Lecture 12: Advanced Regressions & Stargazer Continued

We're going to cover:

Regression related:

- Panel (Fixed and Random Effects)
- Logit
- Margins
- one more stargazer

Advanced Regressions

Panel Data

There are a few packages out there that implement panel data models. We'll cover two of them, plm() - an older frequently used package, and a newer package, fixest(). Another package to note is lfe.

Fixed Effects Models

Fixed effects models are equivalent to including dummy variables for the entity that you are following over time. You can include these dummy variables as factors (either for time or entity). However, that is computationally inefficient if you have some 1,500 people, for example. So, instead, we use the within transformation.

In plm() is an option for "within" - that is the fixed effects model. This is because you are taking differences within an entity.

Let's use world bank data to apply the fixed effects model.

```
library("WDI")
library("tidyverse")
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.4.2     v purrr   1.0.1
## v tibble  3.2.1     v dplyr   1.1.1
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.4     v forcats 1.0.0
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
wdi <- WDI(country = "all", start=2000, end=2015, extra="TRUE",
           indicator=c("NY.GDP.MKTP.KD.ZG","SL.TLF.CACT.FE.ZS","SP.POP.TOTL", "SE.PRM.CUAT.ZS" ))
```

Let's quickly clean this up a bit by renaming the columns. I'm going to leave in unnecessary columns since the data is not very large, but if I was working with a large dataframe, I'd drop it.

```
wdi <- rename(wdi,  gdp= NY.GDP.MKTP.KD.ZG, lfpart_f= SL.TLF.CACT.FE.ZS, pop= SP.POP.TOTL, edu = SE.PRM

names(wdi)
```

```
##  [1] "iso2c"     "country"   "year"      "gdp"       "lfpart_f"  "pop"
##  [7] "edu"       "iso3c"     "region"    "capital"   "longitude" "latitude"
## [13] "income"    "lending"
```

Let's great a baseline model regression with OLS:

```
ols <- lm(gdp~edu + log(pop), data=wdi)
```

Now, let's run our model.

We need to tell plm what entity is being followed and what is the variable for time using the option index, and we need to tell plm what kind of model we are running.

Typically, the function is a class of regression methods and there will be an option for a specific estimation model. plm can be considered a package for panel data models and the estimation model is "within"

A one-way fixed effects model is like this:

```
library("plm")
```

```
##
## Attaching package: 'plm'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, lag, lead
```

```
fixed <- plm(gdp~  edu + log(pop) , data=wdi, index = c("iso2c"), model="within")

summary(fixed)
```

```
## Oneway (individual) effect Within Model
##
## Call:
## plm(formula = gdp ~ edu + log(pop), data = wdi, model = "within",
##     index = c("iso2c"))
##
## Unbalanced Panel: n = 146, T = 1-16, N = 642
##
## Residuals:
##       Min.    1st Qu.     Median    3rd Qu.       Max.
## -17.807502  -1.030927   0.062084   1.449688  17.529051
##
## Coefficients:
##           Estimate Std. Error t-value  Pr(>|t|)
## edu      -0.045747   0.048096 -0.9512 0.3419857
## log(pop) -8.933033   2.602289 -3.4328 0.0006477 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    6719
## Residual Sum of Squares: 6432.1
## R-Squared:      0.042701
## Adj. R-Squared: -0.24216
```

```
## F-statistic: 11.0176 on 2 and 494 DF, p-value: 2.0833e-05
```

```
#equivalent in stata:
#xtset iso2c year
#xtreg gdp edu lnpop, fe
```

But, we can include fixed effects for time, too. We just need to include it in the index.

```
fixed <- plm(gdp~ edu + log(pop) , data=wdi, index = c("iso2c", "year"), model="within")
summary(fixed)
```

```
## Oneway (individual) effect Within Model
##
## Call:
## plm(formula = gdp ~ edu + log(pop), data = wdi, model = "within",
##     index = c("iso2c", "year"))
##
## Unbalanced Panel: n = 146, T = 1-16, N = 642
##
## Residuals:
##       Min.    1st Qu.     Median    3rd Qu.       Max.
## -17.807502  -1.030927   0.062084   1.449688  17.529051
##
## Coefficients:
##           Estimate Std. Error t-value  Pr(>|t|)
## edu      -0.045747   0.048096 -0.9512 0.3419857
## log(pop) -8.933033   2.602289 -3.4328 0.0006477 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    6719
## Residual Sum of Squares: 6432.1
## R-Squared:      0.042701
## Adj. R-Squared: -0.24216
## F-statistic: 11.0176 on 2 and 494 DF, p-value: 2.0833e-05
```

We can see the number of observations (N), the number of groups (n), how many units you have over time (T).

If you want to see the fixedeffects estimtates, you can do this with `fixef()`

```
fixef(fixed)
```

```
##      AD      AE      AL      AM      AO      AR      AU      AW      AZ      BA
## 109.874 144.666 142.315 143.521 159.731 162.002 158.803 108.943 154.586 140.015
##      BB      BD      BE      BF      BG      BH      BI      BN      BO      BR
## 119.938 176.558 150.223 152.882 150.261 129.687 149.156 123.184 150.383 176.638
##      BS      BT      BY      BZ      CA      CD      CH      CI      CL      CM
## 121.265 126.547 148.236 118.247 160.410 172.877 147.870 162.427 157.357 154.681
##      CO      CR      CU      CV      CY      CZ      DE      DK      DM      DO
## 165.302 145.053 151.299 121.491 129.910 151.077 168.698 143.428 103.448 152.663
##      DZ      EC      EE      EG      ES      ET      FJ      FR      GB      GE
## 159.910 155.158 141.030 171.733 162.759 175.747 125.317 166.228 165.965 145.283
##      GH      GM      GN      GR      GT      GY      HK      HN      HR      HU
## 163.230 136.495 149.547 149.697 152.494 125.187 148.522 149.456 142.229 152.122
##      ID      IL      IN      IQ      IT      JO      JP      KE      KG      KH
## 181.453 150.102 194.763 166.557 164.119 149.160 175.413 164.731 145.820 154.897
```

```
##      KR      KW      KY      KZ      LB      LI      LK      LS      LT      LU
## 167.194 140.487 104.108 157.685 150.344 100.591 152.059 136.978 141.163 129.303
##      LV      MD      ME      MH      MK      ML      MN      MO      MR      MT
## 135.925 140.757 126.351 101.273 134.785 154.149 139.960 133.680 140.750 126.720
##      MU      MV      MX      MY      MZ      NA      NE      NG      NL      NO
## 133.982 140.970 171.304 161.654 160.059 132.408 160.395 176.613 154.051 144.033
##      NP      OM      PA      PE      PF      PH      PK      PL      PR      PS
## 158.310 139.254 141.983 162.874 116.477 173.505 176.237 163.345 140.873 145.794
##      PT      PW      PY      QA      RO      RS      RU      RW      SA      SC
## 148.747  88.553 146.039 143.731 157.500 147.560 176.968 153.780 160.789 106.779
##      SE      SG      SI      SK      SL      SN      SR      ST      SV      SY
## 150.239 147.379 135.522 145.678 146.301 148.945 127.470 115.744 144.169 157.942
##      TD      TG      TH      TJ      TO      TR      TT      TZ      UA      UG
## 163.861 148.261 169.001 152.587 114.712 171.461 130.221 165.131 171.364 162.298
##      US      UY      VE      WS      ZA      ZW
## 181.021 142.340 161.932 118.707 164.978 152.496
```

fixest

You can run the same above regression with the package fixest using the function feols(). I'm mentioning this particular package because it is *insanely* fast. It can handle a huge amount of fixed effects where stata might break (unless you use reghdfe). It can support non-linear models, high-dimensional fixed effects, multiway clustering and a bunch of options that come in handy when you work with detailed data or big data. So, I want you to know it exists because of it's flexibility in all things fixed effects.

Let's try it. Rather than setting the fixed effects with an index, you'll include the variables that are fixed after |

Let's see what I mean here:

```
library("fixest")

fixed_est = feols(gdp~  edu + log(pop) | country + year, data = wdi)  ## Fixed effect(s) go after the "
```

```
## NOTE: 3,614 observations removed because of NA values (LHS: 234, RHS: 3,609).
```

```
fixed_est
```

```
## OLS estimation, Dep. Var.: gdp
## Observations: 642
## Fixed-effects: country: 146,  year: 16
## Standard-errors: Clustered (country)
##          Estimate Std. Error   t value Pr(>|t|)
## edu       0.00256   0.057445  0.044569  0.96451
## log(pop) -3.02784   3.540885 -0.855108  0.39390
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## RMSE: 2.65076     Adj. R2: 0.473683
##                 Within R2: 0.003543
```

Standard errors are already clustered by country automatically, but if you want standard errors, you can do so using summary and the option se

```
summary(fixed_est, se = 'standard')
```

```
## OLS estimation, Dep. Var.: gdp
## Observations: 642
```

```
## Fixed-effects: country: 146,  year: 16
## Standard-errors: IID
##          Estimate Std. Error   t value Pr(>|t|)
## edu       0.00256   0.045630  0.056109  0.95528
## log(pop) -3.02784   2.426161 -1.247997  0.21264
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## RMSE: 2.65076     Adj. R2: 0.473683
##                 Within R2: 0.003543
```

There might be time where you want to cluster your standard errors with specific items. You can also specify what you would like to cluster with the option cluster

```
summary(fixed_est, cluster = c('iso2c'))
```

```
## OLS estimation, Dep. Var.: gdp
## Observations: 642
## Fixed-effects: country: 146,  year: 16
## Standard-errors: Clustered (iso2c)
##          Estimate Std. Error   t value Pr(>|t|)
## edu       0.00256   0.057445  0.044569  0.96451
## log(pop) -3.02784   3.540885 -0.855108  0.39390
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## RMSE: 2.65076     Adj. R2: 0.473683
##                 Within R2: 0.003543
```

Random Effects Models

For plm, changing to a random effect model is pretty simple. You just need to change the option "model" to "random"

```
re <- plm(gdp~edu + log(pop), data=wdi, index = c("iso2c"), model="random")
summary(re)
```

```
## Oneway (individual) effect Random Effect Model
##    (Swamy-Arora's transformation)
##
## Call:
## plm(formula = gdp ~ edu + log(pop), data = wdi, model = "random",
##     index = c("iso2c"))
##
## Unbalanced Panel: n = 146, T = 1-16, N = 642
##
## Effects:
##                  var std.dev share
## idiosyncratic 13.020   3.608 0.804
## individual     3.183   1.784 0.196
## theta:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1036  0.2890  0.4410  0.3817  0.4794  0.5488
##
## Residuals:
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -17.9249  -1.7722  -0.0257  -0.0139   1.6395  23.4953
```

```
##
## Coefficients:
##               Estimate Std. Error z-value  Pr(>|z|)
## (Intercept) 11.5927604  2.2389034  5.1779 2.244e-07 ***
## edu         -0.0595799  0.0097478 -6.1122 9.829e-10 ***
## log(pop)    -0.1817504  0.1220187 -1.4895    0.1363
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    9458.9
## Residual Sum of Squares: 8531.2
## R-Squared:      0.098148
## Adj. R-Squared: 0.095326
## Chisq: 37.4558 on 2 DF, p-value: 7.3551e-09
```

There is something else called a mixed model where you might have both fixed effects and random effect parameters. This is particularly useful for nested datasets and when you might want to include fixed effects at one level and random effects for another variable. I've rarely seen it in economics papers, but it is often included in psychology models. I won't discuss this indepthly, but here is a website that explains how to apply it in R - although there are many out there and often people use the package lmer().

Hausman Taylor test You may know that when deciding between random or fixed effects models, you have to ensure that the random variables do not correlate with your independent variables. Frequently, this assumption is broken and is why most people use fixed effects models. However, we often want to test between our fixed or random effect model. We do this with the `phtest()` function in plm.

```
fe <- plm(gdp ~ edu +log(pop) , data=wdi, index = c("iso2c"), model="within")

summary(fe)
```

```
## Oneway (individual) effect Within Model
##
## Call:
## plm(formula = gdp ~ edu + log(pop), data = wdi, model = "within",
##     index = c("iso2c"))
##
## Unbalanced Panel: n = 146, T = 1-16, N = 642
##
## Residuals:
##       Min.    1st Qu.     Median    3rd Qu.       Max.
## -17.807502  -1.030927   0.062084   1.449688  17.529051
##
## Coefficients:
##           Estimate Std. Error t-value  Pr(>|t|)
## edu      -0.045747   0.048096 -0.9512 0.3419857
## log(pop) -8.933033   2.602289 -3.4328 0.0006477 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Total Sum of Squares:    6719
## Residual Sum of Squares: 6432.1
## R-Squared:      0.042701
## Adj. R-Squared: -0.24216
## F-statistic: 11.0176 on 2 and 494 DF, p-value: 2.0833e-05
```

```
phtest(fe, re)
```

```
##
##  Hausman Test
##
## data:  gdp ~ edu + log(pop)
## chisq = 14.484, df = 2, p-value = 0.0007158
## alternative hypothesis: one model is inconsistent
```

There are a variety of tests for serial correlation, heteroskedasticity, random effects (if you should include random effects compared to an ols model, stationarity/unit roots, etc). These are outlined in the vignettes of plm and you can explore them further on your own.

Logit/Probit

Logit/probit models can be run using the generalized linear model packages `glm()` You identify a logit or probit model using the option "family". You need to specify the link function to distinguish between a logit or probit model.

A good introduction to logit models (and many other applied econometric models) can be found at UCLA's website here and for probit models here

Let's start with a logit model:

```
setwd("/Users/mkaltenberg/Documents/GitHub/Data_Analysis_Python_R/Advanced Regressions Stargazer/")
smoking <- read.csv("smoking.csv")
names(smoking)
```

```
##  [1] "smoker"   "smkban"   "age"      "hsdrop"   "hsgrad"   "colsome"
##  [7] "colgrad"  "black"    "hispanic" "female"
```

```
logit <- glm(smoker ~ age + female + hsdrop +hsgrad +colsome +colgrad, data = smoking, family = "binomia
summary(logit)
```

```
##
## Call:
## glm(formula = smoker ~ age + female + hsdrop + hsgrad + colsome +
##     colgrad, family = "binomial", data = smoking)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0622  -0.8203  -0.5884  -0.4056   2.2899
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.940401   0.135308 -14.341  < 2e-16 ***
## age         -0.005998   0.001964  -3.054 0.002256 **
## female      -0.210057   0.048467  -4.334 1.46e-05 ***
## hsdrop       1.771164   0.126802  13.968  < 2e-16 ***
## hsgrad       1.521847   0.113672  13.388  < 2e-16 ***
## colsome      1.161500   0.116063  10.007  < 2e-16 ***
## colgrad      0.424145   0.125638   3.376 0.000736 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 11074  on 9999  degrees of freedom
## Residual deviance: 10583  on 9993  degrees of freedom
## AIC: 10597
##
## Number of Fisher Scoring iterations: 4
```

The probit model just needs a tweak in the link function within the option "family", and that is set like this:

```
probit <- glm(smoker ~ age + female + hsdrop +hsgrad +colsome +colgrad, data = smoking, family = binomia
summary(probit)
```

```
##
## Call:
## glm(formula = smoker ~ age + female + hsdrop + hsgrad + colsome +
##     colgrad, family = binomial(link = "probit"), data = smoking)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.0563  -0.8214  -0.5919  -0.4005   2.3070
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.125266   0.073137 -15.386  < 2e-16 ***
## age         -0.003458   0.001157  -2.990 0.002791 **
## female      -0.123336   0.028445  -4.336 1.45e-05 ***
## hsdrop       1.005005   0.068663  14.637  < 2e-16 ***
## hsgrad       0.851358   0.059047  14.418  < 2e-16 ***
## colsome      0.637198   0.060379  10.553  < 2e-16 ***
## colgrad      0.220119   0.064848   3.394 0.000688 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 11074  on 9999  degrees of freedom
## Residual deviance: 10584  on 9993  degrees of freedom
## AIC: 10598
##
## Number of Fisher Scoring iterations: 4
```

Marginal effects can't be read directly from the output. You need to apply the link function to interpret the results (though the sign and standard error can be interpreted as usual).

This is why the margins function is so important! Margins works with other regressions, as well.

Margins

The margins package is quite similar to the margins package in STATA. You can apply interpretation to a wide variety of regressions - logit/probit, but also non-linear terms in OLS models. You can check out more examples and features in its vignette here.

The package makes it easy to calculate values and plot them.

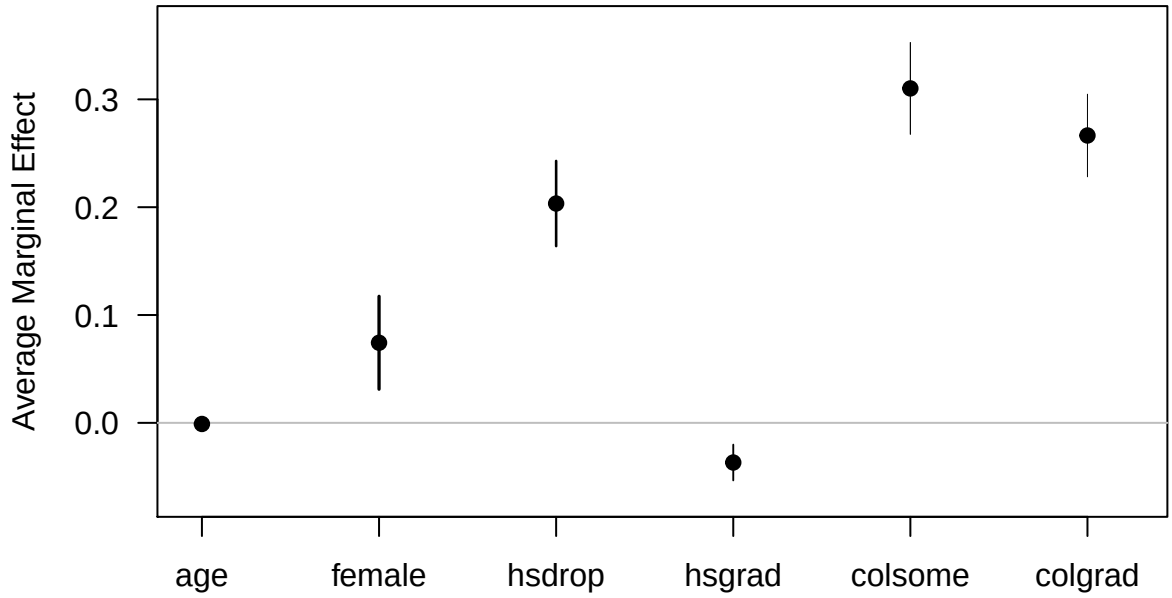
When we use `margins(model_name)` it will give us the *average* marginal effect for each variable

```
library(margins)

logit_m <-margins(logit)
```

And we can plot it quit easily:

```
plot(logit_m)
```



To get the partial effect at the mean (or at any particular value), we us the at option:

Here we specify a range of numbers

```
margins(logit, at = list(age = c(18,40,60)), variables = "age")
```

```
## Average marginal effects at specified values

## glm(formula = smoker ~ age + female + hsdrop + hsgrad + colsome +     colgrad, family = "binomial", d

##  at(age)        age
##       18 -0.0011099
##       40 -0.0010467
##       60 -0.0009871
```

Or at the mean of age

```
mean_age = mean(smoking$age, na.rm=TRUE)

margins(logit, at = list(age = c(mean(smoking$age, na.rm=TRUE))), variables = "age")
```

```
## Average marginal effects at specified values

## glm(formula = smoker ~ age + female + hsdrop + hsgrad + colsome +     colgrad, family = "binomial", d

##  at(age)      age
##    38.69 -0.00105
```

Marignal effects for non-linear terms in OLS

The margins package can also be used to get the marginal effects for non-linear terms in OLS models - often these are interactions or polynomial functions. Let's take an example of a model with the squared term population.

Notice that to square a term I use "^2" and the "I()" to indicate that it's a second order term.

```
int <- lm(gdp ~ lfpart_f + pop + I(pop^2), data = wdi)
summary(int)
```

```
##
## Call:
## lm(formula = gdp ~ lfpart_f + pop + I(pop^2), data = wdi)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -53.978  -2.051   0.007   2.114  83.188
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.191e+00  2.933e-01  10.877  < 2e-16 ***
## lfpart_f     1.301e-02  5.599e-03   2.323 0.020238 *
## pop          1.241e-09  2.413e-10   5.142 2.86e-07 ***
## I(pop^2)    -1.726e-19  4.610e-20  -3.744 0.000184 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.955 on 3659 degrees of freedom
##   (593 observations deleted due to missingness)
## Multiple R-squared:  0.01068,    Adjusted R-squared:  0.00987
## F-statistic: 13.17 on 3 and 3659 DF,  p-value: 1.502e-08
```

Let's apply marginal effect to population. We can take the marginal effect of any particular value of population that we are interested in.

```
# in python wdi['pop']

m_pop <- mean(wdi$pop, na.rm=TRUE)  # getting the mean of population

margins(int, at = list(pop =m_pop))
```

```
## Average marginal effects at specified values

## lm(formula = gdp ~ lfpart_f + pop + I(pop^2), data = wdi)

##    at(pop) lfpart_f       pop
##  272437737  0.01301 1.147e-09
```

We can combine those two lines of code into one (and not save the value of the mean of population that will take up RAM)

```
margins(int, at = list(pop = mean(wdi$pop, na.rm=TRUE)))
```

```
## Average marginal effects at specified values

## lm(formula = gdp ~ lfpart_f + pop + I(pop^2), data = wdi)

##    at(pop) lfpart_f       pop
##  272437737  0.01301 1.147e-09
```

We can input a list of values rather than just one value

```
margins(int, at = list(pop = c(1000000,5000000, 9000000)))
```

```
## Average marginal effects at specified values
```

```
## lm(formula = gdp ~ lfpart_f + pop + I(pop^2), data = wdi)
```

```
##  at(pop) lfpart_f       pop
##    1e+06  0.01301 1.240e-09
##    5e+06  0.01301 1.239e-09
##    9e+06  0.01301 1.238e-09
```

It's far better to use numbers relevant to the value/data that exists. We can use a bunch of summary statistics instead of just one to see the marginal effects of population. Turkey's 5 numbers include min, lower hinge, median, upper hinge and maximum - the function is called `fivenum()`

Looking at the marginal effects, you can see that the labor force participation of women is constant (as expected), but population marginal effects depend on the value of population - there seems to be a non-linear relationship.

```
fivenum(wdi$pop, na.rm= TRUE)  # This is how to get the statistics
```

```
## [1]       9609    1377658    9075626   58056732 7404910892
```

```
margins(int, at = list(pop =fivenum(wdi$pop, na.rm= TRUE)))  #integrated within margins command
```

```
## Warning in check_values(data, at): A 'at' value for 'pop' is outside observed
## data range (102603,7404910892)!
```

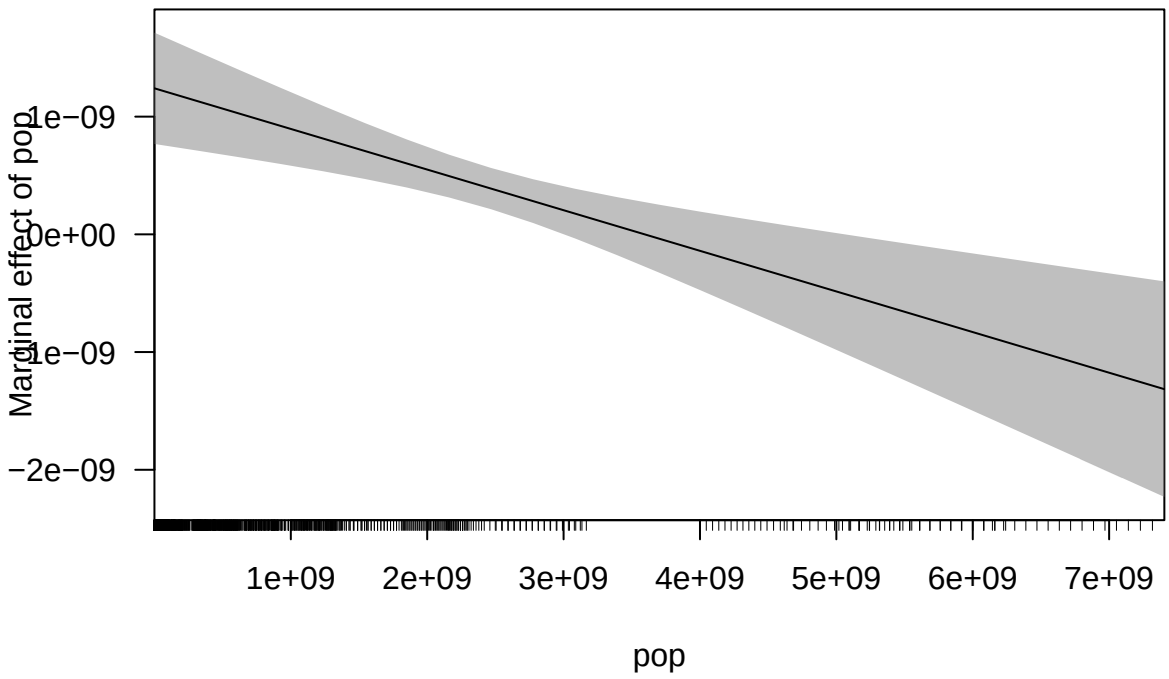
```
## Average marginal effects at specified values
```

```
## lm(formula = gdp ~ lfpart_f + pop + I(pop^2), data = wdi)
```

```
##    at(pop) lfpart_f        pop
##  9.609e+03  0.01301  1.241e-09
##  1.378e+06  0.01301  1.240e-09
##  9.076e+06  0.01301  1.238e-09
##  5.806e+07  0.01301  1.221e-09
##  7.405e+09  0.01301 -1.315e-09
```

And we can also graph the marginal effect of population with `cplot()`

```
cplot(int, "pop", what = "effect", main = "Average Marginal Effect of Population")
```



Stargazer (Again)

Let's show some models side by side

```
library(stargazer)
```

```
##
## Please cite as:
##  Hlavac, Marek (2018). stargazer: Well-Formatted Regression and Summary Statistics Tables.
##  R package version 5.2.2. https://CRAN.R-project.org/package=stargazer
```

```
knitr::opts_chunk$set(echo = TRUE, warning = FALSE)

stargazer(ols, re, fe, type = "html", column.labels = c("OLS", "RE", "FE"), model.names = FALSE,
          dep.var.caption = "",
          title            = "Panel Data Results",
          covariate.labels = c("Edu", "lnPop", "Constant"),
          dep.var.labels   = "GDP per capita",
          out = "panel_results.html")
```

```
##
## <table style="text-align:center"><caption><strong>Panel Data Results</strong></caption>
## <tr><td colspan="4" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left"
## <tr><td style="text-align:left"></td><td>OLS</td><td>RE</td><td>FE</td></tr>
## <tr><td style="text-align:left"></td><td>(1)</td><td>(2)</td><td>(3)</td></tr>
## <tr><td colspan="4" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left"
## <tr><td style="text-align:left"></td><td>(0.008)</td><td>(0.010)</td><td>(0.048)</td></tr>
## <tr><td style="text-align:left"></td><td></td><td></td><td></td><td></td></tr>
## <tr><td style="text-align:left">lnPop</td><td>-0.274<sup>***</sup></td><td>-0.182</td><td>-8.933<sup
```

```
## <tr><td style="text-align:left"></td><td>(0.091)</td><td>(0.122)</td><td>(2.602)</td></tr>
## <tr><td style="text-align:left"></td><td></td><td></td><td></td></tr>
## <tr><td style="text-align:left">Constant</td><td>13.079<sup>***</sup></td><td>11.593<sup>***</sup></
## <tr><td style="text-align:left"></td><td>(1.710)</td><td>(2.239)</td><td></td></tr>
## <tr><td style="text-align:left"></td><td></td><td></td><td></td></tr>
## <tr><td colspan="4" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left"
## <tr><td style="text-align:left">R<sup>2</sup></td><td>0.088</td><td>0.098</td><td>0.043</td></tr>
## <tr><td style="text-align:left">Adjusted R<sup>2</sup></td><td>0.085</td><td>0.095</td><td>-0.242</t
## <tr><td style="text-align:left">Residual Std. Error</td><td>4.046 (df = 639)</td><td></td><td></td><
## <tr><td style="text-align:left">F Statistic</td><td>30.904<sup>***</sup> (df = 2; 639)</td><td>37.45
## <tr><td colspan="4" style="border-bottom: 1px solid black"></td></tr><tr><td style="text-align:left"
## </table>
```