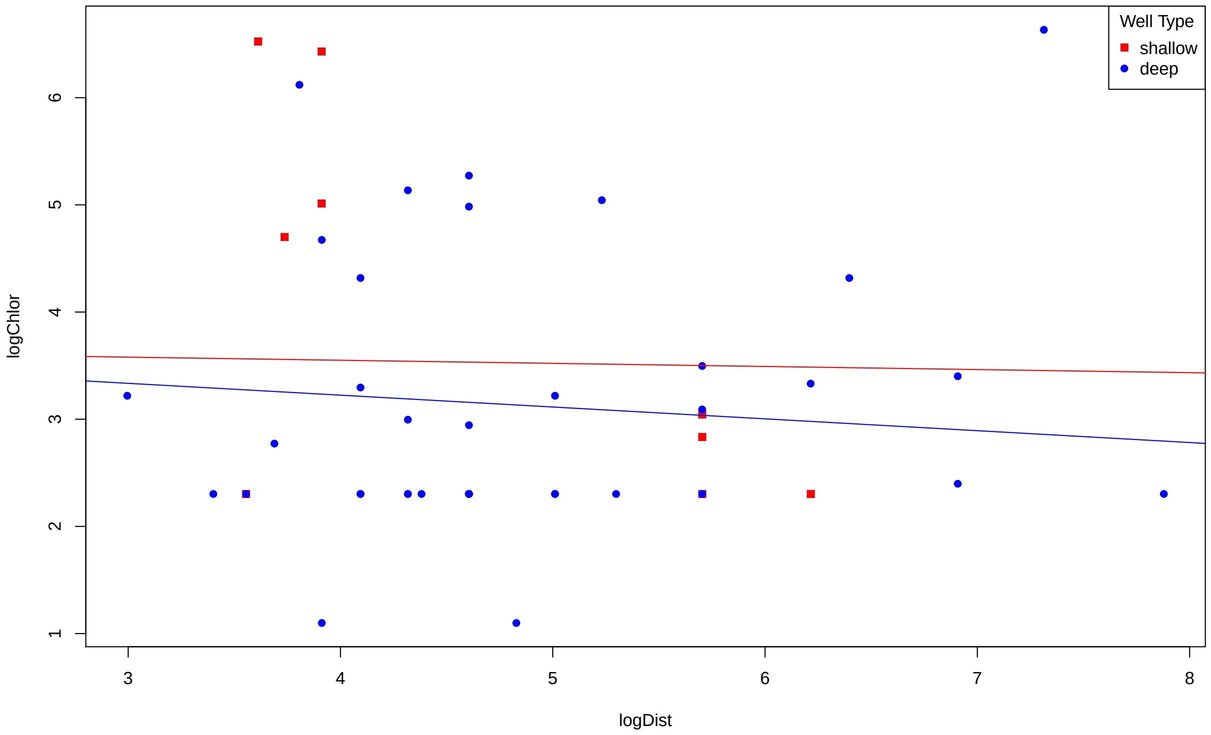


Regression with slope dummy

```
mod4 <- lm(logChlor ~ logDist + logDist:deep)
summary(mod4)
```

```
##
## Call:
## lm(formula = logChlor ~ logDist + logDist:deep)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.1355 -0.8611 -0.5865  0.6493  3.7748
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           3.66615    0.90518   4.050 0.000182 ***
## logDist              -0.02897    0.20187  -0.144 0.886478
## logDist:deepdeep wel -0.08147    0.09946  -0.819 0.416682
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.386 on 49 degrees of freedom
## Multiple R-squared:  0.01948,    Adjusted R-squared:  -0.02054
## F-statistic: 0.4868 on 2 and 49 DF,  p-value: 0.6175
```

```
plot(logDist,logChlor,pch=wellSymbol,col=wellCol)      # slope dummy
abline(mod4$coef[1],mod4$coef[2],col="red")
abline(mod4$coef[1],mod4$coef[2]+mod4$coef[3],col="blue")
legend("topright",legend=c("shallow","deep"), title="Well Type", col=c("red","blue"),pch=c(15,16))
```



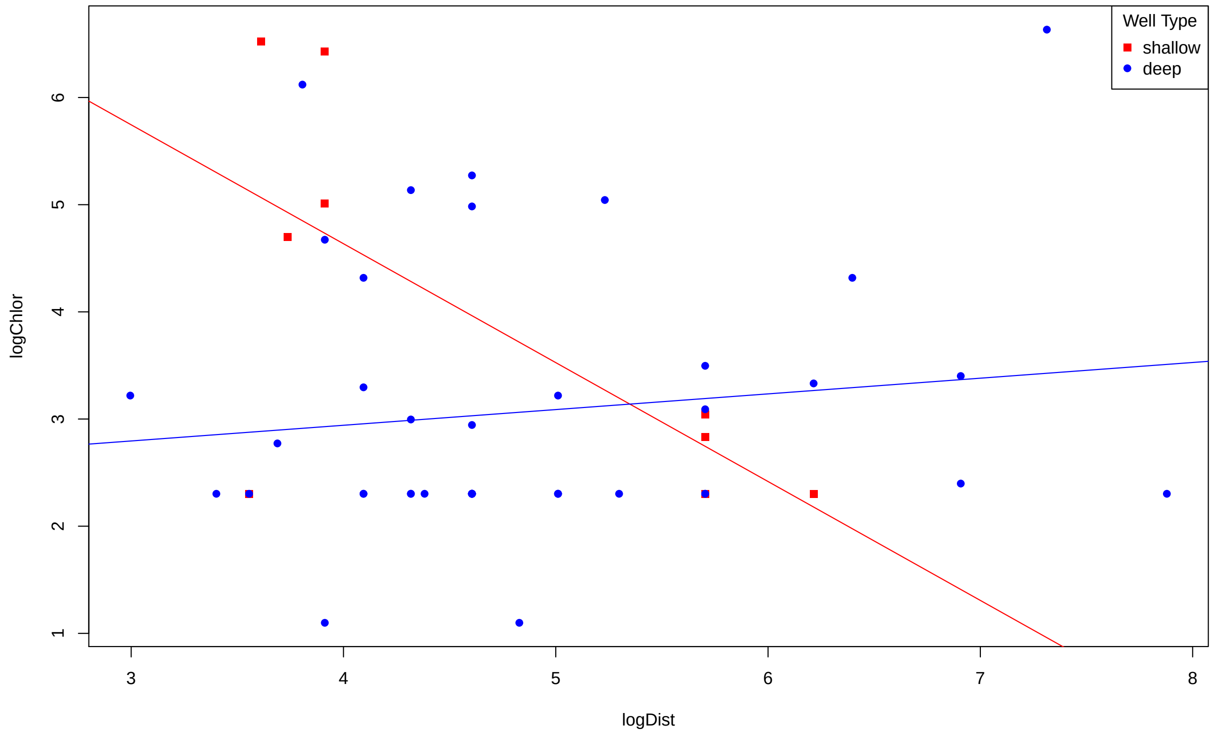
Regression with intercept and slope dummy

```
mod5 <- lm(logChlor ~ deep*logDist)
summary(mod5)
```

```
##
## Call:
## lm(formula = logChlor ~ deep * logDist)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8265 -0.7278 -0.3346  0.3140  3.2068
##
```

```
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)              9.0735     1.8794   4.828 1.45e-05 ***
## deepdeep wel            -6.7174     2.0947  -3.207  0.00239 **
## logDist                 -1.1094     0.3844  -2.886  0.00583 **
## deepdeep wel:logDist     1.2558     0.4269   2.942  0.00501 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.271 on 48 degrees of freedom
## Multiple R-squared:  0.1925, Adjusted R-squared:  0.142
## F-statistic: 3.814 on 3 and 48 DF,  p-value: 0.0157
```

```
mod5 <- lm(logChlor ~ deep + logDist + logDist:deep)
plot(logDist,logChlor,pch=wellSymbol,col=wellCol)      # intercept and slope dummy
abline(mod5$coef[1],mod5$coef[3],col="red")
abline(mod5$coef[1]+mod5$coef[2],mod5$coef[3]+mod5$coef[4],col="blue")
legend("topright",legend=c("shallow","deep"), title="Well Type", col=c("red","blue"),pch=c(15,16))
```



```
anova(mod2,mod5)
```

```
## Analysis of Variance Table
##
## Model 1: logChlor ~ logDist
## Model 2: logChlor ~ deep + logDist + logDist:deep
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1     50 95.441
## 2     48 77.539  2    17.901 5.5409 0.006838 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```