Tutorial 13: Autocorrelation, Fixed, Random, Mixed Effects, and Causal Inference Techniques

Jan Vogler (jan.vogler@duke.edu)

November 25, 2016

Today's Agenda

1. Autocorrelation
2. Fixed, random, and mixed-effects regression
3. Differences-in-differences
4. Regression discontinuity
5. Spatial lag models
6. Learning R - what to do next?

1. Autocorrelation

Let us first load the LDC dataset.

```r
setwd("C:/Users/Jan/OneDrive/Documents/GitHub/ps630_lab/ps630_f16")
library(foreign)
LDC = read.dta("LDC_IO_replication.dta")
```

We use a regression that we have already seen in the past to illustrate the phenomenon of autocorrelation. This is our standard regression in which we analyze the impact of democratization on tariff levels.

```r
lm_basic = lm(newtar ~ l1polity + l1gdp_pc + l1lnpop + l1ecris2 + l1bpc1 + l1avnewtar +
    factor(ctylabel) - 1, data = LDC)
# summary(lm_main)
```

There might be autocorrelation in our model because our units at time t and our units at time t-1 are likely to have similar values for the dependent and independent variables. Let us regress the residuals of our models on the residuals at time t-1:

```r
res_t0 = lm_basic$resid
res_t1 = c(lm_basic$resid[2:length(lm_basic$resid)], NA)
res_data = as.data.frame(cbind(res_t1, res_t0))
head(res_data)
```

```
##          res_t1      res_t0
## 15 -7.42623616  6.99807104
## 16 -3.88875485 -7.42623616
## 17  4.07686936 -3.88875485
## 18  0.04671752  4.07686936
## 19 -2.67290721  0.04671752
## 24  5.54727523 -2.67290721
```

```r
lm_res = lm(res_t1 ~ res_t0, data = res_data)
summary(lm_res)
```

```
##
## Call:
## lm(formula = res_t1 ~ res_t0, data = res_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -22.813  -2.691  -0.068   2.451  31.918
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.006588   0.222448   -0.03    0.976
## res_t0       0.544863   0.031016   17.57   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.018 on 730 degrees of freedom
##   (1 observation deleted due to missingness)
## Multiple R-squared:  0.2971, Adjusted R-squared:  0.2962
## F-statistic: 308.6 on 1 and 730 DF,  p-value: < 2.2e-16
```

What we can see from this simple regression is that there is a high level of autocorrelation in our errors. Note that this code can easily be extended to include multiple lags of our residuals, which you might want to do to check for autocorrelation for several observations.

To address this problem, if we have enough datapoints available, we can create lags of our dependent variable (and other variables) and then include them in a new regression. The *DataCombine* package is designed for data management, especially time series and panel data. It allows us to easily generate lags of our dependent variable.

install.packages("DataCombine")

```r
library(DataCombine)

# summary(LDC)
```

We first define which variables we want to lag and we create a vector that includes prefixes for our lagged variables so that we can easily distinguish them from the non-lagged variables.

```r
# Choose the variables you want to lag
toLag = c("newtar","polityiv_update2")

# Define vector with the lag numbers
numberLag = c("lag1_","lag2_","lag3_")

# For loop to lag the data
for (i in 1:3){
  for (lagVar in toLag){
    LDC=slide(LDC, Var=lagVar, # Specify variable to lag
                 TimeVar="date", # Specificy time variable
                 GroupVar="ctylabel", # Unit variable
```

```
                  NewVar=paste0(numberLag[i],lagVar), # Name of new variable
                  slideBy = -i, # Lag by how many units, minus -> past
                  keepInvalid = FALSE, # Keep observations for which no lag can be created
                  reminder = TRUE) # Remind you to order the data by group variable
  }
}
```

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

##
## Lagging polityiv_update2 by 1 time units.

##
## Lagging newtar by 2 time units.

##
## Lagging polityiv_update2 by 2 time units.

##
## Lagging newtar by 3 time units.

##
## Lagging polityiv_update2 by 3 time units.
```

Let us now create a new model that contains three lags of the dependent variable.

```
lm_lag_dv = lm(newtar ~ lag1_newtar + lag2_newtar + lag3_newtar + l1polity +
    l1gdp_pc + l1lnpop + l1ecris2 + l1bpc1 + l1avnewtar + factor(ctylabel) -
    1, data = LDC)

# summary(lm_lag_dv)
```

How can we interpret the results of this model?

Also, there might be a time trend. Tariff levels might be moving up or down over time. If we can capture such a time trend with our model, it might reduce the autocorrelation in our errors.

```
lmTime = lm(newtar ~ lag1_newtar + lag2_newtar + lag3_newtar + l1polity + l1gdp_pc +
    l1lnpop + l1ecris2 + l1bpc1 + l1avnewtar + date + factor(ctylabel) - 1,
    data = LDC)

# summary(lmTime)
```

How would we interpret the results of our regression?

Let us use data on the United States from **Castellacci & Natera (2011)** to illustrate how we can correct for autocorrelation using Newey West Standard Errors. Our theory here is that the number of patents (Patents) that were awarded in the US is a function of economic openness (Openness) and education expenditures (EducExp). Economic openness could be associated with both more opportunities to create innovations and more economic pressure due to higher levels of competition.

```r
setwd("C:/Users/Jan/OneDrive/Documents/GitHub/ps630_lab/ps630_f16/w13/")
load("USdata.Rdata")
lm_patents = lm(Patents ~ Openness + EducExp, data = USdata)
summary(lm_patents)
```

```
##
## Call:
## lm(formula = Patents ~ Openness + EducExp, data = USdata)
##
## Residuals:
##        Min         1Q     Median         3Q        Max
## -5.224e-05 -3.213e-05 -1.214e-06  1.797e-05  7.204e-05
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.026e-05  1.100e-04   0.639    0.529
## Openness     1.534e-03  2.396e-04   6.404 8.76e-07 ***
## EducExp     -2.055e-05  1.859e-05  -1.106    0.279
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.615e-05 on 26 degrees of freedom
## Multiple R-squared:  0.6146, Adjusted R-squared:  0.585
## F-statistic: 20.73 on 2 and 26 DF,  p-value: 4.134e-06
```

How would we interpret this regression? Why might autocorrelation bias our results here?

Let us use Newey West SE which correct for autocorrelation.

DELETE THIS PART.

Credit to Sascha Riaz for this part of the code.

install.packages("sandwich")

install.packages("lmtest")

```r
library(sandwich)
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```r
coeftest(lm_patents, NeweyWest(lm_patents, lag = 3))
```

```
##
## t test of coefficients:
```

```
##
##                 Estimate  Std. Error t value Pr(>|t|)
## (Intercept)  7.0262e-05  3.9783e-04  0.1766  0.86118
## Openness     1.5342e-03  7.3953e-04  2.0745  0.04807 *
## EducExp     -2.0554e-05  5.3472e-05 -0.3844  0.70381
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

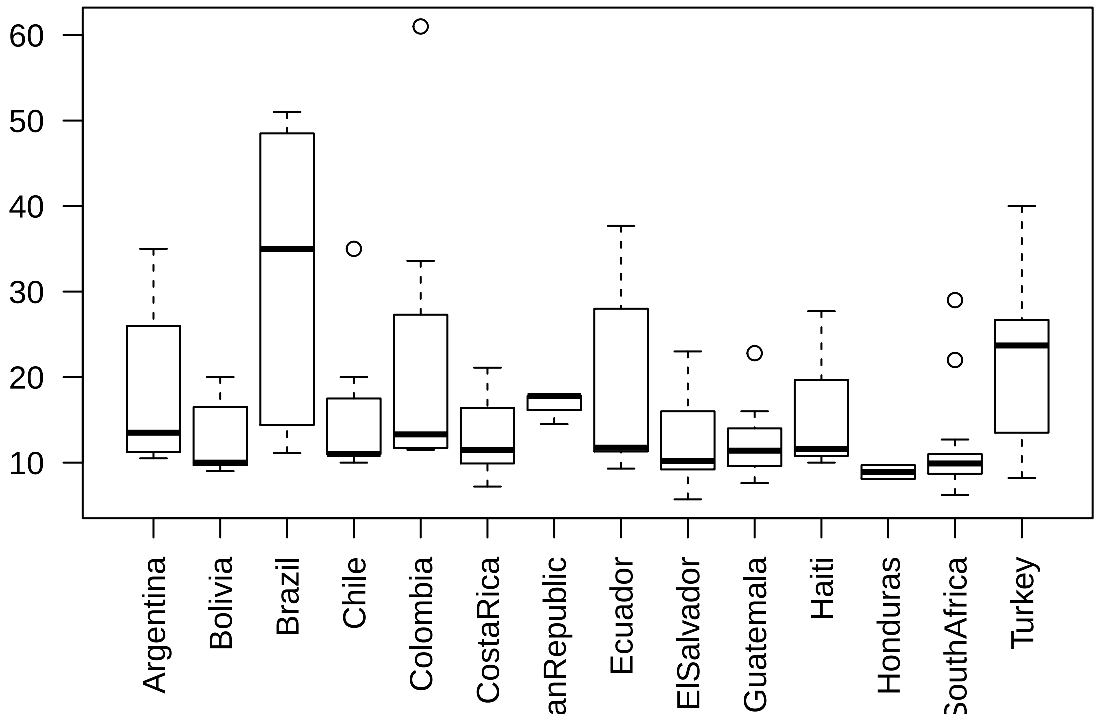
How did the regression results change using Newey West SE?

2. Fixed, random, and mixed-effects regression

Why would we use fixed effects? The main reason is that there might be many unobserved factors that contribute to heterogeneity across our units. For example, historical developments and country-specific peculiarities that we cannot easily measure (such as culture) could contribute to very different levels of, for instance, tariff levels. Let us illustrate the heterogeneity in tariff levels across countries using several plots.

Note that there are too many countries in our LDC dataset, so we will use an arbitrary subset of the data to demonstrate this.

```
LDC2 = LDC[1:450, ]
boxplot(newtar ~ ctylabel, par(las = 2), data = LDC2)
```



There are more plots to look at changes in variables over time for several units.

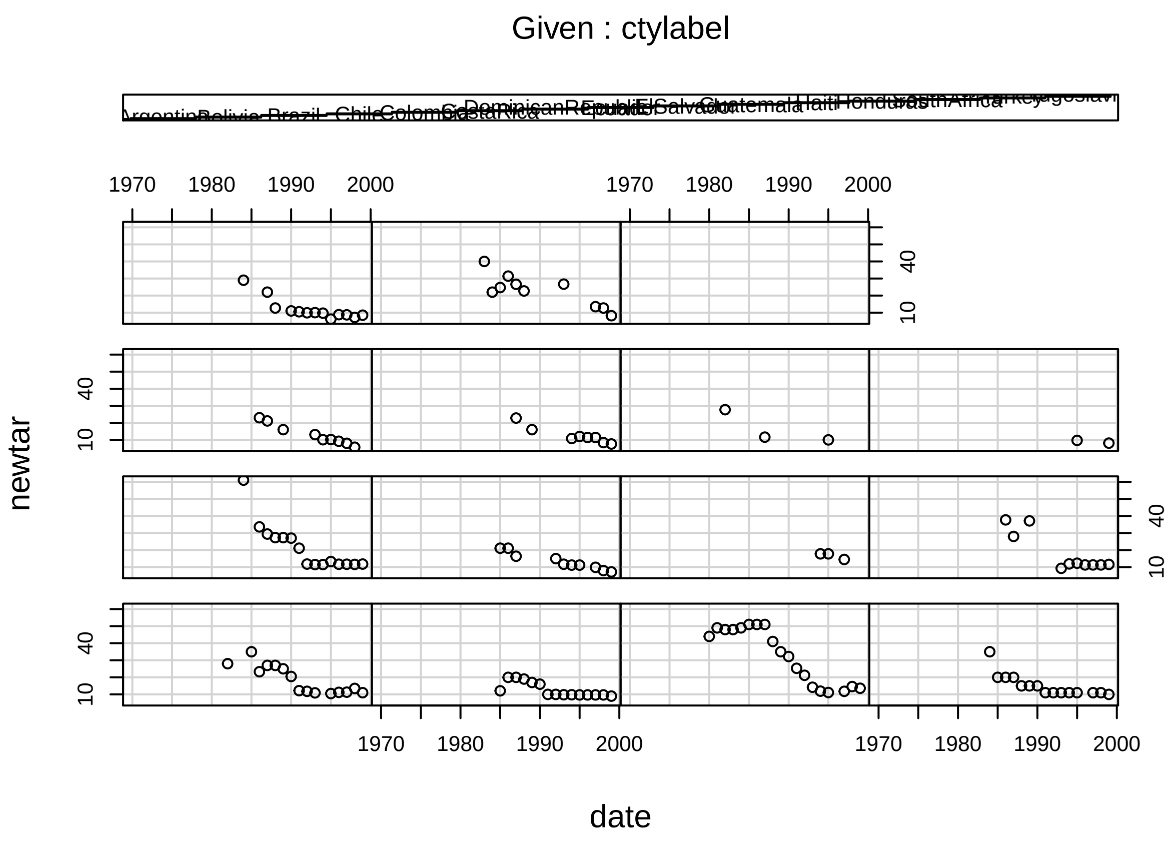
install.packages("gplots")

```
library(gplots)
```

```
##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##
##     lowess
```

```
coplot(newtar ~ date | ctylabel, data = LDC2)
```

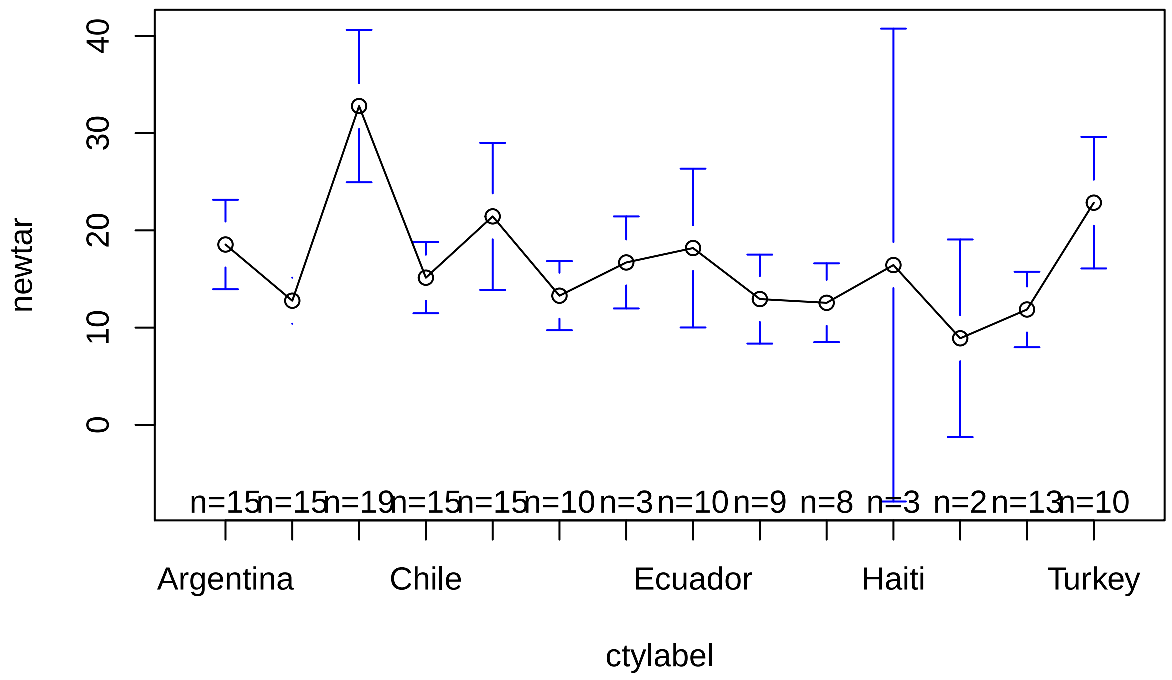


```
##
##  Missing rows: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 20, 21, 22, 23, 25, 26, 27, 31, 32, 33, 34
```

```
plotmeans(newtar ~ ctylabel, data = LDC2)
```

```
## Warning in arrows(x, li, x, pmax(y - gap, li), col = barcol, lwd = lwd, :
## zero-length arrow is of indeterminate angle and so skipped

## Warning in arrows(x, ui, x, pmin(y + gap, ui), col = barcol, lwd = lwd, :
## zero-length arrow is of indeterminate angle and so skipped
```



How would one implement fixed effects? We have seen fixed effects previously. We can simply integrate fixed effects by using the `factor()` command.

There are two ways to use fixed effects.

```
# Estimates an intercept for every unit but omits an intercept
lm_fe1 = lm(newtar ~ l1polity + l1gdp_pc + l1lnpop + l1avnewtar + factor(ctylabel) -
    1, data = LDC2)
# summary(lm_fe1)

# Estimates an intercept but omits Albania
lm_fe2 = lm(newtar ~ l1polity + l1gdp_pc + l1lnpop + l1avnewtar + factor(ctylabel),
    data = LDC2)
# summary(lm_fe2)
```

Is there a substantive difference between those two models? Does one of them bias our results?

Now let's introduce random effects (RE). The term "random effects" can mean different things depending on the context. Although there are different meanings depending on the context, let us first refer to "random effects" as merely reflecting a procedure to estimate the **intercept** for each unit. In order to do this, we use the very useful *lme4* package.

install.packages("lme4")

```
library(lme4)
```

```
## Loading required package: Matrix
```

```r
lm_re = lmer(newtar ~ l1polity + l1gdp_pc + l1lnpop + l1avnewtar + (1 | ctylabel),
    data = LDC2)
```

```
## Warning: Some predictor variables are on very different scales: consider
## rescaling
```

```r
summary(lm_re)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula:
## newtar ~ l1polity + l1gdp_pc + l1lnpop + l1avnewtar + (1 | ctylabel)
##    Data: LDC2
##
## REML criterion at convergence: 992.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -2.1325 -0.5084  0.0088  0.4241  4.5583
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  ctylabel (Intercept) 18.28    4.276
##  Residual             47.49    6.891
## Number of obs: 144, groups:  ctylabel, 14
##
## Fixed effects:
##               Estimate Std. Error t value
## (Intercept) -3.576e+01  2.251e+01  -1.589
## l1polity    -7.836e-01  1.643e-01  -4.771
## l1gdp_pc     2.834e-04  7.910e-04   0.358
## l1lnpop      2.175e+00  1.402e+00   1.551
## l1avnewtar   1.015e+00  1.115e-01   9.101
##
## Correlation of Fixed Effects:
##            (Intr) l1plty l1gdp_ l1lnpp
## l1polity   -0.122
## l1gdp_pc    0.414 -0.130
## l1lnpop    -0.986  0.060 -0.503
## l1avnewtar -0.115  0.257  0.172 -0.016
## fit warnings:
## Some predictor variables are on very different scales: consider rescaling
```

How would we interpret this output?

Let us look specifically at the distribution of the intercept across different units.

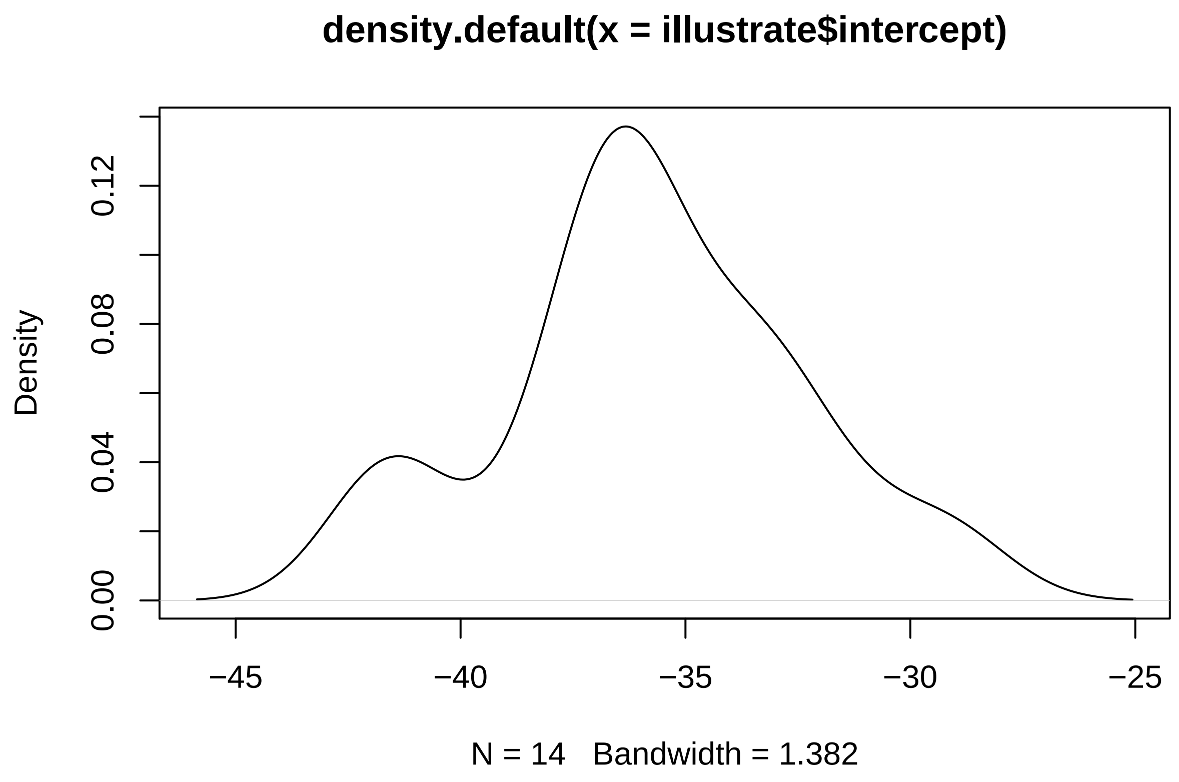
```r
coef(lm_re)$ctylabel
```

```
##                    (Intercept)   l1polity     l1gdp_pc l1lnpop l1avnewtar
## Argentina            -37.86805 -0.7836089 0.0002833564 2.17501   1.014965
## Bolivia              -36.43868 -0.7836089 0.0002833564 2.17501   1.014965
## Brazil               -29.21037 -0.7836089 0.0002833564 2.17501   1.014965
## Chile                  -41.27217 -0.7836089 0.0002833564 2.17501   1.014965
## Colombia               -33.11516 -0.7836089 0.0002833564 2.17501   1.014965
## CostaRica              -34.10698 -0.7836089 0.0002833564 2.17501   1.014965
## DominicanRepublic      -33.35111 -0.7836089 0.0002833564 2.17501   1.014965
## Ecuador                -31.67486 -0.7836089 0.0002833564 2.17501   1.014965
## ElSalvador             -36.41011 -0.7836089 0.0002833564 2.17501   1.014965
## Guatemala              -37.22574 -0.7836089 0.0002833564 2.17501   1.014965
## Haiti                  -36.42138 -0.7836089 0.0002833564 2.17501   1.014965
## Honduras               -36.25466 -0.7836089 0.0002833564 2.17501   1.014965
## SouthAfrica            -41.71392 -0.7836089 0.0002833564 2.17501   1.014965
## Turkey                 -35.58732 -0.7836089 0.0002833564 2.17501   1.014965
```

```
illustrate = coef(lm_re)$ctylabel

colnames(illustrate)[1] = "intercept"

plot(density(illustrate$intercept))
```



How does this distribution look like?

Fortunately, the *lme4* package allows us to also estimate models in which our coefficients are randomly distributed — a random coefficients model. As you learned in lecture, sometimes models that include both regular coefficients and random coefficients are labeled "mixed models". However, the definition may change depending on the context, so be careful about the use of this term.

```r
lm_mixed = lmer(newtar ~ l1polity + l1gdp_pc + l1lnpop + l1avnewtar + (1 + l1polity |
    ctylabel), data = LDC2)
```

```
## Warning: Some predictor variables are on very different scales: consider
## rescaling
```

```r
summary(lm_mixed)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula:
## newtar ~ l1polity + l1gdp_pc + l1lnpop + l1avnewtar + (1 + l1polity |
##     ctylabel)
##    Data: LDC2
##
## REML criterion at convergence: 968.1
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.7567 -0.5315 -0.0684  0.3404  4.9379
##
## Random effects:
##  Groups   Name        Variance Std.Dev. Corr
##  ctylabel (Intercept) 73.2280  8.5573
##           l1polity     0.4983  0.7059   -1.00
##  Residual             38.8861  6.2359
## Number of obs: 144, groups:  ctylabel, 14
##
## Fixed effects:
##               Estimate Std. Error t value
## (Intercept) -2.383e+01  1.319e+01  -1.807
## l1polity    -6.811e-01  3.091e-01  -2.204
## l1gdp_pc     1.217e-04  5.340e-04   0.228
## l1lnpop      1.368e+00  8.161e-01   1.676
## l1avnewtar   1.080e+00  9.952e-02  10.855
##
## Correlation of Fixed Effects:
##            (Intr) l1plty l1gdp_ l1lnpp
## l1polity   -0.106
## l1gdp_pc    0.342 -0.269
## l1lnpop    -0.958 -0.093 -0.411
## l1avnewtar -0.166  0.094  0.095 -0.017
## fit warnings:
## Some predictor variables are on very different scales: consider rescaling
```

How would we interpret these results? What is “mixed” in this model?

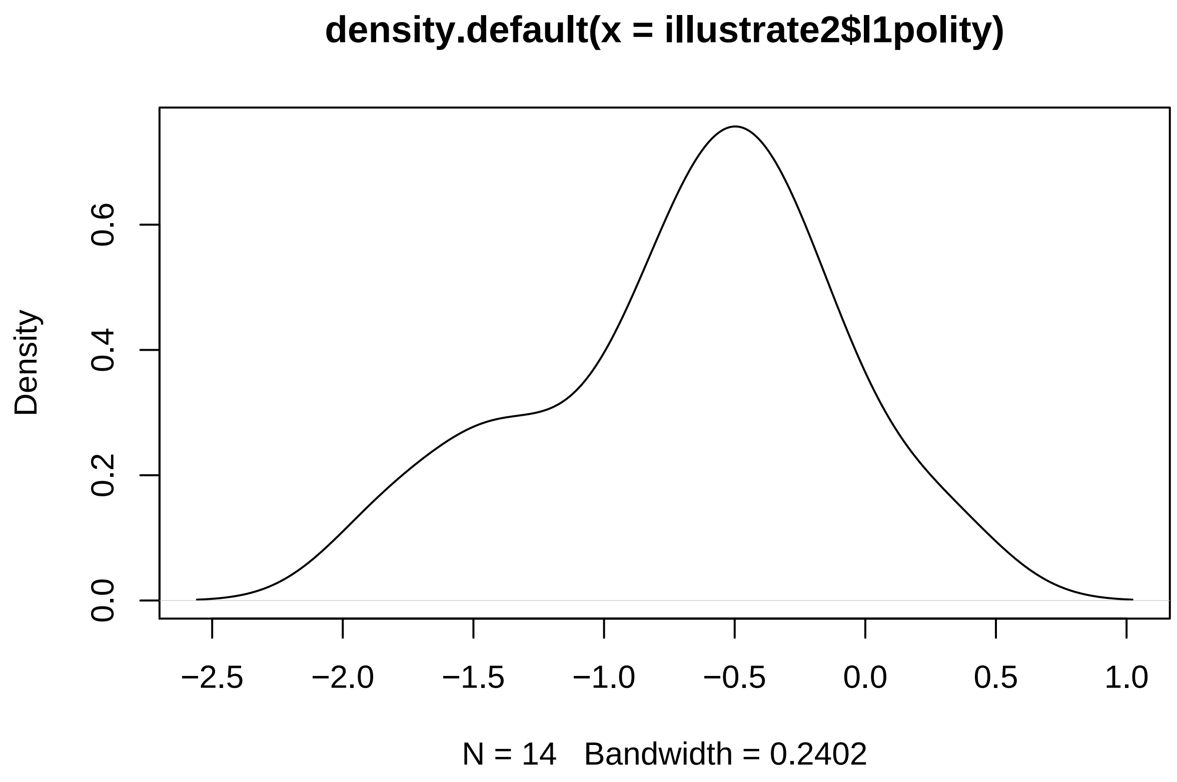
Let us look at how the coefficient changes from country to country.

```r
coef(lm_mixed)$ctylabel
```

```
##                     (Intercept)    l1polity     l1gdp_pc  l1lnpop l1avnewtar
## Argentina             -26.34511 -0.47379717 0.0001216576 1.367765   1.080279
## Bolivia              -29.89260 -0.18115530 0.0001216576 1.367765   1.080279
## Brazil                -9.81273 -1.83759798 0.0001216576 1.367765   1.080279
## Chile                -31.28192 -0.06654642 0.0001216576 1.367765   1.080279
## Colombia             -15.80662 -1.34314537 0.0001216576 1.367765   1.080279
## CostaRica            -23.14409 -0.73785767 0.0001216576 1.367765   1.080279
## DominicanRepublic    -19.75134 -1.01773482 0.0001216576 1.367765   1.080279
## Ecuador              -13.97779 -1.49401059 0.0001216576 1.367765   1.080279
## ElSalvador           -27.23384 -0.40048331 0.0001216576 1.367765   1.080279
## Guatemala            -27.92296 -0.34363610 0.0001216576 1.367765   1.080279
## Haiti                -24.56266 -0.62083638 0.0001216576 1.367765   1.080279
## Honduras             -25.81887 -0.51720791 0.0001216576 1.367765   1.080279
## SouthAfrica          -35.74701  0.30179134 0.0001216576 1.367765   1.080279
## Turkey               -22.34614 -0.80368292 0.0001216576 1.367765   1.080279
```

```
illustrate2 = coef(lm_mixed)$ctylabel

plot(density(illustrate2$l1polity))
```



How would we interpret these differences?

3. Differences-in-differences

What is differences-in-differences (diff-in-diff)?

When we use a diff-in-diff design, we have two groups of units and we are interested in how one of the two groups is affected by an “exogeneous” treatment. In order to estimate the effect of the treatment, we make the so-called “parallel trends” assumption, i.e. we believe that the variable we are interested in moves would have moved in the same fashion if it was not for the treatment. The “control group is not affected by the treatment. The name”differences-in-differences" stems from the fact that we observe a variable that moves over time (has differences over time) and we are interested in how this movement is different between units.

Note that although I use experimental language, such as the terms “exogeneous”, “treatment” and “control group”, a diff-in-diff does not meet the same rigorous standards as a controlled, randomized experiment.

The following content is from Kevin Goulding.

The original can be found here:

https://thetarzan.wordpress.com/2011/06/20/differences-in-differences-estimation-in-r-and-stata/

The following plot illustrates the concept of differences-in-differences:

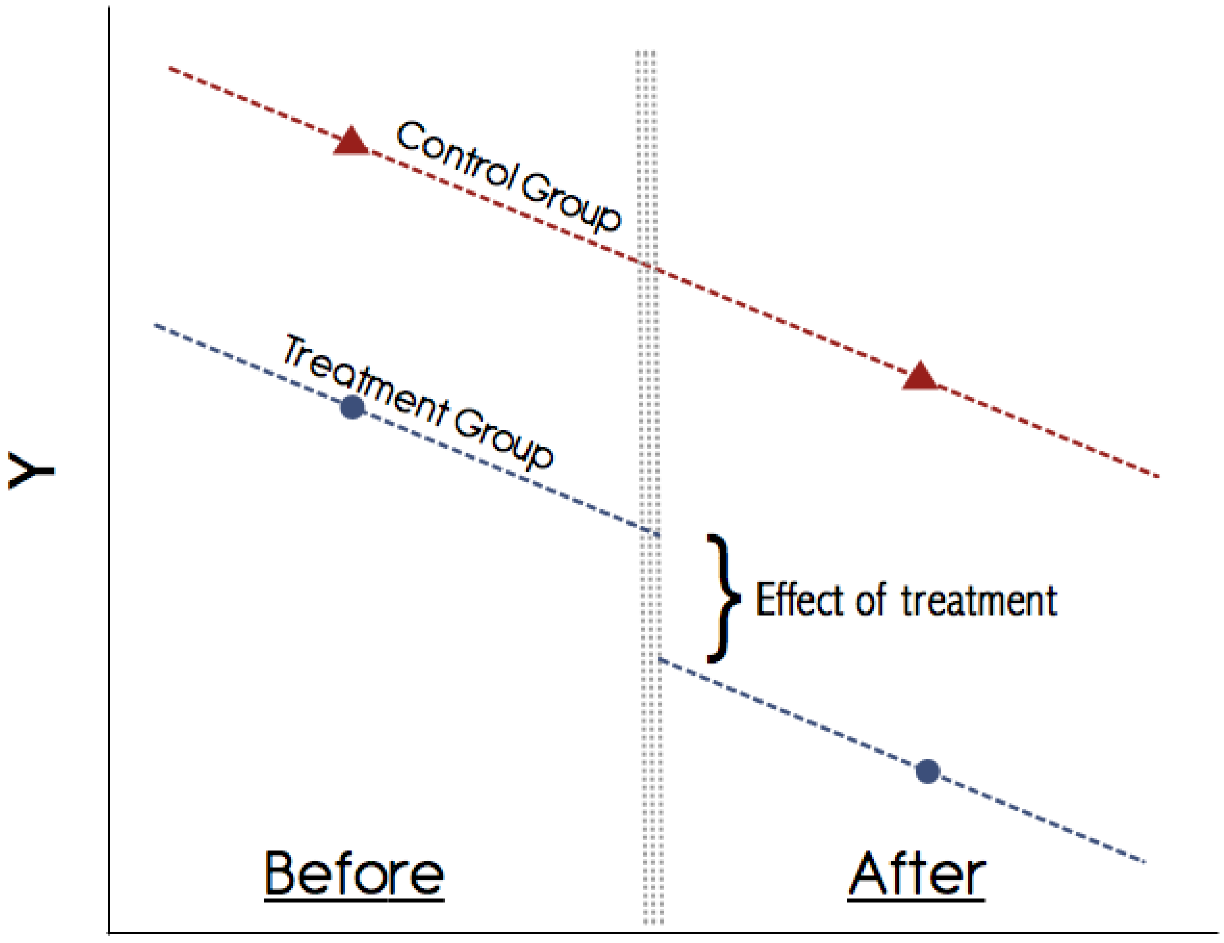


Figure 1: Differences-in-differences

In order to illustrate differences-in-differences, we use a dataset that shows the effect of the *Earned Income Tax Credit (EITC)* that was introduced in 1993 and benefited women with children. The question we are interested in is: how did the EITC affect the employment status of women? Let us first load the dataset.

```r
setwd("C:/Users/Jan/OneDrive/Documents/GitHub/ps630_lab/ps630_f16/w13/")
eitc = read.dta("eitc.dta")
summary(eitc)
```

```
##      state            year          urate           children    
##  Min.   :11.00   Min.   :1991   Min.   : 2.600   Min.   :0.000  
##  1st Qu.:31.00   1st Qu.:1992   1st Qu.: 5.700   1st Qu.:0.000  
##  Median :56.00   Median :1993   Median : 6.800   Median :1.000  
##  Mean   :54.52   Mean   :1993   Mean   : 6.762   Mean   :1.193  
##  3rd Qu.:81.00   3rd Qu.:1995   3rd Qu.: 7.700   3rd Qu.:2.000  
##  Max.   :95.00   Max.   :1996   Max.   :11.400   Max.   :9.000  
##     nonwhite           finc             earn             age       
##  Min.   :0.0000   Min.   :     0   Min.   :     0   Min.   :20.00  
##  1st Qu.:0.0000   1st Qu.:  5123   1st Qu.:     0   1st Qu.:26.00  
##  Median :1.0000   Median :  9637   Median :  3332   Median :34.00  
##  Mean   :0.6007   Mean   : 15255   Mean   : 10432   Mean   :35.21  
##  3rd Qu.:1.0000   3rd Qu.: 18659   3rd Qu.: 14321   3rd Qu.:44.00  
##  Max.   :1.0000   Max.   :575617   Max.   :537881   Max.   :54.00  
##        ed              work           unearn       
##  Min.   : 0.000   Min.   :0.000   Min.   :  0.000  
##  1st Qu.: 7.000   1st Qu.:0.000   1st Qu.:  0.000  
##  Median :10.000   Median :1.000   Median :  2.973  
##  Mean   : 8.806   Mean   :0.513   Mean   :  4.823  
##  3rd Qu.:11.000   3rd Qu.:1.000   3rd Qu.:  6.864  
##  Max.   :11.000   Max.   :1.000   Max.   :134.058  
```

Our dependent variable is "work" - a binary variable indicating employment with 1.

We need a minimum of four groups to conduct a diff-in-diff analysis. Those groups are separated according to two categories - time point 1 and time point 2 to estimate the trend. And treatment group and control group to estimate the effect of the treatment. The EITC went into effect in the year 1994, meaning that the time before 1994 is our time point 1 and 1994 onwards is time point 2.

```r
# We turn our logical vector into a numeric (1/0) The value '1' stands for
# 1994 onwards
eitc$post93 = as.numeric(eitc$year >= 1994)
```

As stated above, the EITC benefits women with children, so let us subset the data and only look at people with children.

```r
# We create a new binary variable The value '1' stands for someone with 1 or
# more child(ren)
eitc$anykids = as.numeric(eitc$children >= 1)
```

Next we look at the means of the four possible combinations.

```r
a = sapply(subset(eitc, post93 == 0 & anykids == 0, select = work), mean)
b = sapply(subset(eitc, post93 == 0 & anykids == 1, select = work), mean)
c = sapply(subset(eitc, post93 == 1 & anykids == 0, select = work), mean)
d = sapply(subset(eitc, post93 == 1 & anykids == 1, select = work), mean)
```

Finally, we compute the differences-in-differences by using the above four means.

```r
(d - c) - (b - a)
```

```
##       work 
## 0.04687313
```

How does this capture the causal effect?

Another way to look at this result is to take into account the following regression.

$$work = \beta_0 + \delta_0 post93 + \beta_1 anykids + \delta_1 (anykids \times post93) + \varepsilon$$

Figure 2: Regression

```r
eitc$p93kids.interaction = eitc$post93 * eitc$anykids
reg1 = lm(work ~ post93 + anykids + p93kids.interaction, data = eitc)
summary(reg1)
```

```
## 
## Call:
## lm(formula = work ~ post93 + anykids + p93kids.interaction, data = eitc)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -0.5755 -0.4908  0.4245  0.5092  0.5540 
## 
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)    
## (Intercept)          0.575460   0.008845  65.060  < 2e-16 ***
## post93              -0.002074   0.012931  -0.160  0.87261    
## anykids             -0.129498   0.011676 -11.091  < 2e-16 ***
## p93kids.interaction  0.046873   0.017158   2.732  0.00631 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.4967 on 13742 degrees of freedom
## Multiple R-squared:  0.0126, Adjusted R-squared:  0.01238 
## F-statistic: 58.45 on 3 and 13742 DF,  p-value: < 2.2e-16
```

How would we interpret this output? What does it tell us about the four groups we created?

For more on diff-in-diff see the following sources:

Angrist & Pischke (2015) *Mastering 'Metrics*, Chapter 5

Angrist & Pischke (2009) *Mostly Harmless Econometrics*, Chapter 5.2

For a more comprehensive discussion of the above example, see the following website:

https://thetarzan.wordpress.com/2011/05/24/surviving-graduate-econometrics-with-r-difference-in-difference-estimation-2-of-

4. Regression discontinuity

Regression discontinuity designs are powerful tools for causal inference where a real randomized experiment is not feasible (like diff-in-diff). They make use of the existence of a cut-off point which is associated with only minor changes in a continuous/interval variable that are expected to lead to significant substantive effects between the affected units.

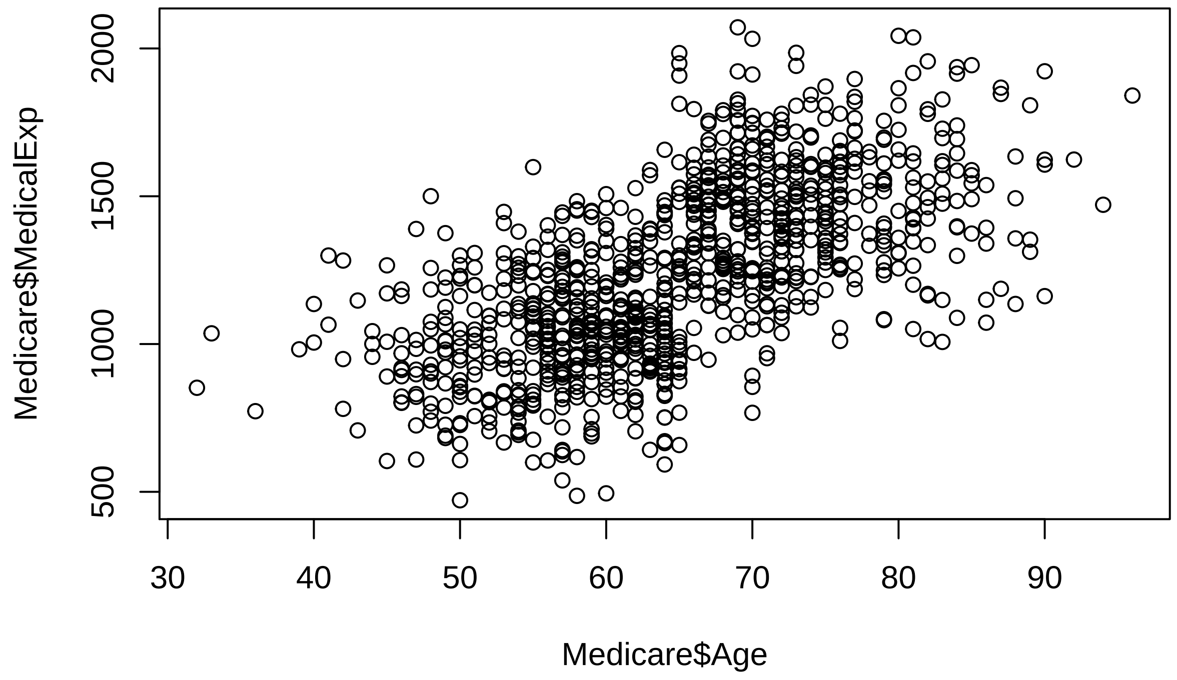
You were asked to read the article **Hidalgo & Nichter (2015) Voter Buying - Shaping the Electorate through Clientelism**. In this article, the authors show how the "random" introduction of voter auditing can affect the extent to which parties can buy votes. This is a good example of how applied political research can make use of as-if-random real-world discontinuities.

In order to illustrate RDD, let us first load an artificial dataset that contains medical expenditures by age for US citizens. We will make use of a discontinuity at the age of 65. In the United States, the medicare program helps citizens at and above the age of 65 to finance their medical expenditures. Although citizens at the age of 64 can be expected to not substantially differ from citizens at the age of 65 in terms of their medical needs, the fact that medicare supports citizens at the age of 65 indicates that there could be a discontinuity in medical expenditures at this cutoff point.

```
load("Medicare.Rdata")
summary(Medicare)
```

```
##       Age          MedicalExp       covariate1       covariate2
##  Min.   :32.00   Min.   : 471.6   Min.   : 69.44   Min.   : 66.38
##  1st Qu.:57.00   1st Qu.:1014.8   1st Qu.: 93.15   1st Qu.: 93.37
##  Median :64.00   Median :1234.0   Median :100.32   Median :100.23
##  Mean   :64.78   Mean   :1241.3   Mean   :100.06   Mean   : 99.88
##  3rd Qu.:72.00   3rd Qu.:1464.2   3rd Qu.:106.77   3rd Qu.:106.44
##  Max.   :96.00   Max.   :2071.4   Max.   :135.19   Max.   :126.70
```

```
plot(Medicare$Age, Medicare$MedicalExp)
```



The plot does not really tell us what is going. There might or might not be a discontinuity at the age of 65. Let us use some RDD tools to figure this out.

The following content is based on code from Matthieu Stigler. Note that the R commands in the original instructions are based on an older version of the package. The modified commands are below.

The original can be found here: https://github.com/MatthieuStigler/RDDtools

We will use a package called "rddtools" to conduct this analysis.

install.packages("rddtools")

```
library(rddtools)
```

```
## Loading required package: AER
```

```
## Loading required package: car
```

```
## Loading required package: survival
```

```
## Loading required package: np
```

```
## Nonparametric Kernel Methods for Mixed Datatypes (version 0.60-2)
## [vignette("np_faq",package="np") provides answers to frequently asked questions]
```

"rddtools" requires us to declare our data an rdd object so that it can apply the correct tools.

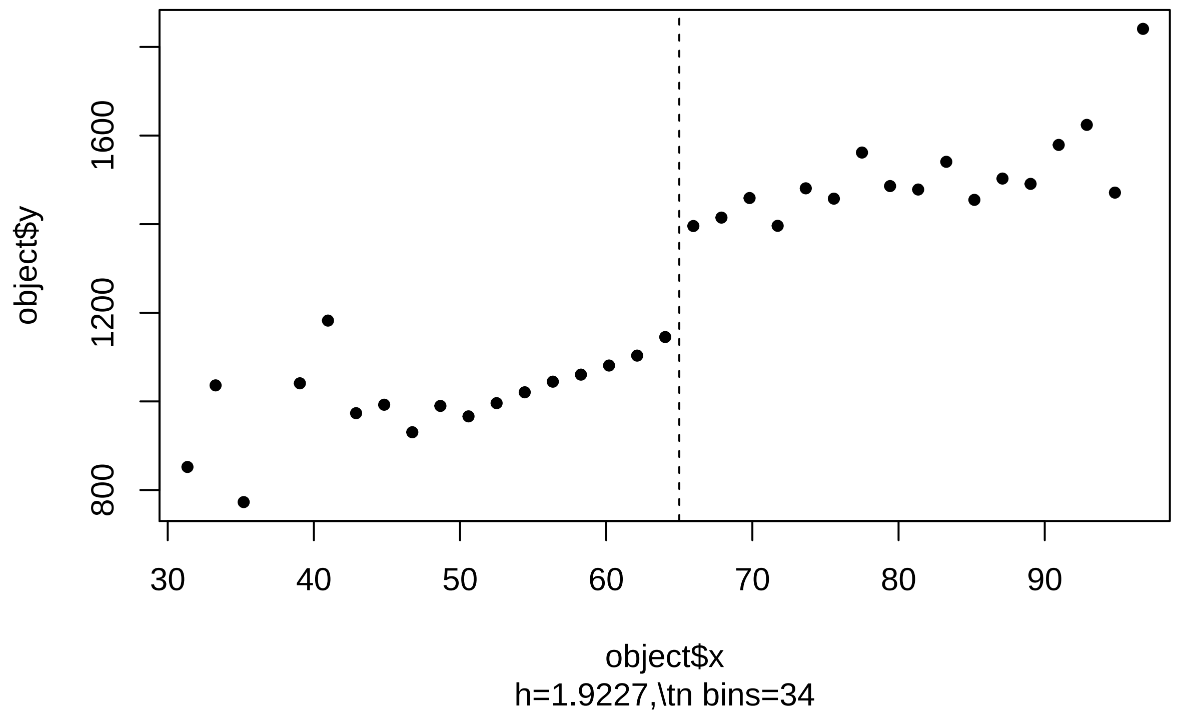
```
medicare_rdd <- rdd_data(y = Medicare$MedicalExp, # Dependent variable
                         x = Medicare$Age, # Key independent variable
                         covar = Medicare, # Dataset containing control variables
                         cutpoint = 65) # Our cutpoint
```

We can now apply "summary" and "plot" to get an overview of the data.

```
summary(medicare_rdd)
```

```
## ### rdd_data object ###
##
## Cutpoint: 65
## Sample size:
##  -Full : 1000
##  -Left : 503
##  -Right: 497
## Covariates: yes
```

```
plot(medicare_rdd)
```



The plot shows the average values in each of the bins, where each bin. Unlike our scatterplot, it gives us a clear impression of whether or not there might be a discontinuity. Indeed, it appears that there is some discontinuity in our data.

Now, let us turn to parametric regression estimation. In this regression approach, we assume that we know the function that describes the data. Linear regression is a parametric regression because we assume a linear relationship of form y = a + bx. The “rddtools” package allows us to include polynomials of different order in our regression to estimate non-linear regressions. Even with a discontinuity, there might be non-linear relationships.

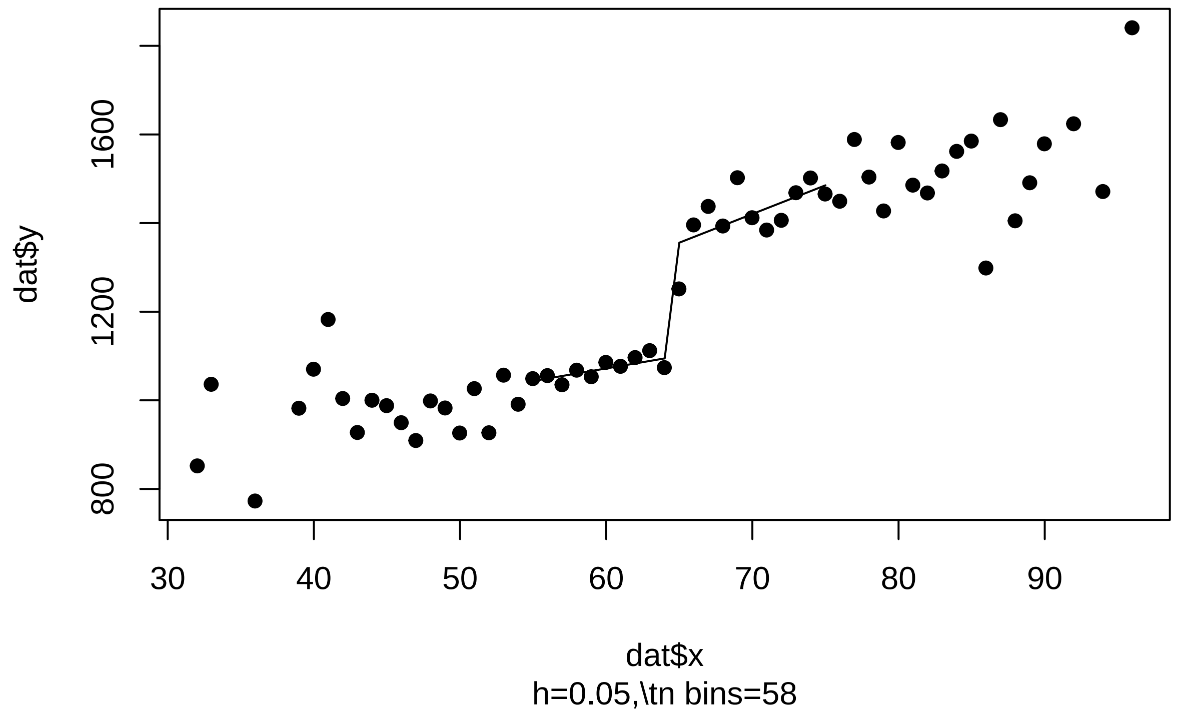
```
reg_para <- rdd_reg_lm(rdd_object = medicare_rdd, order = 1, bw = 10)  # Bandwidth of the regression di
reg_para
```

```
## ### RDD regression: parametric ###
##  Polynomial order:  1
##  Slopes:  separate
##  Bandwidth:  10
##  Number of obs: 690 (left: 344, right: 346)
##
##  Coefficient:
##   Estimate Std. Error t value  Pr(>|t|)
## D  255.496     33.554  7.6145 8.793e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

How would we interpret this result?

Now let us plot the regression.

```r
plot(reg_para)
```

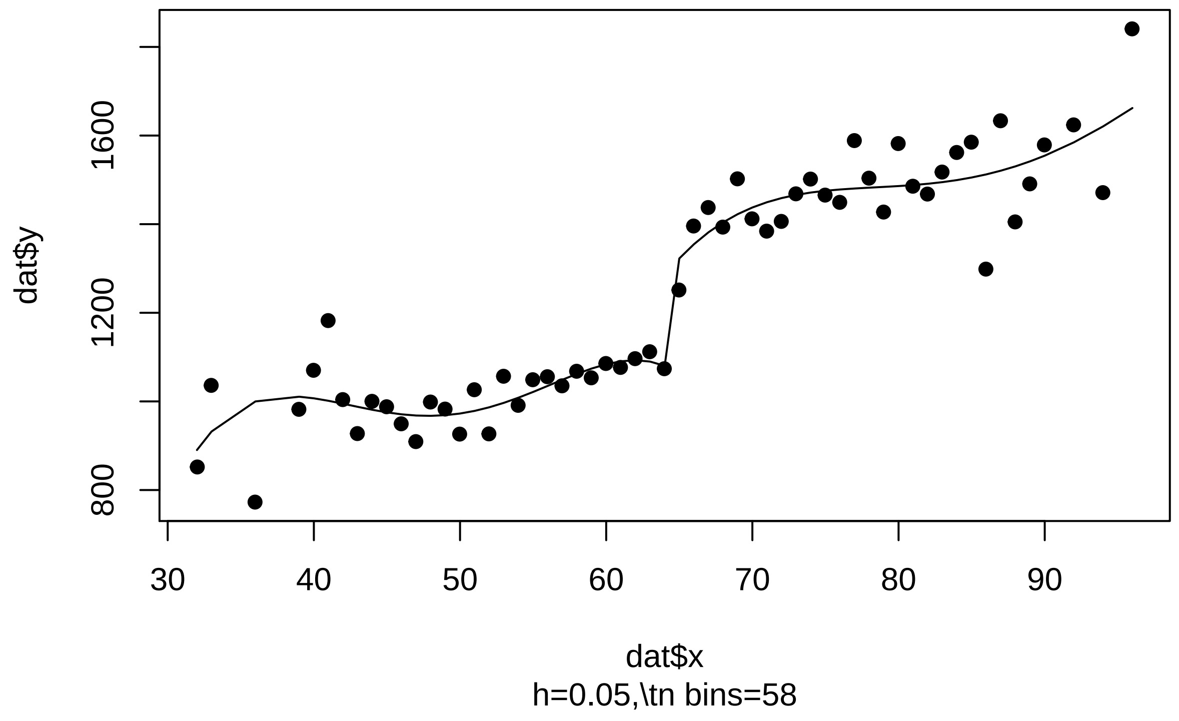


Alternatively, let us use a regression with a higher-order polynomial.

```r
reg_para2 <- rdd_reg_lm(rdd_object = medicare_rdd, order = 4)
reg_para2
```

```
## ### RDD regression: parametric ###
##  Polynomial order:  4
##  Slopes:  separate
##  Number of obs: 1000 (left: 503, right: 497)
##
##  Coefficient:
##   Estimate Std. Error t value  Pr(>|t|)
## D  260.007     52.539  4.9488 8.772e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
plot(reg_para2)
```



Note: if we have control variables, we can include those control variables in the following way:

```
reg_para3 <- rdd_reg_lm(rdd_object = medicare_rdd, covariates = "covariate1 + covariate2",
    order = 4)
reg_para3
```

```
## ### RDD regression: parametric ###
##  Polynomial order:  4
##  Slopes:  separate
##  Number of obs: 1000 (left: 503, right: 497)
##
##  Coefficient:
##   Estimate Std. Error t value  Pr(>|t|)
## D  260.117     52.583  4.9468 8.864e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(reg_para3)
```

```
##
## Call:
## lm(formula = y ~ ., data = dat_step1, weights = weights)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -665.65 -147.91   -9.58  149.31  658.69
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.098e+03  1.100e+02   9.989  < 2e-16 ***
## D            2.601e+02  5.258e+01   4.947 8.86e-07 ***
## x           -2.204e+01  2.117e+01  -1.041   0.2981
## `x^2`       -4.678e+00  3.083e+00  -1.517   0.1295
## `x^3`       -2.461e-01  1.627e-01  -1.512   0.1308
## `x^4`       -3.918e-03  2.738e-03  -1.431   0.1528
## x_right      5.675e+01  2.724e+01   2.083   0.0375 *
## `x^2_right`  1.879e+00  4.144e+00   0.453   0.6503
## `x^3_right`  3.415e-01  2.284e-01   1.495   0.1352
## `x^4_right`  2.948e-03  4.027e-03   0.732   0.4644
## covariate1  -4.335e-01  7.053e-01  -0.615   0.5389
## covariate2   7.570e-02  7.139e-01   0.106   0.9156
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 220.7 on 988 degrees of freedom
## Multiple R-squared:  0.4706, Adjusted R-squared:  0.4647
## F-statistic: 79.84 on 11 and 988 DF,  p-value: < 2.2e-16
```

The following command computes the "optimal bandwidth" of the RDD, following a procedure by Imbens, Guido and Karthik Kalyanaraman (2012).

```
bw_ik <- rdd_bw_ik(medicare_rdd)
bw_ik
```

```
##    h_opt
## 14.77028
```

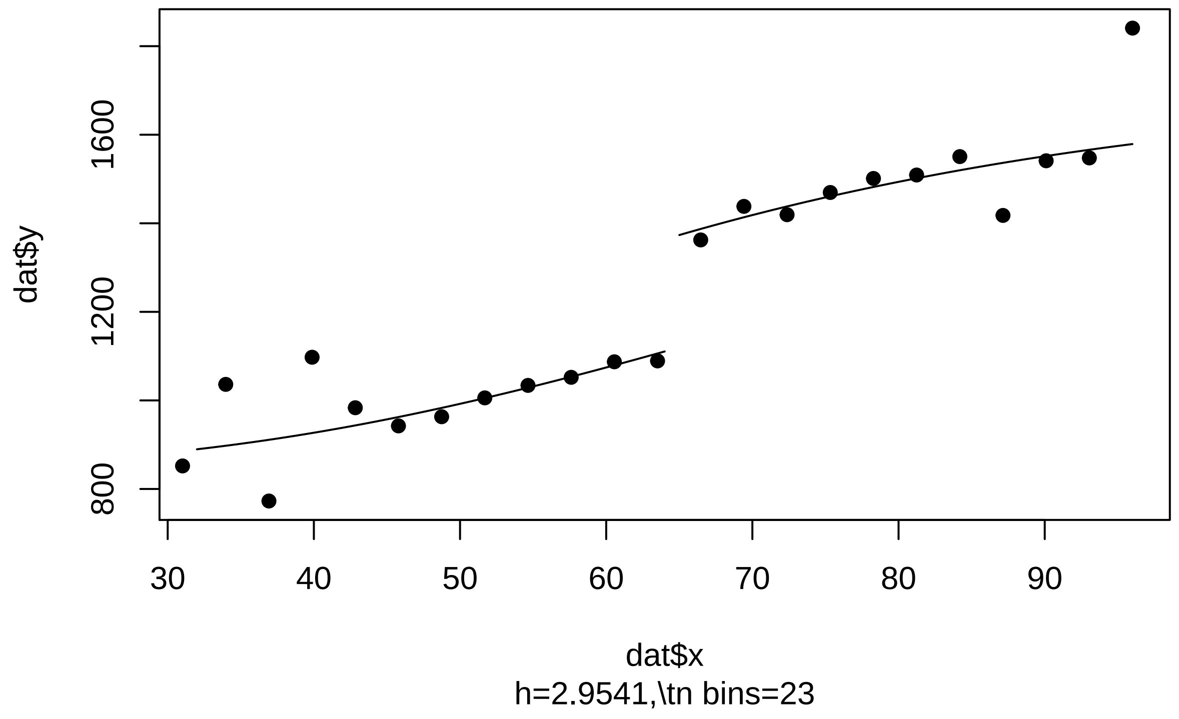
Non-parametric regression refers to regressions where we do not make any assumptions about the function that links our variables.

```
reg_nonpara <- rdd_reg_np(rdd_object = medicare_rdd, bw = bw_ik)
print(reg_nonpara)
```

```
## ### RDD regression: nonparametric local linear###
##  Bandwidth:  14.77028
##  Number of obs: 824 (left: 413, right: 411)
##
##  Coefficient:
##   Estimate Std. Error z value  Pr(>|z|)
## D  247.776     33.827  7.3249 2.391e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

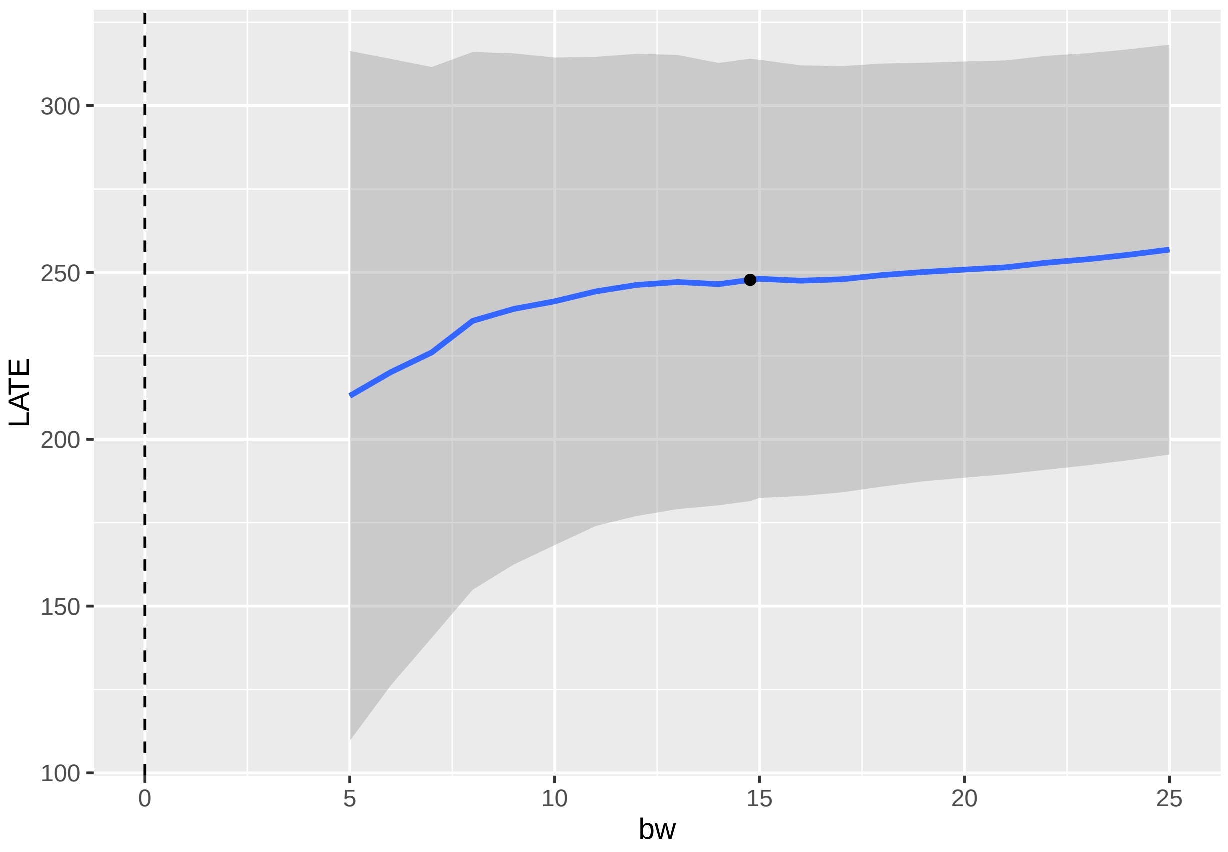
Next, we plot the nonparametric regression.

```r
plot(x = reg_nonpara)
```



Let us apply a regression sensitivity test. This plot shows us how our results change if we use different bandwidths for estimating the differences. In the example below we look at bandwidths 5 to 25.

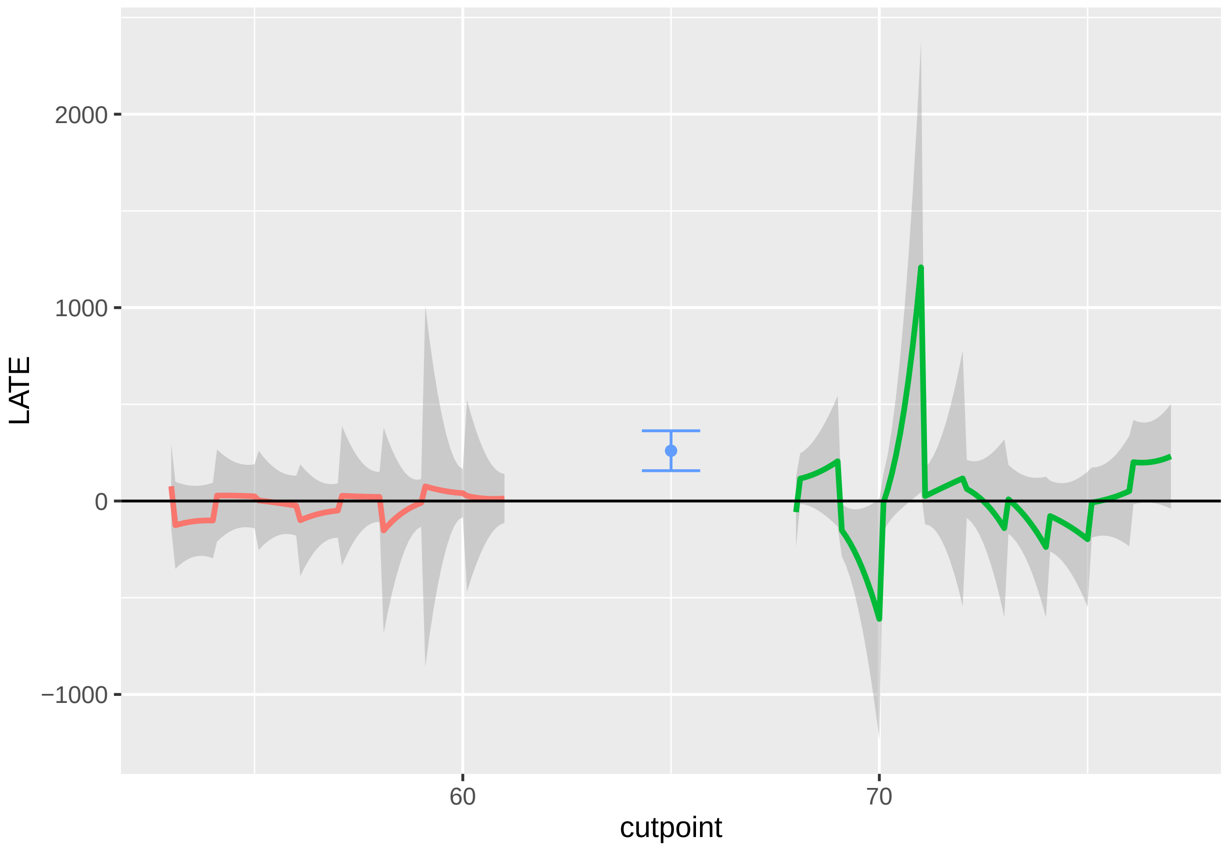
```r
plotSensi(reg_nonpara, from = 5, to = 25, by = 1)
```



How would we interpret these results?

Another way to check the robustness of our results is a placebo test. Here we check for other values in the regression discontinuity whether they would yield any significant results.

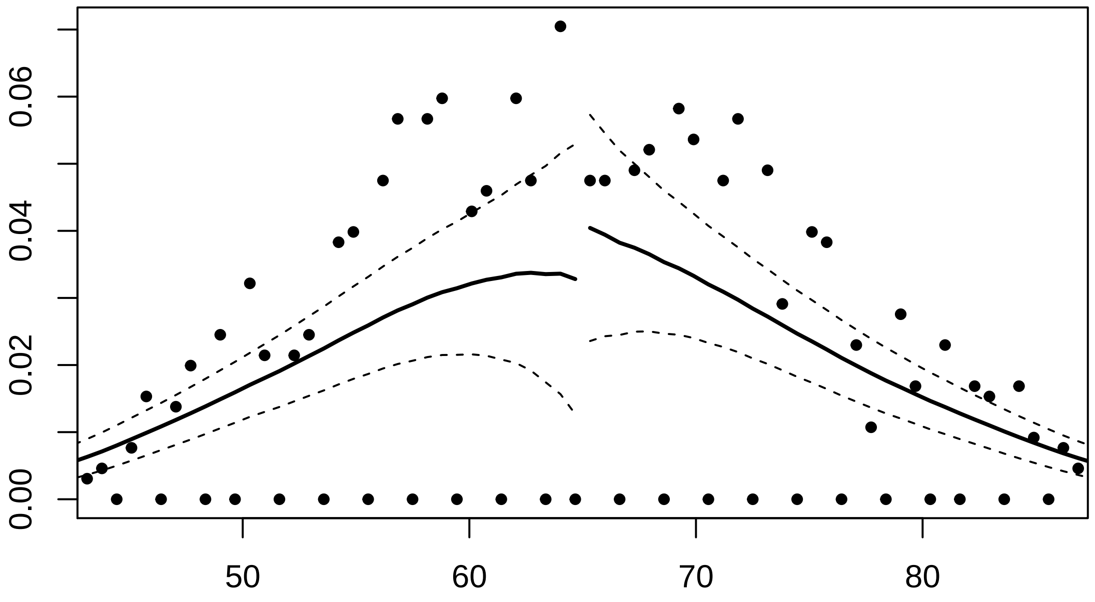
```
plotPlacebo(reg_para2)
```



How would we interpret this plot?

Another problem that might arise in regression discontinuity designs is that there might be some units that should not have been treated but were treated. For example, assume in our case that some 63- or 64-year olds who are in need of medical treatment deliberately claim that they are 65 year old to receive treatment. In this case, we would observe an unnatural discontinuity in the density of values around the threshold. The following command allows us to conduct such a test:

```
dens_test(reg_nonpara)
```



```
##
##  McCrary Test for no discontinuity of density around cutpoint
##
## data:  reg_nonpara
## z-val = 1.6753, p-value = 0.09388
## alternative hypothesis: Density is discontinuous around cutpoint
## sample estimates:
## Discontinuity
##     0.2243895
```

How would we interpret these results? Considering the p-value, should we have more or less confidence in our estimates?

Finally, we can check whether or not our discontinuity comes from covariates. In our example of medicare expenditures, we could assume that several factors can contribute to the increase in medical expenditures. For example, many people stop working at the age of 65. Maybe people who do not work anymore become bored and go to the doctor more frequently to talk to someone. If we have data on whether or not people are still in employment, we could use this covariate test to identify whether or not the employment status after 65 primarily contributes to the spike in medical expenditures.

The following test shows us whether there is a significant difference in the covariates around the threshold with a bandwidth of 5.

```
covarTest_mean(medicare_rdd, bw = 5)
```

```
##            mean of x mean of y Difference statistic  p.value
## Age        62.21264  67.60697  5.394321   -33.40622  4.295048e-114
## MedicalExp 1088.108  1402.745  314.6375   -13.12917  1.29602e-32
## covariate1 100.4661  99.70858  -0.757559  0.7700209  0.441775
## covariate2 99.6914   100.5774  0.8859947  -0.8563658 0.3923748
```

How would we interpret this output?

RDD replication dataset: Hidalgo & Nichter (2015)

If you would like to see another implementation of RDD based on the article that you have read, please check out the data and replication files that have been made available by **Hidalgo & Nichter (2015) Voter Buying - Shaping the Electorate through Clientelism**. The replication files include both the data and the code, making it possible for you to follow their steps one by one. It can be found here:

https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/9OOLQ7

Note that their code is very complex, so you should only look at it when you plan to implement RDD yourself.

5. Spatial lag models

The following content is based on code from Kyle Beardsley.

For the application of a spatial lag model, we will need the package "cshapes".

install.packages("cshapes")

```r
library(cshapes)
```

```
## Loading required package: sp
```

```
## Loading required package: maptools
```

```
## Checking rgeos availability: FALSE
##      Note: when rgeos is not available, polygon geometry     computations in maptools depend on gpcli
##      which has a restricted licence. It is disabled by default;
##      to enable gpclib, type gpclibPermit()
```

```
## Loading required package: plyr
```

We first load the Uppsala Conflict Data Program (UCDP) intrastate conflict data.

```r
setwd("C:/Users/Jan/OneDrive/Documents/GitHub/ps630_lab/ps630_f16/w13/")
ucdp = read.csv("ucdp_1onset2012.csv")
summary(ucdp)
```

```
##       year           gwno      incidencev412    newconflictinyearv412
##  Min.   :1946   Min.   :-99   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:1968   1st Qu.:230   1st Qu.:0.0000   1st Qu.:0.00000
##  Median :1984   Median :450   Median :0.0000   Median :0.00000
##  Mean   :1983   Mean   :453   Mean   :0.1472   Mean   :0.01791
##  3rd Qu.:1998   3rd Qu.:666   3rd Qu.:0.0000   3rd Qu.:0.00000
##  Max.   :2011   Max.   :950   Max.   :1.0000   Max.   :1.00000
##    onset1v412        onset2v412       onset5v412        onset8v412
##  Min.   :0.00000   Min.   :0.0000   Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.0000
##  Median :0.00000   Median :0.0000   Median :0.00000   Median :0.0000
##  Mean   :0.03748   Mean   :0.0322   Mean   :0.02528   Mean   :0.0233
##  3rd Qu.:0.00000   3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.0000
##  Max.   :1.00000   Max.   :1.0000   Max.   :1.00000   Max.   :1.0000
##   onset20v412      maxintyearv412    govonlyv412      terronlyv412
##  Min.   :0.00000   Min.   :0.0000   Min.   :0.0000   Min.   :0.00000
##  1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00000
##  Median :0.00000   Median :0.0000   Median :0.0000   Median :0.00000
##  Mean   :0.01923   Mean   :0.1956   Mean   :0.0788   Mean   :0.05023
##  3rd Qu.:0.00000   3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.00000
##  Max.   :1.00000   Max.   :2.0000   Max.   :1.0000   Max.   :1.00000
##  bothgovterrv412    sumconfv412
##  Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.00000   1st Qu.:0.0000
##  Median :0.00000   Median :0.0000
##  Mean   :0.01813   Mean   :0.2004
##  3rd Qu.:0.00000   3rd Qu.:0.0000
##  Max.   :1.00000   Max.   :8.0000
```

```r
ucdp = ucdp[ucdp$incidencev412 == 1, ]
```

We use this data to form neighborhood connectivity matrices and associated data for 2005

Specifically, we use the cshapes distmatrix (distance matrix) function to form a matrix of distances in 2005

```r
# Distance matrix with distance between capitals in km
dmat = distmatrix(as.Date("2005-1-1"), type = "capdist")
```

Let us inspect the object that we created.

Note: the cshapes package uses countrycodes. These countrycodes follow the coding scheme of the Correlates of War Project (COW) or Gleditsch and Ward (1999). So what we see here are not country names.

Next, we define neighbors as having a distance of less than 900km.

```r
# Adjacency matrix with values 0 and 1
adjmat = ifelse(dmat > 900, 0, 1)
```

Turn the values on the diagonale into zeros. (States do are not neighbors of themselves)

```r
diag(adjmat) = 0
```

“adjmat”" is now the connectivity matrix with 1 implying a connection and 0 no connection.

Some countries have more neighbors than others. We would expect each neighbor to have a smaller influence if the country has many neighbors, so we row-standardize by taking each element and dividing by the row sums.

```r
adjmat_rs = adjmat
for (i in 1:dim(adjmat)[1]) {
    adjmat_rs[i, ] = adjmat_rs[i, ]/colSums(adjmat, 1)[i]
}
```

“adjmat_rs”" is now the row-standardized connectivity matrix (W) for use in the spatial lag model.

Note that values are missing for rows in which there are no neighbors.

Next, we form country-level data with neighbor information in 2005.

For this purpose, we need to get the get the country codes.

```r
# Extract the codes from the row names
codes = row.names(adjmat)

# Turn the codes into a data frame
nbr = data.frame(codes)
```

Now we create a variable for whether the country experienced an intrastate armed conflict in 2005.

```r
# If a country is in the intrastate armed conflict set in the year 2005, we
# assign a 1
nbr$intra = ifelse(codes %in% ucdp$gwno[ucdp$year == 2005] == TRUE, 1, 0)
```

Then, we create a temporal lag variable, which will be used to form the spatial lags and avoid simultaneity bias

```r
# If a country experienced an intrastate armed conflict in the year 2004, we
# assign a 1
nbr$intra_lag = ifelse(codes %in% ucdp$gwno[ucdp$year == 2004] == TRUE, 1, 0)
```

We move on to compute the number of conflicts that all the neighbors had in the previous year, by multiplying the non-standarized connectivity matrix times the lagged conflict variable.

```r
nbr$nbr_intra = adjmat %*% nbr$intra_lag
```

Then we calculate the number of neighbors each state has.

```r
ones = matrix(1, ncol = 1, nrow = dim(nbr)[1])
nbr$borders = adjmat %*% ones
```

We then put in the year.

```r
nbr$year = matrix(2005, ncol = 1, nrow = dim(nbr)[1])
```

Finally, we compute Wy (actually Wy_[t-1]).

Note that this is the same as the number of neighboring conflicts divided by the number neighbors (nbr_intra/borders).

```r
nbr$nbr_intra_rs = adjmat_rs %*% nbr$intra_lag
```

Finally, we run the spatial lag model, controlling for number of borders. Note that because the DV is binary (0/1), this is a linear probability model. This is not ideal, but useful for the example.

```r
spat_lag = lm(intra ~ nbr_intra_rs + borders, data = nbr)
summary(spat_lag)
```

```
##
## Call:
## lm(formula = intra ~ nbr_intra_rs + borders, data = nbr)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.39631 -0.16486 -0.11986 -0.00567  0.90587
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.184190   0.048821   3.773 0.000227 ***
## nbr_intra_rs  0.224983   0.127751   1.761 0.080118 .
## borders      -0.012866   0.005505  -2.337 0.020676 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3321 on 161 degrees of freedom
##   (28 observations deleted due to missingness)
## Multiple R-squared:  0.06757,    Adjusted R-squared:  0.05599
## F-statistic: 5.834 on 2 and 161 DF,  p-value: 0.00358
```

How would we interpret this output?

6. Learning R - what to do next?

Here are some pieces of advice on how to proceed from here.

First, when you have some free time during the winter break, use it to review what you have learned in this class. You will need it in the future.

The following classes are most useful in terms of pushing your R skills further:

1. PolSci 733 - *Maximum Likelihood Methods* (2nd semester)
2. Stats 523 - *Statistical Programming* (3rd semester/5th semester)
3. Stats 601 - *Bayesian and Modern Statistics* (4th semester/6th semester)

Note: R is best learned gradually. Don't take everything at once. Give yourself the opportunity to learn more in the future.

Also, Stats 250 has an R lab, too. This R lab focuses on introducing R for mathematical purposes in the context of statistical theory (and it does not assume a background in R).

The following books will help you to master R:

1. Fox & Weisberg (2010) *An R Companion to Applied Regression* (2nd Edition)
2. Gelman & Hill (2006) *Data Analysis Using Regression and Multilevel/Hierarchical Models*
3. Chang (2013) *R Graphics Cookbook*
4. Kabacoff (2011) *R in Action*
5. Maindonald & Braun (2010) *Data analysis and graphics using R*
6. Matloff (2011) *The Art of R Programming - a Tour of Statistical Software Design*

In the future, you will often find yourself in the situation that the tutorial has not covered a problem you run into. Then you need to be able to find a solution yourself. Taking advanced classes in R will also help you to *develop* a solution yourself, i.e. to write your own script in R to deal with the problem.

R is a continuous learning experience.

Remember

From our first session.

```r
fun = c("R", "is", "fun")
paste(fun, collapse = " ")
```

```
## [1] "R is fun"
```