Answers for Plotting

Hao Wang

March 20, 2017

1. Load essential packages

```
library(VGAM)
library(car)
library(MASS)
library(effects)
library(ggplot2)
library(Zelig)
library(ZeligChoice)
library(readstata13)
play<- read.dta13('https://github.com/haowang666/POS603-Lab/blob/master/Lab%206/stdSingh.dta?raw=true')
```

2. Run Zelig estimates

```
zlogit <- zelig(voted ~severity*age + polinfrel + income + efficacy + partyID +
                  dist_magnitude + enep + vicmarg_dist +
                  parliamentary + development, model ="logit", data=play, cite = FALSE)
summary(zlogit)
```

```
## Model:
##
## Call:
## z5$zelig(formula = voted ~ severity * age + polinfrel + income +
##     efficacy + partyID + dist_magnitude + enep + vicmarg_dist +
##     parliamentary + development, data = play)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.5291   0.2564   0.4129   0.5869   1.7531
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)      2.105223   0.012538 167.911  < 2e-16
## severity         0.637247   0.020068  31.754  < 2e-16
## age              0.303940   0.011051  27.504  < 2e-16
## polinfrel        0.333097   0.010654  31.264  < 2e-16
## income           0.220452   0.010350  21.300  < 2e-16
## efficacy         0.362777   0.009043  40.116  < 2e-16
## partyID          0.424113   0.010723  39.553  < 2e-16
## dist_magnitude   0.185052   0.012521  14.779  < 2e-16
## enep            -0.107988   0.014018  -7.703 1.32e-14
## vicmarg_dist    -0.035428   0.010394  -3.408 0.000653
## parliamentary    0.137085   0.011165  12.278  < 2e-16
## development      0.311358   0.012371  25.168  < 2e-16
```

```
## severity:age   -0.119769   0.016218  -7.385 1.52e-13
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 77479  on 92740  degrees of freedom
## Residual deviance: 67443  on 92728  degrees of freedom
## AIC: 67469
##
## Number of Fisher Scoring iterations: 6
##
## Next step: Use 'setx' method
```

Check Ranges

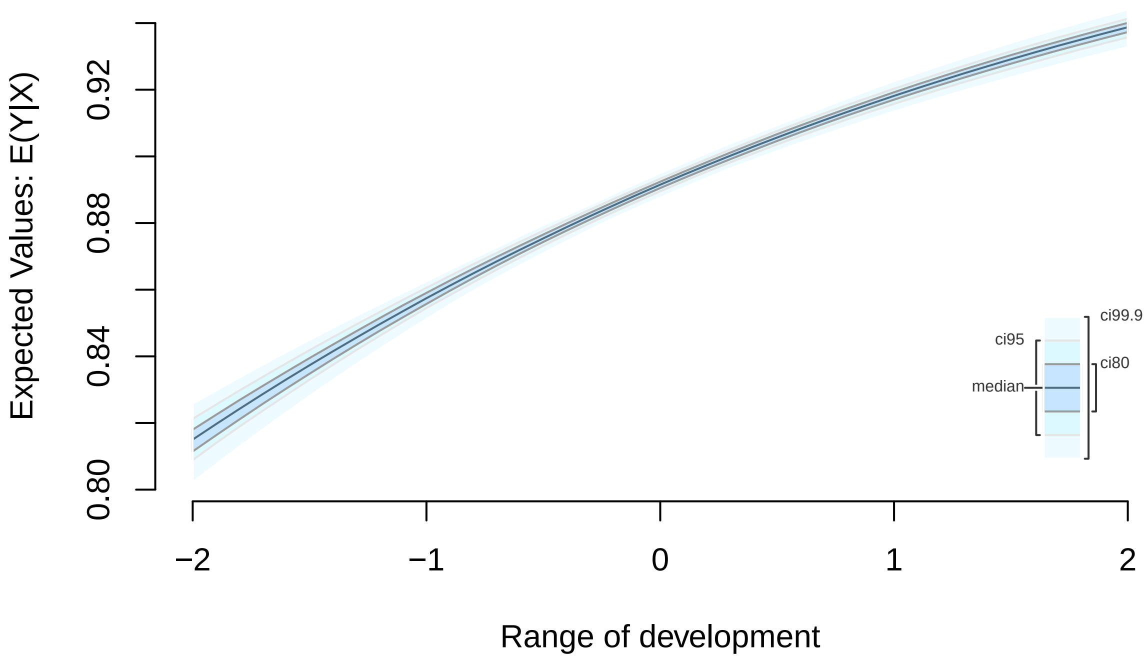
```r
#check range
summary(play$development)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -1.8040 -0.7614  0.2206  0.0000  0.7541  1.9530
```

Create data and run simulation

```r
#41 Xs
x.sim <- setx(zlogit, development = seq(from = -2, to = 2, by = 0.1))
s.out <- sim(zlogit, x = x.sim)
```

```r
#plot(s.out)
ci.plot(s.out, qi ="ev")
```



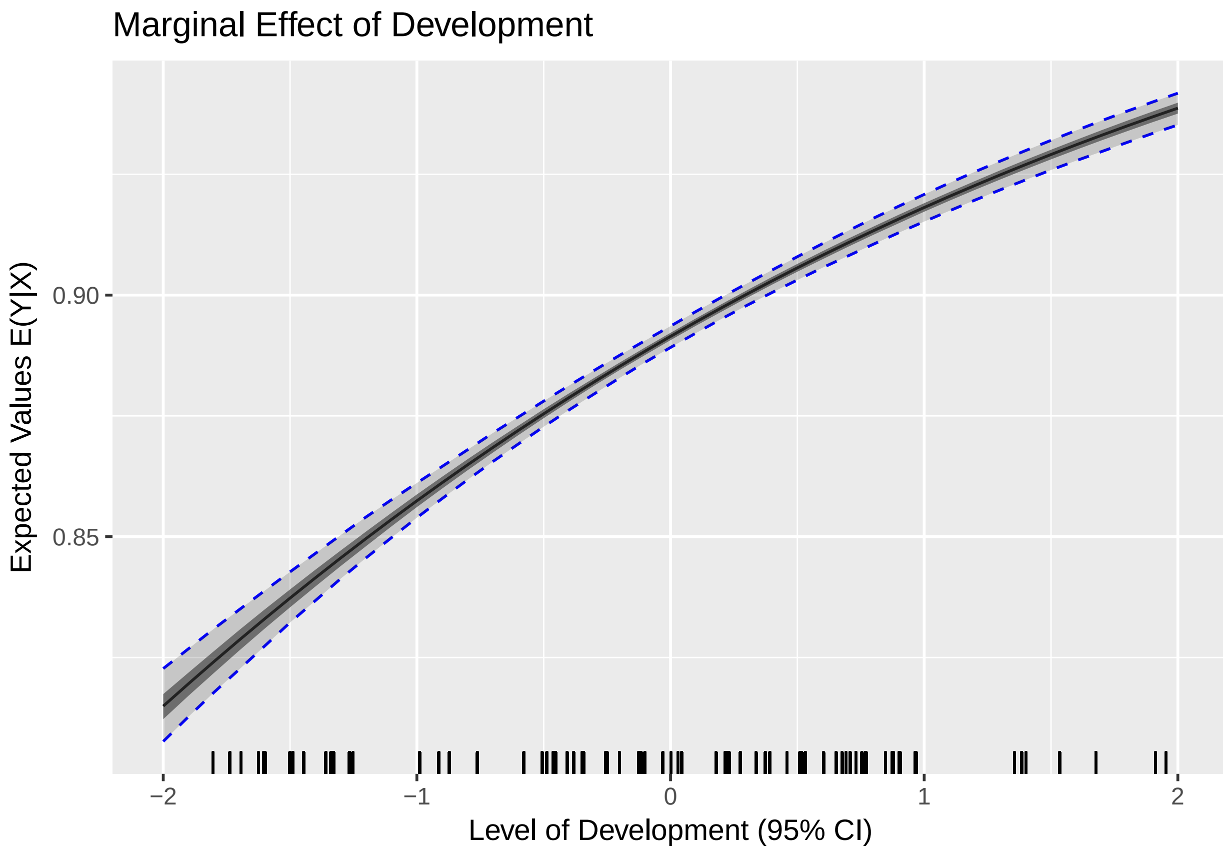
If you are unhappy with the default plot...

We need to extract info from zelig simulation

```
## Ranges of values
development = seq(from = -2, to = 2, by = 0.1)

# Extract expected values from simulations
# ev is expected values
phat<- s.out$get_qi(qi='ev', xvalue = 'range')
# collapse list to dataframe
phat <- as.data.frame(matrix(unlist(phat), nrow =1000))
# apply function to column
a<- apply(phat, 2, quantile, probs = c(0.025, 0.975, 0.25, 0.75))
low <- a[1,]
high <- a[2,]
qt.1 <- a[3,]
qt.3 <- a[4,]
fit <- apply(phat, 2, mean)
plotdata <- as.data.frame(cbind(low, high, fit, qt.1, qt.3))
plotdata$development <- development

#in ggplot2
ggplot() +
  geom_line(data = plotdata, aes(x = development, y =fit)) +
  geom_line(data = plotdata, aes(x = development, y =high), linetype="dashed", color="blue") +
  geom_line(data = plotdata, aes(x = development, y =low), linetype="dashed", color="blue") +
  geom_rug(data = play, aes(x= development)) +
  xlab("Level of Development (95% CI)") +
  ylab("Expected Values E(Y|X)") +
  ggtitle("Marginal Effect of Development")+
  geom_ribbon(data=plotdata, aes(x=development, ymin=low, ymax=high), alpha=0.2) +
  geom_ribbon(data=plotdata, aes(x=development, ymin=qt.1, ymax=qt.3), alpha=0.6)
```



3. Or use vanilla GLM code

```
glm <-  glm(formula = voted ~severity*age + polinfrel + income
                 + efficacy + partyID +
                 dist_magnitude + enep + vicmarg_dist +
                 parliamentary + development,
             data = play, family = binomial(link = "logit"))
development<-seq(-2,2,by=.1)
attach(play)
newdata <- as.data.frame(cbind(1,development,mean(severity),
                            mean(age),mean(polinfrel),mean(income),
                            mean(efficacy),mean(partyID),
                            mean(vicmarg_dist),mean(parliamentary),
                            mean(dist_magnitude),mean(enep)))

colnames(newdata)<-c("constant","development",
                     "severity","age","polinfrel","income",
                     "efficacy","partyID","vicmarg_dist",
                     "parliamentary","dist_magnitude","enep")
#response offers expected value
pr <- predict(glm, newdata = newdata, se.fit=TRUE, type="response")
#what if you missed the command type="reaponse"?
#you get linear prediction results
pr2 <- predict(glm, newdata = newdata)
```

```r
fit2 <- 1/(1+exp(- pr2))
cor(pr$fit, fit2)
```

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

```r
detach(play)
```

Plot it

```r
plotdata <- as.data.frame(pr$fit)
colnames(plotdata)<-c("fit")
plotdata$se <- pr$se.fit
plotdata$low <- plotdata$fit - 1.96*plotdata$se
plotdata$high <- plotdata$fit + 1.96*plotdata$se
plotdata$development <- development

ggplot() +
  geom_line(data = plotdata, aes(x = development, y =fit)) +
  geom_line(data = plotdata, aes(x = development, y =high), linetype="dashed", color="blue") +
  geom_line(data = plotdata, aes(x = development, y =low), linetype="dashed", color="blue") +
  geom_rug(data = play, aes(x= development)) +
  xlab("Level of Development (95% CI)") +
  ylab("Expected Values E(Y|X)") +
  ggtitle("Marginal Effect of Development")+
  geom_ribbon(data=plotdata, aes(x=development, ymin=low, ymax=high), alpha=0.5)
```