Homework 5

BBC: Bichay, Bovee, Coeman

March 22, 2018

Substantive Effects Simulations

First we read in the data files needed for the model.

```
dyad <- read.table("/Users/nickbichay/Desktop/ /aPLS 900/Midterm/Dyadicdata.tab.tsv", sep="\t", header=

polity <- read.csv("/Users/nickbichay/Desktop/ /aPLS 900/Midterm/p4v2016.csv", header = TRUE, stringsAs

lji <- read.table("/Users/nickbichay/Desktop/ /aPLS 900/Midterm/LJI-estimates-20140422.tab.tsv", sep="\
```

Second, we merge the datasets, keeping only the variables of interest. We also create a lagged-DV.

```
# subset only US and variables of interest
dyad <- dyad[dyad$ccode1==2,c("ccode2", "year", "absidealdiff")]

### merge in institutional model variables

# subset only variables of interest
polity <- polity[,c("ccode", "year", "polity2")]
lji <- lji[,c("ccode", "year", "LJI")]

# gen dummies
polity$democracy <- ifelse(polity$polity2 >= 6, 1, 0)
polity$autocracy <- ifelse(polity$polity2 <= -6, 1, 0)

# merge, keeping only observations in both datasets
dyad <- merge(dyad, polity, by.x=c("ccode2","year"), by.y=c("ccode","year"), all=FALSE)
dyad <- merge(dyad, lji, by.x=c("ccode2","year"), by.y=c("ccode","year"), all=FALSE)

# create lag
library(DataCombine)
dyad <- slide(dyad, Var = "absidealdiff", GroupVar = "ccode2", slideBy = -1)
```

```
##
## Remember to order dyad by ccode2 and the time variable before running.
```

```
##
## Lagging absidealdiff by 1 time units.
```

```
# rename ccode and lagged dv
colnames(dyad)[colnames(dyad) == "ccode2"] <- "ccode"
colnames(dyad)[colnames(dyad) == "absidealdiff-1"] <- "lagdv"

# drop NAs
dyad <- na.omit(dyad)
```

Here we run the model, save the coefficents, and create a vector of coefficients based on their variance

```r
inst_lm <- lm(absidealdiff ~  democracy + autocracy + LJI + lagdv, data=dyad)

library(MASS)

betaMean <- coef(inst_lm)
betaDist <- vcov(inst_lm)
betaDraws <- mvrnorm(1000, betaMean, betaDist)
```

Next we simulate changes in each variable and create a graph to show the change as the variable of interest changes from its min to its max value, while the remaining variables stay at their mean (or, for the democracy/autocracy dummies: at zero)

```r
### sim for democracy dummy

# create matrix of hypothetical x values to simulate over
x1Values = c(0, 1)
scenario = cbind(intercept=1, democracy=x1Values, autocracy=0, LJI=mean(dyad$LJI), lagdv=mean(dyad$lagd

# generate predictions

library(dplyr)
```

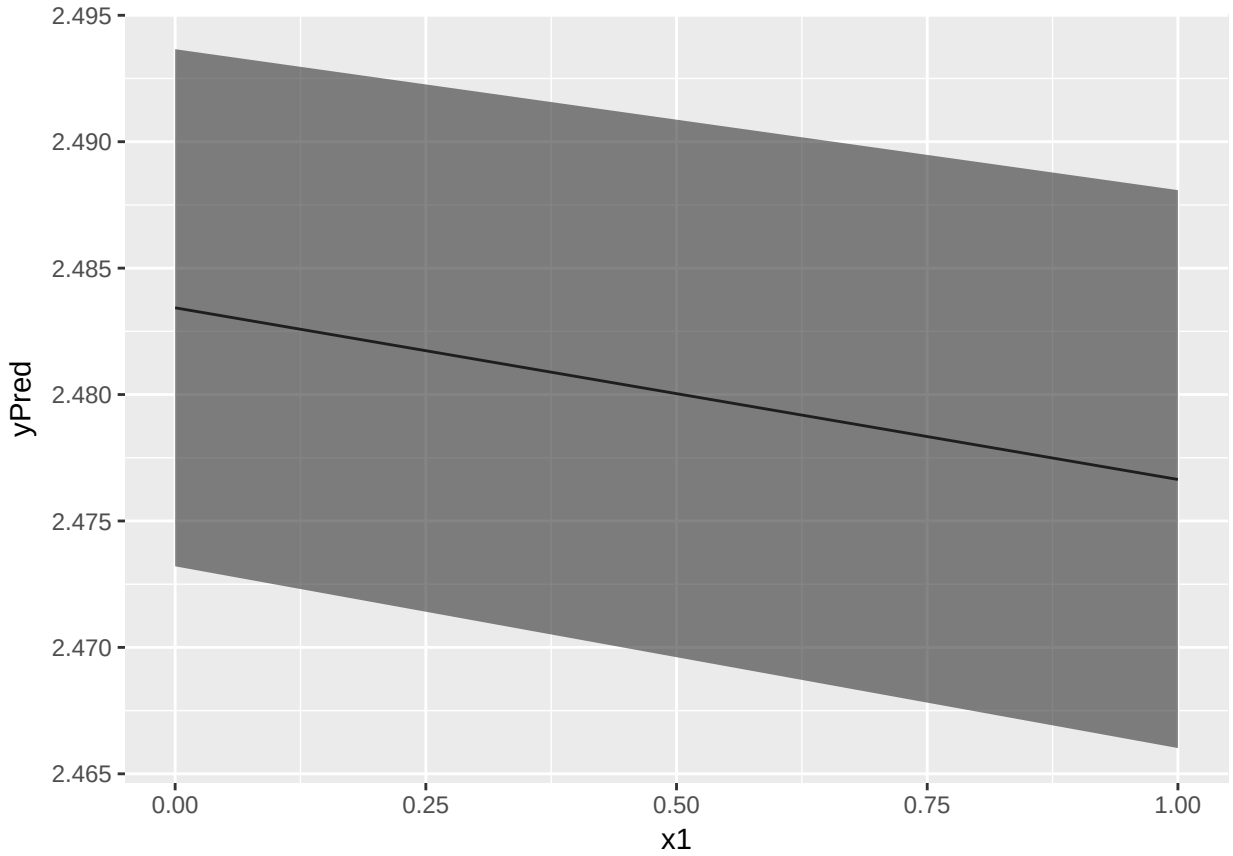
```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:MASS':
##
##     select
```

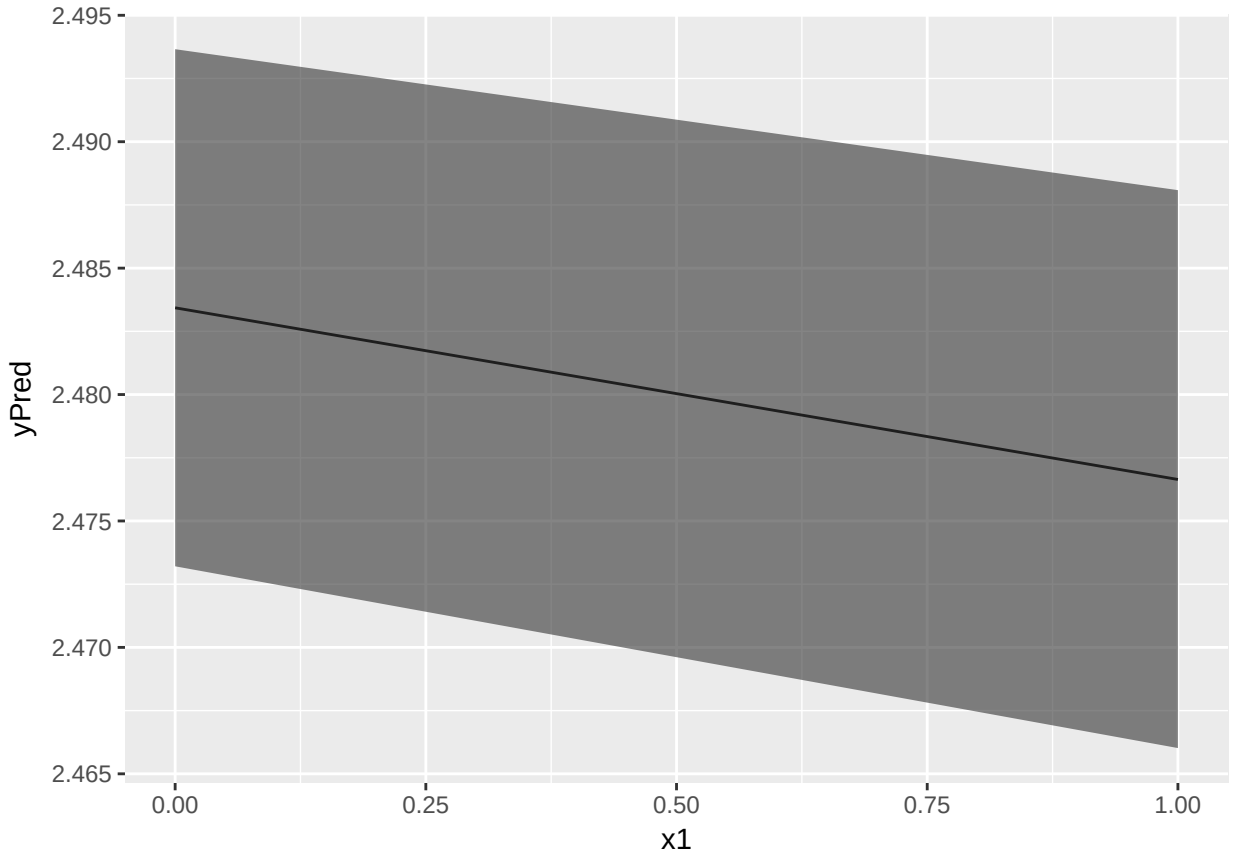
```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

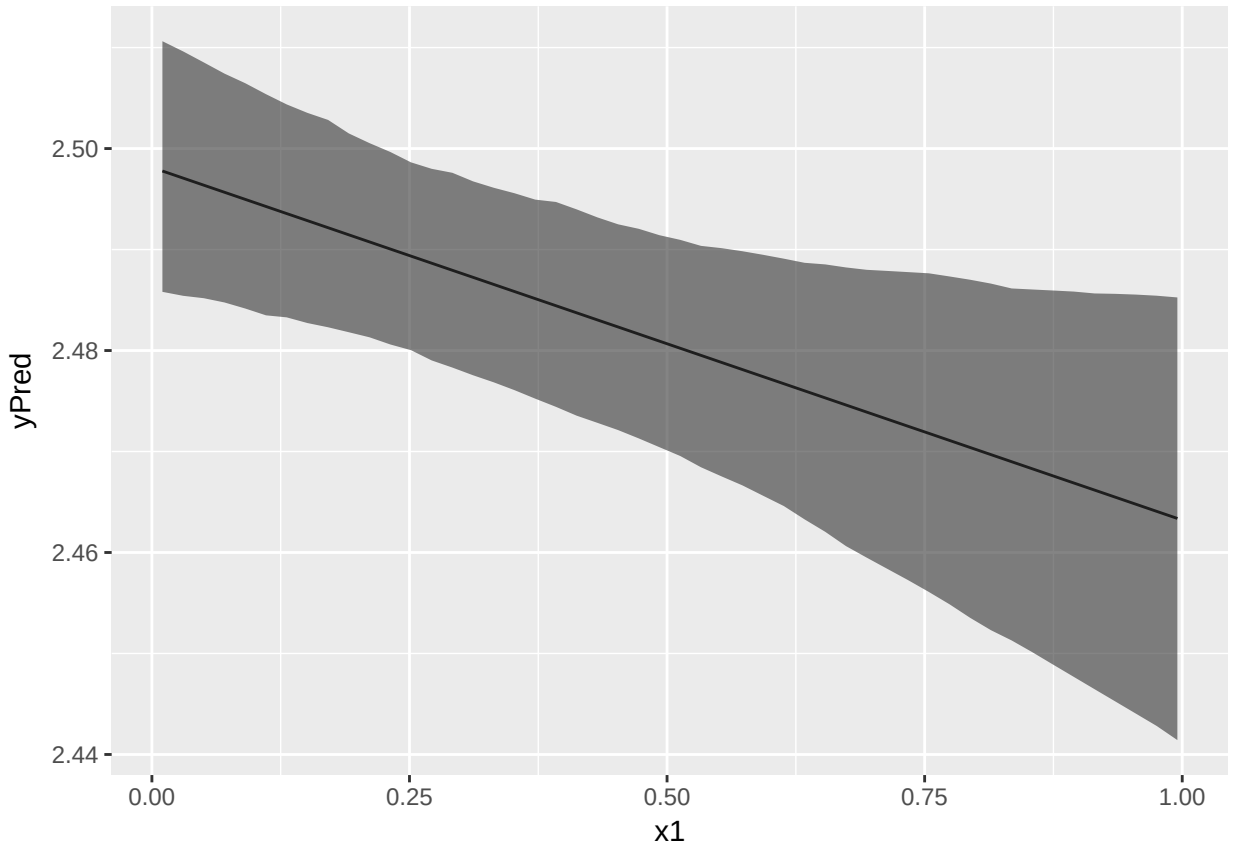
```r
yPred  <-  scenario %*% betaMean

yPredUncert <- scenario %*% t(betaDraws)
yPredInt  <-  apply(yPredUncert, 1, function(x){
    quantile(x, c(0.025, 0.975), na.rm=TRUE) }) %>% t()

simAnalysis  <- data.frame(x1=x1Values, yPred=yPred, yPredInt)
names(simAnalysis)[3:4] = c('q95lo','q95hi')

# graph

library(ggplot2)

ggplot(simAnalysis, aes(x=x1, y=yPred)) +
    geom_line() +
    geom_ribbon(aes(ymin=q95lo, ymax=q95hi), alpha=.6)
```




```
### sim for autocracy dummy

# create matrix of hypothetical x values to simulate over
x1Values_aut = c(0, 1)
scenario_aut = cbind(intercept=1, autocracy=x1Values_aut, democracy=0, LJI=mean(dyad$LJI), lagdv=mean(dy

# generate predictions

yPred_aut  <-  scenario_aut %*% betaMean

yPredUncert_aut <- scenario_aut %*% t(betaDraws)
yPredInt_aut  <-  apply(yPredUncert_aut, 1, function(x){
    quantile(x, c(0.025, 0.975), na.rm=TRUE) }) %>% t()

simAnalysis_aut  <- data.frame(x1=x1Values_aut, yPred=yPred_aut, yPredInt_aut)
names(simAnalysis_aut)[3:4] = c('q95lo','q95hi')

# graph

ggplot(simAnalysis_aut, aes(x=x1, y=yPred)) +
    geom_line() +
    geom_ribbon(aes(ymin=q95lo, ymax=q95hi), alpha=.6)
```




```
### sim for lji

# create matrix of hypothetical x values to simulate over
x1Values_lji = seq(min(dyad$LJI), max(dyad$LJI), length.out=50)
scenario_lji = cbind(intercept=1, autocracy=0, democracy=0, LJI=x1Values_lji, lagdv=mean(dyad$lagdv) )

# generate predictions

yPred_lji  <-  scenario_lji %*% betaMean

yPredUncert_lji <- scenario_lji %*% t(betaDraws)
yPredInt_lji  <-  apply(yPredUncert_lji, 1, function(x){
    quantile(x, c(0.025, 0.975), na.rm=TRUE) }) %>% t()

simAnalysis_lji  <- data.frame(x1=x1Values_lji, yPred=yPred_lji, yPredInt_lji)
names(simAnalysis_lji)[3:4] = c('q95lo','q95hi')

# graph

ggplot(simAnalysis_lji, aes(x=x1, y=yPred)) +
    geom_line() +
    geom_ribbon(aes(ymin=q95lo, ymax=q95hi), alpha=.6)
```



Graph 1

Graph 2