

Lab 6: Interactions

Hao Wang

March 09, 2017

1. Load essential packages

```
library(VGAM)
library(MASS)
library(effects)
library(ggplot2)
library(Zelig)
library(ZeligChoice)
library(car)
library(interflex)
```

More to read:

1. Hainmueller et al. 2017: How Much Should We Trust Estimates from Multiplicative Interaction Models [click here](#)
2. Matt Golder's Webpage [click here](#)

2. Interaction in a glance

Linear interaction term has the the following general function:

$$Y = f(X, D, Z) = \mu + \eta X + \alpha D + \beta(D * X) + \gamma Z + \epsilon$$

In this model Y is the dependent variable, D is the main independent variable (or the 'treatment' in the experimental settings). X is the moderator, and the D*X is the interaction term between D and X. Z is a list of other control variables, μ is the intercept and ϵ is the error term.

In this model the interaction term implies a linear interaction effect (LIE), which means the *marginal effect* of treatment D on the dependent variable Y is

$$MAR_D = \frac{\partial Y}{\partial D} = \alpha + \beta X$$

MAR_D stands for the marginal effect of D. This implies that the effect of D on Y has a linear relation with the moderator X. Because it is linear, the relative treatment effect of $D = d_1$ and $D = d_2$ can be expressed as

$$\begin{aligned} Eff(d_1, d_2) &= Y(D = d_1 | X, Z) - Y(D = d_2 | X, Z) \\ &= (\mu + \alpha d_1 + \eta X + \beta d_1 X) - (\mu + \alpha d_2 + \eta X + \beta d_2 X) \\ &= \alpha(d_1 - d_2) + \beta(d_1 - d_2)X \end{aligned}$$

We can see the effects of D on Y is a linear function on X. This is a strong assumption and will often be violated.

3. Interpretation and Plot Marginal Effects

Following Brambor, Clark and Golder (2006), we should always include the treatment and moderator terms when we have an interaction. To interpret the effects with interactions, it is better to calculate the marginal probabilities.

In the following example we play with simulated data.

```
set.seed(61) #Barcelona beat Paris SG!
#set number of obs
n <- 1000
d <- rnorm(n,0,1) # continuous treatment
x <- rnorm(n,3,1) # moderator
z <- rnorm(n,3,1) # covariate
e <- rnorm(n,0,3) # error term, normal distribution, mean 0, sd 5
#Let's build our Y
Y <- 10 - 2 * x - 3 * d + 2 * x * d + 1 * z + e
df <- cbind.data.frame(Y=Y, D=d, X=x, Z=z)
```

Now we can estimate our model with interaction

```
int <- lm(Y ~ D*X + Z, data=df)
summary(int)
```

```
##
## Call:
## lm(formula = Y ~ D * X + Z, data = df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -10.8061  -2.0191   0.0009   2.0617   9.2531
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  10.27310    0.40528  25.348  <2e-16 ***
## D           -2.86941    0.30266  -9.481  <2e-16 ***
## X           -2.07239    0.09548 -21.705  <2e-16 ***
## Z             0.97348    0.09251  10.523  <2e-16 ***
## D:X           1.97523    0.09399  21.016  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.007 on 995 degrees of freedom
## Multiple R-squared:  0.6854, Adjusted R-squared:  0.6841
## F-statistic: 541.9 on 4 and 995 DF,  p-value: < 2.2e-16
```

The marginal effect of treatment D is $-2.86 + 1.97X$. To get the standard error, we need to take a square root of $Var(\beta_D + \beta_{DX}X)$

expand this function

$$\begin{aligned} & Var(\beta_D + \beta_{DX}X) \\ &= Var(\beta_D) + 2Cov(\beta_D, \beta_{DX}) * X + Var(\beta_{DX}) * Var(X) \end{aligned}$$

And X can be regarded as a given value (as we're calculating the marginal effect of D with the value of X given), thus

$$= Var(\beta_D) + 2Cov(\beta_D, \beta_{DX}) * X + Var(\beta_{DX}) * X^2$$

We can get the variance covariance matrix by `vcov()`

```
int <- lm(Y ~ D*X + Z, data=df)
attach(df)
vcov(int)
```

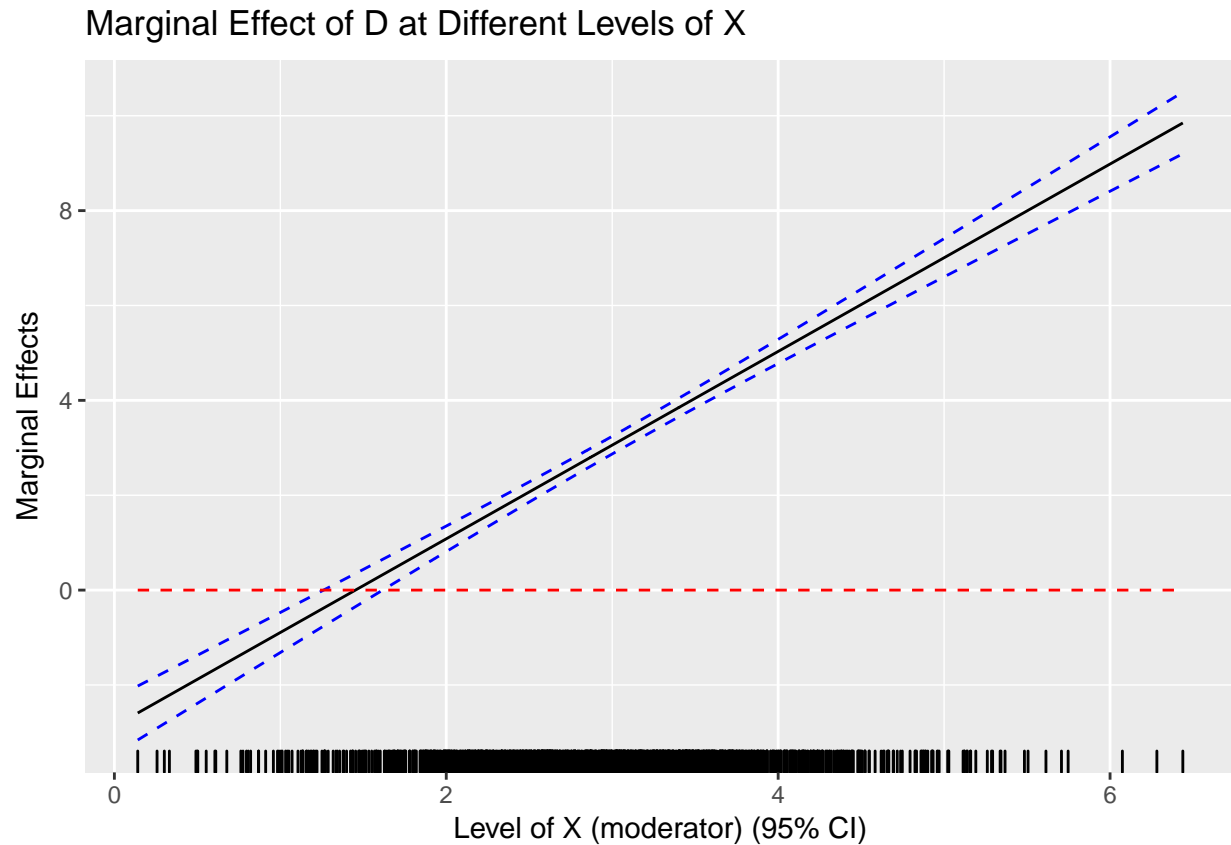
```
##              (Intercept)              D              X              Z
## (Intercept)  0.1642540159 -2.314035e-03 -0.0267344378 -2.506400e-02
## D            -0.0023140345  9.160226e-02  0.0007432555 -6.821738e-05
## X            -0.0267344378  7.432555e-04  0.0091166893 -1.928436e-04
## Z            -0.0250639955 -6.821738e-05 -0.0001928436  8.557326e-03
## D:X          0.0008126947 -2.701098e-02 -0.0001643986 -3.696516e-05
##              D:X
## (Intercept)  8.126947e-04
## D            -2.701098e-02
## X            -1.643986e-04
## Z            -3.696516e-05
## D:X          8.833932e-03
```

And we can now calculate the standard error of marginal effects given by the previous equation.

```
var <- vcov(int)[2,2] + 2*vcov(int)[2,5]*X + vcov(int)[5,5]*X^2
se <- sqrt(var)
df$mar <- int$coefficients[2] + int$coefficients[5]*X
#calculate the 95% CI of the marginal effect
df$up <- df$mar + 1.96*se
df$low <- df$mar - 1.96*se
```

Don't forget to plot it in ggplot2!

```
#graphic presentation
ggplot(data=df, aes(x = X))+
  geom_line(aes(y =mar)) +
  geom_line(aes(y =up), linetype="dashed", color="blue") +
  geom_line(aes(y =low), linetype="dashed", color="blue") +
  geom_line(aes(y =0), color = "red", linetype="dashed") +
  xlab("Level of X (moderator) (95% CI)") +
  ylab("Marginal Effects") +
  ggtitle("Marginal Effect of D at Different Levels of X")+
  geom_rug(sides="b")
```



4. Diagnostics

Hainmueller et al. (2017) suggest several ways to check the interaction models.

1. The first is to check if there are enough points at the different levels of moderators (avoid extrapolation)
2. The second is to check if the linear interactive relation holds

When the treatment is binary (0 and 1, receive or not), we can simply plot the relation between Y and X at $D = 0$ and $D = 1$.

4.1 Raw Plots

Let's first create a binary treatment model.

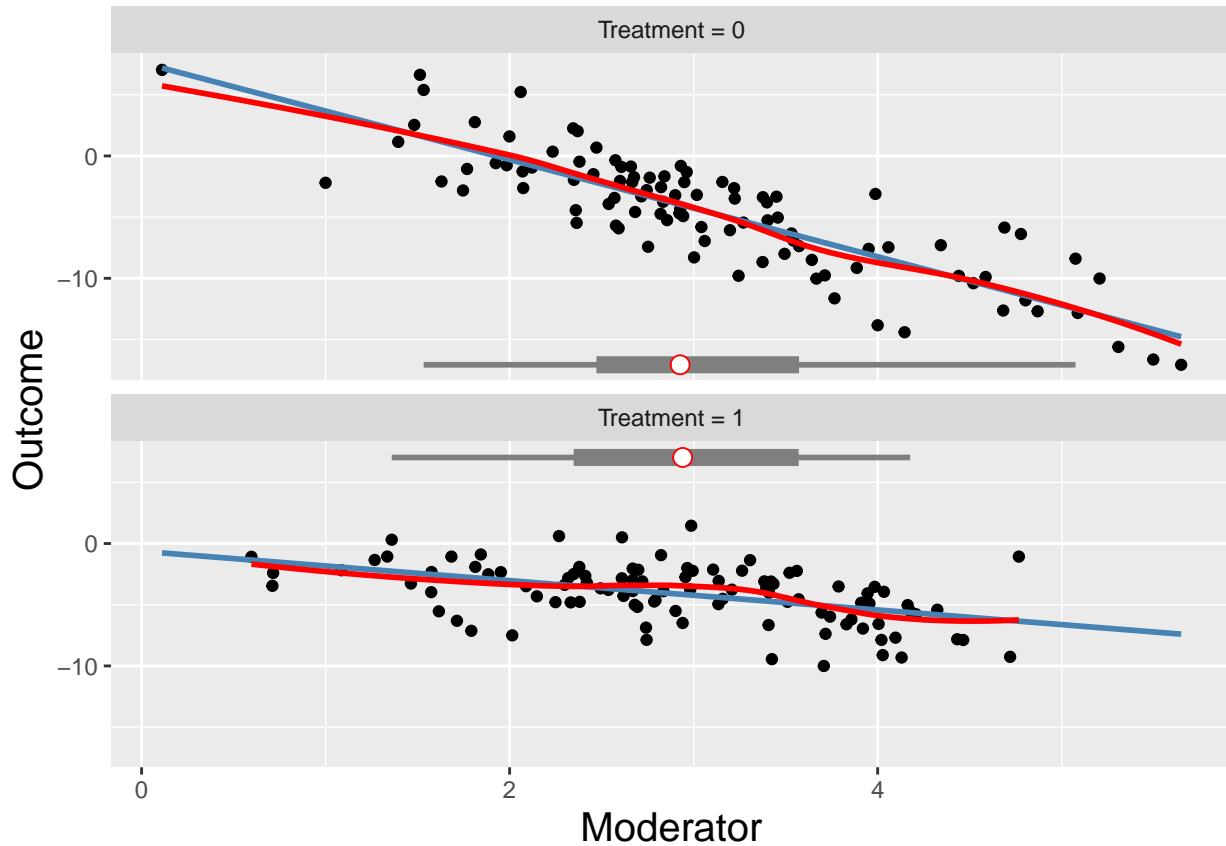
```
set.seed(1)
n<-200
d1<-sample(c(0,1),n,replace=TRUE) # dichotomous treatment
x<-rnorm(n,3,1) # moderator
z<-rnorm(n,3,1) # covariate
e<-rnorm(n,0,1) # error term
y1<-5 - 4 * x - 9 * d1 + 3 * x * d1 + 1 * z + 2 * e
s1<-cbind.data.frame(Y = y1, D = d1, X = x, Z1 = z)
```

We can use the written function in the *interflex* package to check the binary treatment case. In the graph, the red line is the LOESS fit and the blue line is the linear fit. To check the linear effect assumption, we

need to check if the LOESS fit overlaps with the linear fit.

```
#You can specify your moderator at X option and your treatment at the D option.
inter.raw(Y = "Y", D = "D", X = "X", data = s1,
          weights = NULL, Ylabel = "Outcome", Dlabel = "Treatment", Xlabel="Moderator")
```

```
## $graph
```



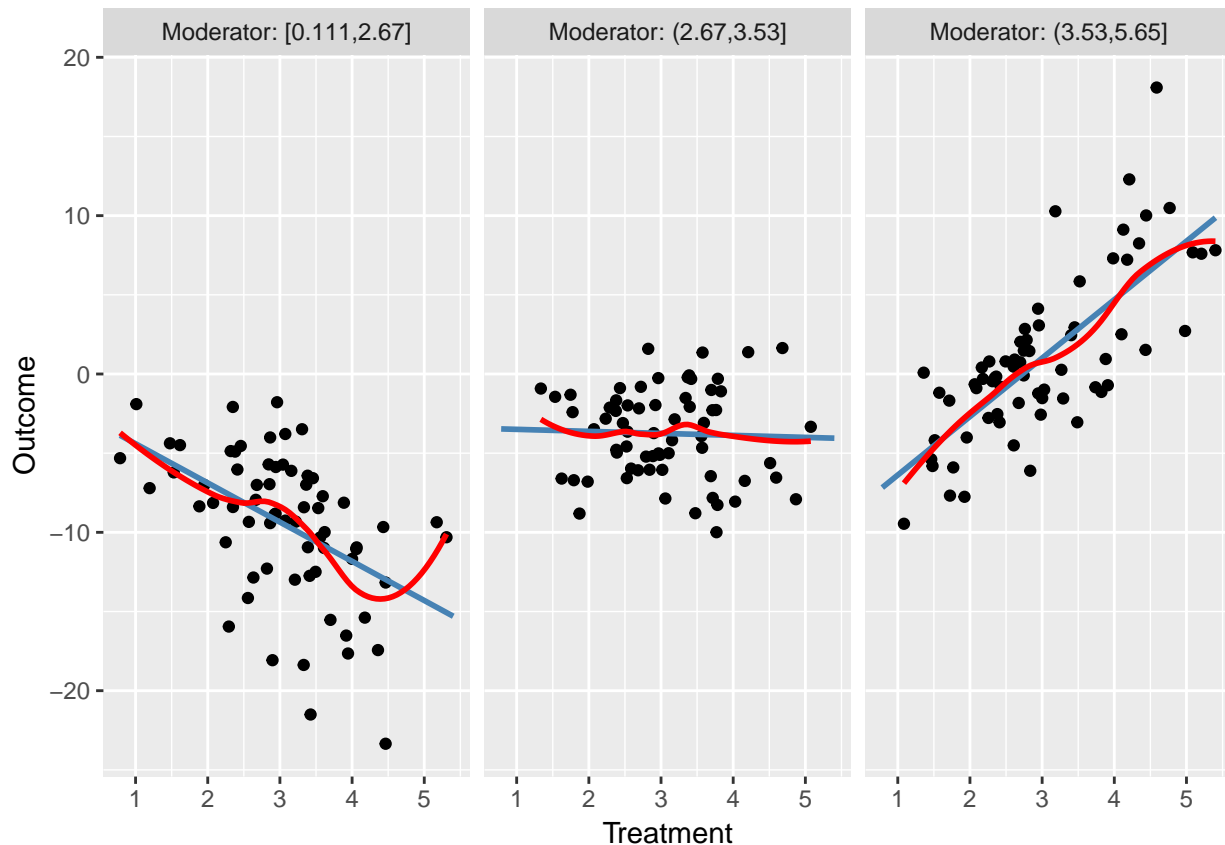
When the treatment is continuous, we can apply the similar idea: plot the data at different levels of moderator X. Package *interflex* has a built-in function for continuous treatment, in which it plots data at first tercile (low), second tercile (medium) and third tercile (high).

Let's create a continuous treatment model

```
set.seed(1)
n<-200
d2<-rnorm(n,3,1) # continuous treatment
x<-rnorm(n,3,1) # moderator
z<-rnorm(n,3,1) # covariate
e<-rnorm(n,0,1) # error term
y2<-5 - 4 * x - 9 * d2 + 3 * x * d2 + 1 * z + 2 * e
s2<-cbind.data.frame(Y = y2, D = d2, X = x, Z1 = z)
```

```
inter.raw(Y = "Y", D = "D", X = "X", data = s2, Ylabel = "Outcome", Dlabel = "Treatment", Xlabel="Moderator")
```

```
## $graph
```

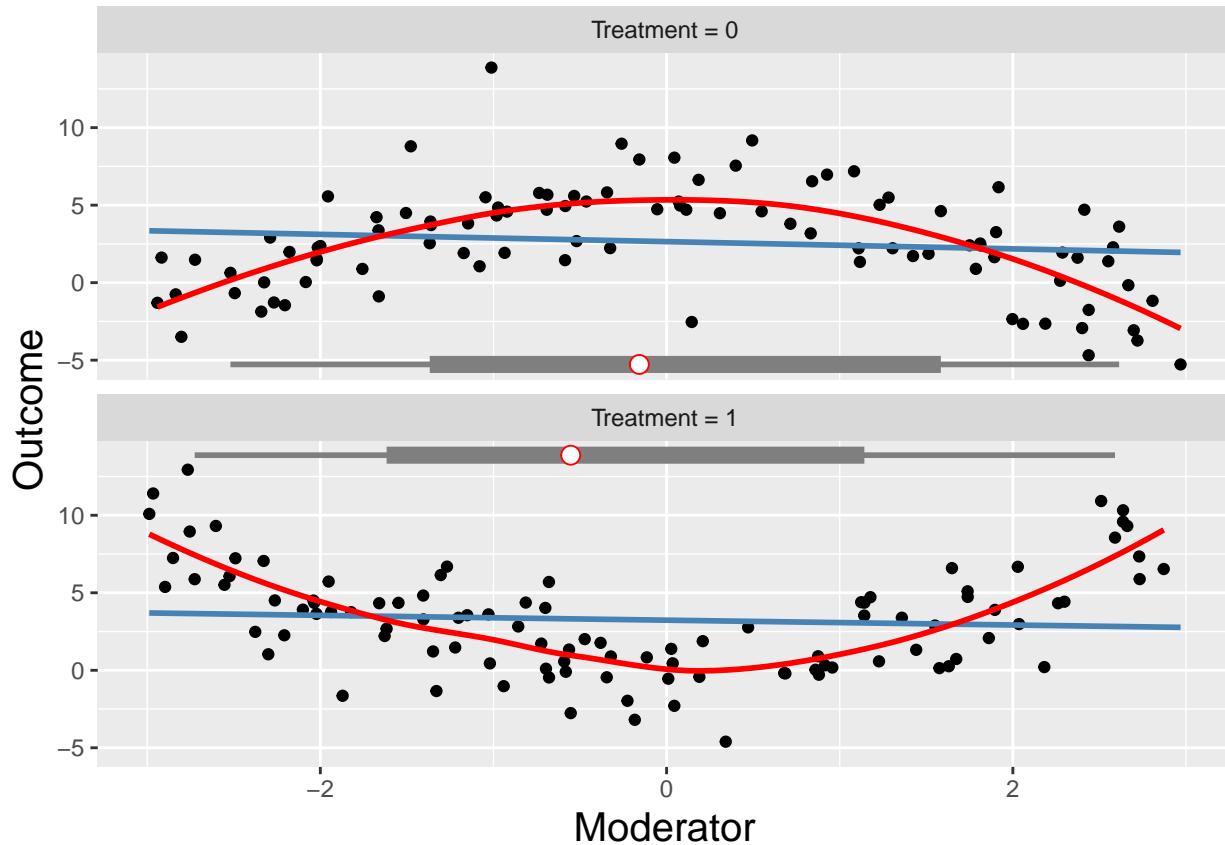


Sometimes our interaction term can be nonlinear. The following example is a quadratic interaction with binary treatment

```
set.seed(1)
n<-200
d1<-sample(c(0,1),n,replace=TRUE) # dichotomous treatment
z<-rnorm(n,3,1) # covariate
e<-rnorm(n,0,1) # error term
x3 <- runif(n, -3,3) # uniformly distributed moderator
#quadratic interaction
y3 <- d1*(x3^2-2.5) + (1-d1)*(-1*x3^2+2.5) + 1 * z + 2 * e
s3 <- cbind.data.frame(D=d1, X=x3, Y=y3, Z1 = z)
```

The diagnostic function is the same. In this case, we found the LOESS fit and linear fit differ a lot, which indicating using a quadratic function.

```
inter.raw(Y = "Y", D = "D", X = "X", data = s3, Ylabel = "Outcome", Dlabel = "Treatment", Xlabel="Moderator")
## $graph
```



4.2 Binning Plots

Interflex also provides binning plot diagnostics. The `nbins` option sets the number of bins. The default number of bins is 3, and equal-sized bins are created based on the distribution of the moderator. The red vertical lines are the CI of the maginal effect at the evaluation point of each bin (default option is the median value).

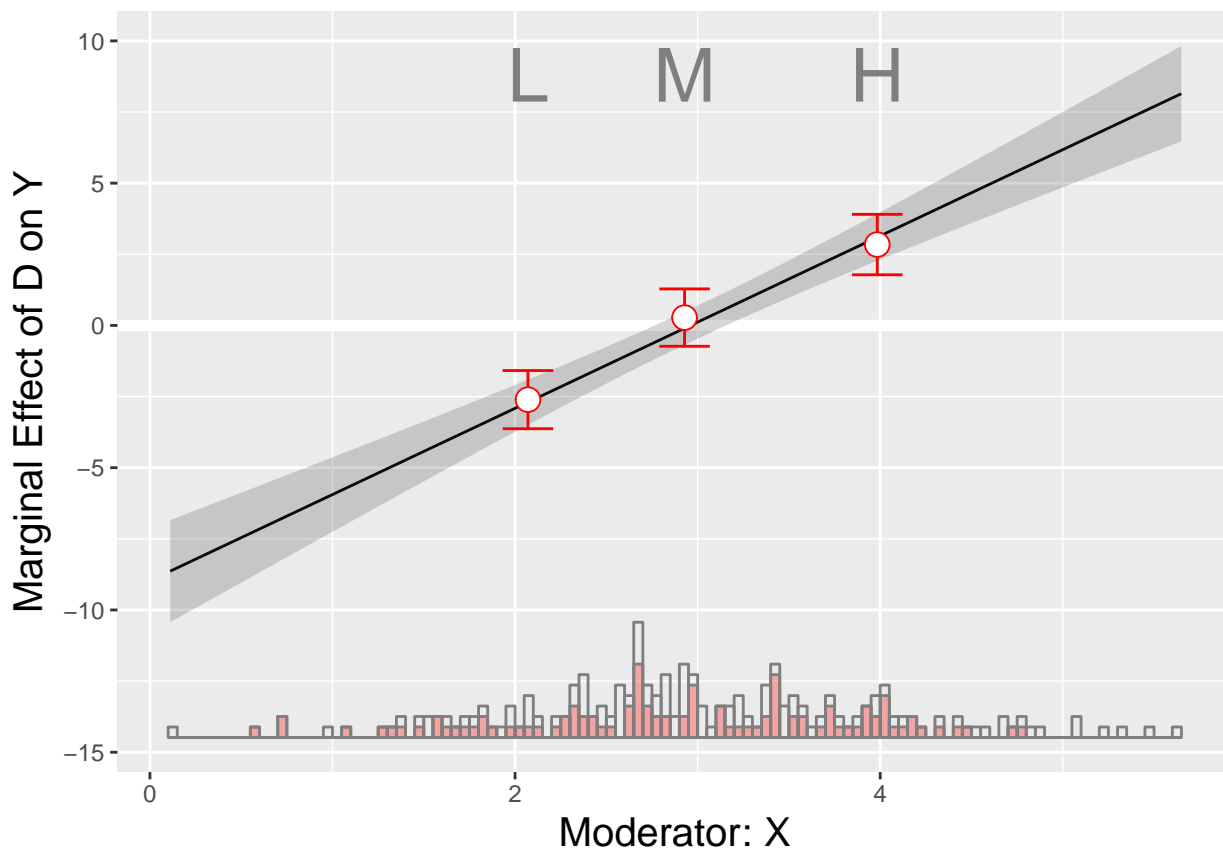
There are four options for the choice of the variance estimator: `vartype="homoscedastic"`, `"robust"`, `"cluster"`, and `"pcse"`. The default option is `"homoscedastic"`.

In the first case

```
inter.binning(Y = "Y", D = "D", X = "X", Z = "Z1", data = s1, vartype = "robust")
```

```
## $est.binning
##               x0      coef      se CI_lower CI_upper
## X: [0.111,2.61] 2.0706 -2.6080 0.5181 -3.6300 -1.5860
## X: (2.61,3.41] 2.9284  0.2749 0.5109 -0.7330  1.2828
## X: (3.41,5.65] 3.9839  2.8421 0.5382  1.7804  3.9038
##
## $binary.treatment
## [1] 1
##
## $bin.size
##      1      2      3
## 0.335 0.330 0.335
##
```

```
## $treat.variation.byGroup
## [1] 0.9853 1.0788 0.8742
##
## $X.Lkurtosis
## [1] 0.1441
##
## $correctOrder
## [1] TRUE
##
## $p.twosided
## p.1v2 p.2v3 p.1v3
## 1e-04 7e-04 0e+00
##
## $p.wald
## [1] 0.1135
##
## $graph
```



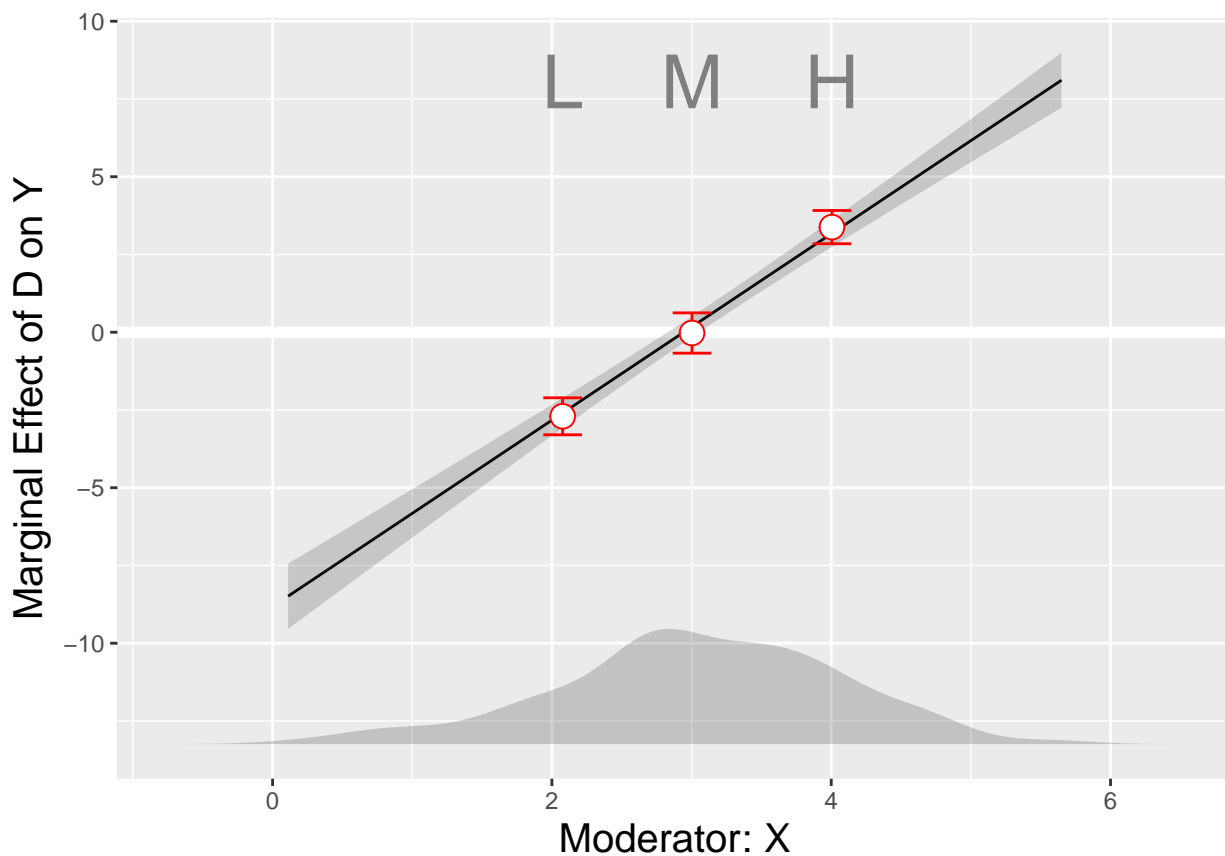
In the second continuous case

```
inter.binning(Y = "Y", D = "D", X = "X", Z = "Z1", data = s2,Xdistr = "density")
```

```
## $est.binning
##          x0      coef      se CI_lower CI_upper
## X: [0.111,2.67] 2.0768 -2.7030 0.3014 -3.2976 -2.1084
## X: (2.67,3.53]  3.0033 -0.0250 0.3284 -0.6727  0.6228
## X: (3.53,5.65]  4.0052  3.3798 0.2710  2.8452  3.9143
##
## $binary.treatment
```



```
## [1] 0
##
## $bin.size
##      1      2      3
## 0.335 0.330 0.335
##
## $treat.variation.byGroup
## [1] 0.0892 0.0754 0.1197
##
## $X.Lkurtosis
## [1] 0.1392
##
## $correctOrder
## [1] TRUE
##
## $p.twosided
## p.1v2 p.2v3 p.1v3
##      0      0      0
##
## $p.wald
## [1] 0.6368
##
## $graph
```



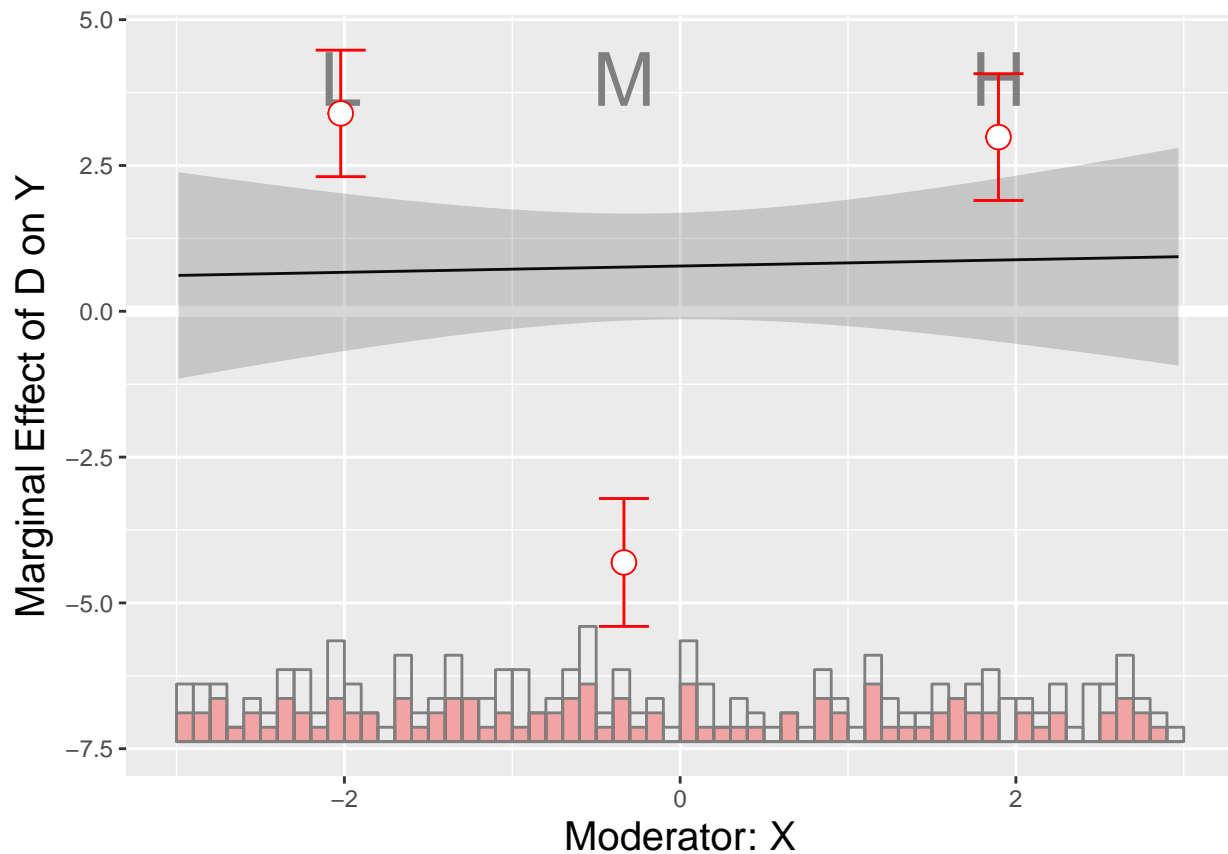
And the nonlinear case

```
inter.binning(Y = "Y", D = "D", X = "X", Z = "Z1", data = s3)
```

```

## $est.binning
##           x0      coef      se CI_lower CI_upper
## X: [-2.99,-1.12] -2.0221  3.3927 0.5496   2.3084   4.4769
## X: (-1.12,0.878] -0.3349 -4.3067 0.5558  -5.4032  -3.2102
## X: (0.878,2.97]   1.8963  2.9867 0.5509   1.9000   4.0734
##
## $binary.treatment
## [1] 1
##
## $bin.size
##      1      2      3
## 0.335 0.330 0.335
##
## $treat.variation.byGroup
## [1] 0.8231 1.0154 1.1103
##
## $X.Lkurtosis
## [1] -0.0014
##
## $correctOrder
## [1] FALSE
##
## $p.twosided
##      p.1v2  p.2v3  p.1v3
## 0.0000 0.0000 0.6019
##
## $p.wald
## [1] 0
##
## $graph

```

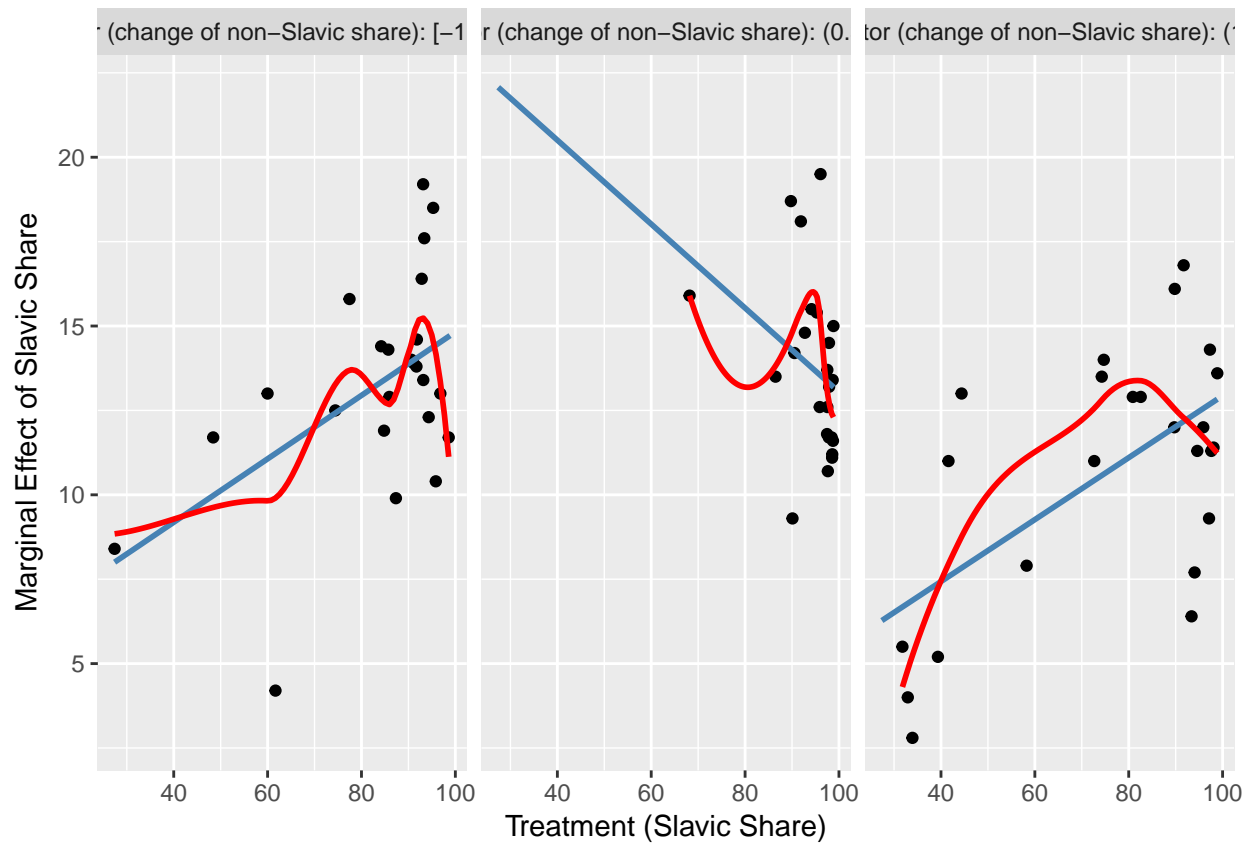


5. Actual example: Slavic Share Voting

Let's replicate the Slavic share voting example shown in Matt Golder's webpage, because it looks terrible.

```
library(foreign)
mydata <- read.dta('https://github.com/haowang666/P0S603-Lab/blob/master/Lab%207/alexseev.dta?raw=true')
lm <- lm(xenovote ~ slavicshare*changenonslav+
  inc9903 + eduhi02 + unemp02 + apt9200 +vsall03 +brdcont, data=mydata)
inter.raw(Y = "xenovote", D = "slavicshare",
  X = "changenonslav", data = mydata,
  Ylabel = "Marginal Effect of Slavic Share",
  Dlabel = "Treatment (Slavic Share)",
  Xlabel="Moderator (change of non-Slavic share)")

## $graph
```

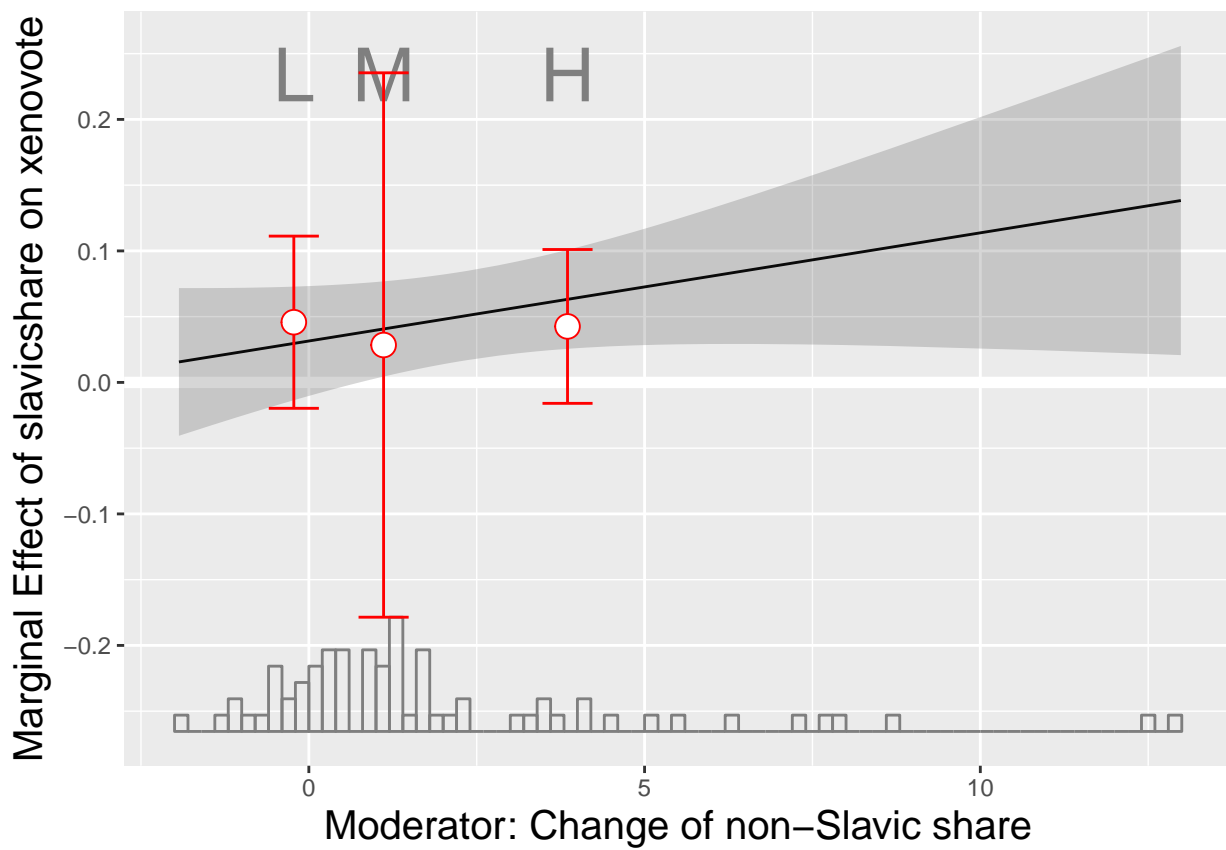


The raw plot shows that the liner fitting is very poor at medium and high moderator level.

```
inter.binning(Y = "xenovote", D = "slavicshare",
  X = "changenonslav",
  Z = c("inc9903", "eduhi02", "unemp02", "apt9200",
    "vsall03", "brdcont"),
  data = mydata,
  Xlabel="Change of non-Slavic share", na.rm = TRUE)
```

```
## $est.binning
##
##          x0    coef    se CI_lower
## Change of non-Slavic share: [-1.93,0.396] -0.2235 0.0458 0.0327 -0.0197
## Change of non-Slavic share: (0.396,1.66]  1.1134 0.0284 0.1032 -0.1785
## Change of non-Slavic share: (1.66,13]     3.8524 0.0426 0.0292 -0.0159
##
##          CI_upper
## Change of non-Slavic share: [-1.93,0.396]  0.1113
## Change of non-Slavic share: (0.396,1.66]   0.2354
## Change of non-Slavic share: (1.66,13]      0.1011
##
## $binary.treatment
## [1] 0
##
## $bin.size
##      1      2      3
## 0.3333 0.3333 0.3333
##
## $treat.variation.byGroup
## [1] 0.0439 0.0049 0.1065
```

```
##
## $X.Lkurtosis
## [1] 0.27
##
## $correctOrder
## [1] FALSE
##
## $p.twosided
## p.1v2 p.2v3 p.1v3
## 0.8754 0.8973 0.9413
##
## $p.wald
## [1] 0.6777
##
## $graph
```



Can't be worse!