Big Data and Economics

Fixed Effects and Difference-in-differences

Kyle Coombs

Bates College | ECON/DCS 368

Contents

Today's lecture explores

Software requirements

R packages

It's important to note that "base" R already provides all of the tools to implement a fixed effects regression, **but** you'll quickly hit walls due to memory caps. Instead, I want to introduce **fixest**, short for Fixed-Effects Estimation, which provides lightning fast fixed effects estimation and make your life much easier.

- New: **fixest**, **wooldridge**
- Already used: **tidyverse**, **hrbrthemes**, **listviewer**, **estimatr**, **ivreg**, **sandwich**, **lmtest**, **mfx**, **margins**, **broom**, **modelsummary**, **vtable**, **rstanarm**

A convenient way to install (if necessary) and load everything is by running the below code chunk.

```
## Load and install the packages that we'll be using today
if (!require("pacman")) install.packages("pacman")
pacman::p_load(mfx, tidyverse, hrbrthemes, estimatr, ivreg, fixest, sandwich, wooldridge)

## My preferred ggplot2 plotting theme (optional)
theme_set(theme_minimal())
```

Note on fixest and feols I'll be using fixest and feols throughout these notes. The fixest package is a new package that is very fast and has a lot of functionality. It has several bits of funtionality like `feols()` and `etable()`, which are powerful functions for making regressions and putting the output into tables that work well together. `feols()` works very much like `lm()` in base R, but with a few added bonuses.

Review of last lecture

Last lecture we covered how fixed effects are extremely useful for removing variation between units. That means any of the average differences between groups of the fixed effect are removed. We can then look at underlying variation within these groups to see if there is a relationship between our variables of interest.

This is extremely useful for dealing with omitted variable bias. If we have an omitted variable that is correlated with our independent variable, we can't tell if the relationship we see is due to the independent variable or the omitted variable. But if we have a fixed effect for the omitted variable, we can remove the variation between units and then look at the variation within units.

In practice, fixed effects amount to de-meaning our variables of interest. There are a handful of ways to do this.

Panel models

A panel dataset is one in which we view a single unit over multiple periods of time, so a balanced panel has the same number of observations for each unit. For example, we might have data on 100 countries over 10 years, or 50 US states over 20 years. We can then take unit fixed effects, which lets us compare between years within a single unit. Similarly, we can take time fixed effects to compare between units within a given point in time. If our dataset has other dimensions that vary in a way that is not collinear with unit or time, we can also take a fixed effect for that – though again, you want to be careful about throwing in fixed effects.

Dataset

Let me introduce the dataset we'll be using, `crime4`. It comes from Jeffrey Wooldridge's R package – Dr. Wooldridge is one of the most accomplished professors of econometrics on the planet. I was tipped off about his package by Nick Huntington-Klein's own lecture notes.. The dataset shows county probability of arrest and county crime rate by year.

```
data(crime4)
crime4 %>%
  select(county, year, crmrte, prbarr) %>%
  rename(County = county,
         Year = year,
         CrimeRate = crmrte,
         ProbofArrest = prbarr) %>%
  slice(1:9) %>%
  knitr::kable(note = '...') %>%
  kableExtra::add_footnote('9 rows out of 630. "Prob. of Arrest" is estimated probability of being arre
```

County

Year

CrimeRate

ProbofArrest

1

81

0.0398849

0.2896960

1

82

0.0383449

0.3381110

1

83

0.0303048

0.3304490

1

84

0.0347259

0.3625250

1

85

0.0365730

0.3253950

1

86

0.0347524

0.3260620

1

87

0.0356036

0.2982700

3

81

0.0163921

0.2028990

3

82

0.0190651

0.1622180

3

83

0.0151492

0.1815860

3

84

0.0136621

0.1949860

3

85

0.0120346

0.2068970

3

86

0.0129982

0.1560690

3

87

0.0152532

0.1320290

5

81

0.0093372

0.4065930

5

82

0.0123229

0.3800000

5

83

0.0128064

0.5619050

5

84

0.0138299

0.6666670

5

85

0.0124753

0.6039600

5

86

0.0142385

0.6956520

5

87

0.0129603

0.4444440

7

81

0.0219159

0.4310950

7

82

0.0173807

0.6123350

7

83

0.0242110

0.4194050

7

84

0.0223434

0.4124580

7

85

0.0245848

0.3806550

7

86

0.0241281

0.3080570

7

87

0.0267532

0.3647600

9

81

0.0075178

0.6315790

9

82

0.0087368

0.5671640

9

83

0.0136804

0.3773580

9

84

0.0143262

0.4101800

9

85

0.0133173

0.3935480

9

86

0.0114417

0.4626870

9

87

0.0106232

0.5182190

11

81

0.0188472

0.3696500

11

82

0.0198536

0.4258560

11

83

0.0191668

0.3542440

11

84

0.0168042

0.4464290

11

85

0.0141194

0.5308060

11

86

0.0170081

0.4015440

11

87

0.0146067

0.5246640

13

81

0.0300493

0.3196650

13

82

0.0363259

0.3328850

13

83

0.0325271

0.3260210

13

84

0.0301272

0.3021760

13

85

0.0349786

0.2989620

13

86

0.0335360

0.3092060

13

87

0.0296409

0.3650040

15

81

0.0168055

0.4532580

15

82

0.0200047

0.4447060

15

83

0.0142092

0.3836070

15

84

0.0163460

0.3457140

15

85

0.0160135

0.4239770

15

86

0.0194461

0.4192770

15

87

0.0202814

0.3921110

17

81

0.0186469

0.4839860

17

82

0.0211586

0.4201550

17

83

0.0202915

0.3632000

17

84

0.0179276

0.4223830

17

85

0.0196959

0.4644630

17

86

0.0185879

0.3952570

17

87

0.0304289

0.2515990

19

81

0.0225338

0.3261960

19

82

0.0183782

0.1893130

19

83

0.0145267

0.2322830

19

84

0.0136860

0.2173080

19

85

0.0128511

0.2916670

19

86

0.0174337

0.2448190

19

87

0.0221567

0.1628600

21

81

0.0350348

0.2628540

21

82

0.0367896

0.2405730

21

83

0.0366069

0.2458240

21

84

0.0384395

0.2505430

21

85

0.0413296

0.2581310

21

86

0.0441642

0.2483850

21

87

0.0437355

0.2347600

23

81

0.0319175

0.1943030

23

82

0.0290211

0.2866390

23

83

0.0286164

0.2805220

23

84

0.0275500

0.3346150

23

85

0.0293095

0.2874420

23

86

0.0256248

0.3045770

23

87

0.0269836

0.2891210

25

81

0.0360686

0.2559690

25

82

0.0370917

0.2568110

25

83

0.0312138

0.2912170

25

84

0.0294440

0.2773230

25

85

0.0305075

0.2820430

25

86

0.0319331

0.3016020

25

87

0.0302542

0.3235480

27

81

0.0353349

0.2643300

27

82

0.0371544

0.3036060

27

83

0.0294409

0.2441520

27

84

0.0300368

0.2473580

27

85

0.0355283

0.2218620

27

86

0.0366744

0.2173570

27

87

0.0382489

0.2680180

33

81

0.0200390

0.4845610

33

82

0.0144477

0.4853420

33

83

0.0161148

0.2485550

33

84

0.0141497

0.2596150

33

85

0.0153797

0.2179100

33

86

0.0153723

0.2550720

33

87

0.0159189

0.2709500

35

81

0.0414138

0.2876530

35

82

0.0489555

0.2390770

35

83

0.0412081

0.2854270

35

84

0.0404708

0.2822760

35

85

0.0455489

0.2608440

35

86

0.0417065

0.2522810

35

87

0.0408569

0.2660260

37

81

0.0223026

0.2387860

37

82

0.0201309

0.3251450

37

83

0.0190303

0.3071100

37

84

0.0196145

0.3033380

37

85

0.0207011

0.3715470

37

86

0.0252807

0.3366890

37

87

0.0226017

0.3218670

39

81

0.0158129

0.2918030

39

82

0.0188128

0.2588560

39

83

0.0168941

0.3003000

39

84

0.0142249

0.2992960

39

85

0.0130587

0.2868220

39

86

0.0113079

0.2035400

39

87

0.0119154

0.3083330

41

81

0.0365365

0.3892470

41

82

0.0304537

0.4234690

41

83

0.0220667

0.4634150

41

84

0.0158939

0.4634150

41

85

0.0125242

0.5493830

41

86

0.0246998

0.3323080

41

87

0.0257713

0.3072460

45

81

0.0344852

0.2168800

45

82

0.0319727

0.2194760

45

83

0.0351112

0.2297390

45

84

0.0376378

0.2385440

45

85

0.0385400

0.2157890

45

86

0.0312755

0.2781930

45

87

0.0362807

0.2026270

47

81

0.0319193

0.3025260

47

82

0.0305205

0.3369270

47

83

0.0291268

0.2103420

47

84

0.0296451

0.1788620

47

85

0.0293452

0.2356240

47

86

0.0319310

0.1888470

47

87

0.0313623

0.1829270

49

81

0.0359381

0.2707760

49

82

0.0361096

0.3104120

49

83

0.0359286

0.2726240

49

84

0.0319024

0.3261410

49

85

0.0301830

0.3228880

49

86

0.0320147

0.2531120

49

87

0.0374979

0.2644200

51

81

0.0630904

0.2013160

51

82

0.0628671

0.2237990

51

83

0.0600560

0.2143510

51

84

0.0591762

0.1901000

51

85

0.0712392

0.1715190

51

86

0.0704214

0.1661310

51

87

0.0883849

0.1552480

53

81

0.0196669

0.2522520

53

82

0.0199702

0.1842110

53

83

0.0222781

0.2178990

53

84

0.0138866

0.2289160

53

85

0.0170847

0.1590910

53

86

0.0138049

0.3370790

53

87

0.0140655

0.3031910

55

81

0.0397498

0.1434260

55

82

0.0467392

0.1641540

55

83

0.0530843

0.1343070

55

84

0.0504352

0.1805010

55

85

0.0635231

0.2028850

55

86

0.0747550

0.2575640

55

87

0.0790163

0.2246280

57

81

0.0368013

0.2051650

57

82

0.0312481

0.2819310

57

83

0.0290823

0.2751540

57

84

0.0258322

0.2905900

57

85

0.0259898

0.3036720

57

86

0.0277820

0.2914250

57

87

0.0300216

0.2220020

59

81

0.0183936

0.2576420

59

82

0.0160810

0.2098770

59

83

0.0159541

0.2339900

59

84

0.0160226

0.3011760

59

85

0.0159240

0.2316510

59

86

0.0193287

0.7962620

59

87

0.0233327

0.2277530

61

81

0.0248057

0.1917000

61

82

0.0225081

0.2797810

61

83

0.0177485

0.4115230

61

84

0.0169917

0.3965270

61

85

0.0179394

0.4959240

61

86

0.0215632

0.3435370

61

87

0.0233677

0.3981190

63

81

0.0813022

0.1574540

63

82

0.0785857

0.1556250

63

83

0.0760822

0.1407420

63

84

0.0739590

0.1289800

63

85

0.0703023

0.1427940

63

86

0.0690603

0.1130180

63

87

0.0706599

0.1332250

65

81

0.0541680

0.1303320

65

82

0.0524684

0.1586040

65

83

0.0458803

0.1552600

65

84

0.0481331

0.1359910

65

85

0.0603418

0.3117580

65

86

0.0636435

0.2867220

65

87

0.0658801

0.2873300

67

81

0.0619138

0.1850280

67

82

0.0623111

0.2031600

67

83

0.0593568

0.2055250

67

84

0.0518556

0.2202710

67

85

0.0475793

0.2225610

67

86

0.0573538

0.2010520

67

87

0.0614177

0.2172150

69

81

0.0151278

0.4301800

69

82

0.0125657

0.4664880

69

83

0.0140695

0.4691940

69

84

0.0129610

0.4923860

69

85

0.0123078

0.3863010

69

86

0.0166801

0.3048330

69

87

0.0173158

0.2835050

71

81

0.0578315

0.2000210

71

82

0.0674420

0.1795220

71

83

0.0609887

0.1629850

71

84

0.0585782

0.1686590

71

85

0.0571815

0.2199470

71

86

0.0555218

0.2494630

71

87

0.0544061

0.2431190

77

81

0.0321369

0.2144310

77

82

0.0380832

0.2138550

77

83

0.0347969

0.2332790

77

84

0.0352097

0.2785210

77

85

0.0398521

0.2063710

77

86

0.0432589

0.1796970

77

87

0.0441957

0.1908760

79

81

0.0187770

0.3322370

79

82

0.0166299

0.2987550

79

83

0.0137267

0.2288560

79

84

0.0115195

0.3571430

79

85

0.0118946

0.3163270

79

86

0.0112693

0.2419350

79

87

0.0156759

0.4115380

81

81

0.0634680

0.2592100

81

82

0.0597609

0.2800190

81

83

0.0524466

0.2887800

81

84

0.0485070

0.2810990

81

85

0.0527319

0.3093800

81

86

0.0570522

0.2828280

81

87

0.0604498

0.3002150

83

81

0.0314616

0.4064520

83

82

0.0315983

0.4081990

83

83

0.0265988

0.4552340

83

84

0.0268714

0.3498990

83

85

0.0289042

0.3984870

83

86

0.0265029

0.4551680

83

87

0.0315752

0.4563940

85

81

0.0383181

0.2296710

85

82

0.0448326

0.2076020

85

83

0.0411592

0.2135540

85

84

0.0354902

0.2327470

85

85

0.0372284

0.2036320

85

86

0.0419200

0.1897010

85

87

0.0490712

0.1461320

87

81

0.0249672

0.2718040

87

82

0.0294296

0.2535510

87

83

0.0292305

0.2646850

87

84

0.0299963

0.2641770

87

85

0.0302089

0.2846150

87

86

0.0235257

0.2670810

87

87

0.0286781

0.2151080

89

81

0.0268278

0.3415250

89

82

0.0290652

0.2679700

89

83

0.0266806

0.2703040

89

84

0.0222881

0.3408770

89

85

0.0268714

0.2871060

89

86

0.0286617

0.2298370

89

87

0.0283836

0.2966460

91

81

0.0327883

0.3413980

91

82

0.0339434

0.3645700

91

83

0.0288930

0.3358210

91

84

0.0292986

0.3451330

91

85

0.0265877

0.2812010

91

86

0.0315165

0.3541110

91

87

0.0384263

0.3430740

93

81

0.0371141

0.3324640

93

82

0.0312410

0.3185300

93

83

0.0292154

0.3273910

93

84

0.0336595

0.3162850

93

85

0.0313736

0.3087820

93

86

0.0384803

0.3175520

93

87

0.0362222

0.3389020

97

81

0.0417355

0.2926830

97

82

0.0427797

0.2670910

97

83

0.0363496

0.2450030

97

84

0.0345321

0.3359160

97

85

0.0359180

0.2706530

97

86

0.0408816

0.2408190

97

87

0.0334506

0.3022310

99

81

0.0186696

0.1178860

99

82

0.0198469

0.1417770

99

83

0.0155948

0.0595238

99

84

0.0150138

0.1176470

99

85

0.0132968

0.1151520

99

86

0.0185803

0.1203500

99

87

0.0171865

0.1538460

101

81

0.0306114

0.1738530

101

82

0.0331686

0.1839080

101

83

0.0284245

0.1843660

101

84

0.0300807

0.1821510

101

85

0.0315835

0.1590320

101

86

0.0377065

0.4193900

101

87

0.0409403

0.1499360

105

81

0.0626787

0.2553280

105

82

0.0579285

0.2786140

105

83

0.0482863

0.2669190

105

84

0.0354481

0.3000000

105

85

0.0445928

0.2685710

105

86

0.0504341

0.2943770

105

87

0.0514152

0.3814000

107

81

0.0434509

0.2317350

107

82

0.0465892

0.2252630

107

83

0.0489079

0.2461130

107

84

0.0479757

0.2339990

107

85

0.0451353

0.2954130

107

86

0.0450373

0.3064990

107

87

0.0497552

0.2128950

109

81

0.0149547

0.3075730

109

82

0.0174821

0.2078430

109

83

0.0189750

0.2348030

109

84

0.0193141

0.2219630

109

85

0.0199467

0.2443440

109

86

0.0178687

0.3462500

109

87

0.0230995

0.1627690

111

81

0.0231378

0.3212120

111

82

0.0207409

0.2446520

111

83

0.0130077

0.3523210

111

84

0.0127420

0.3829790

111

85

0.0144764

0.3567250

111

86

0.0170996

0.2454700

111

87

0.0183048

0.2021120

113

81

0.0121416

0.4176710

113

82

0.0093530

0.5463920

113

83

0.0118809

0.4257030

113

84

0.0111992

0.5320000

113

85

0.0115498

0.5019010

113

86

0.0111381

0.2334630

113

87

0.0142071

0.1798780

115

81

0.0053566

0.3125000

115

82

0.0036596

1.0000000

115

83

0.0018116

2.7500000

115

84

0.0018815

1.0000000

115

85

0.0047949

0.8888890

115

86

0.0047518

1.0000000

115

87

0.0055332

1.0909100

117

81

0.0127531

0.6090910

117

82

0.0175385

0.2723310

117

83

0.0145596

0.3194810

117

84

0.0188914

0.2614770

117

85

0.0177404

0.3073680

117

86

0.0228867

0.2491800

117

87

0.0268723

0.3704740

119

81

0.0816495

0.1729610

119

82

0.0890352

0.1759040

119

83

0.0854819

0.1722800

119

84

0.0868929

0.1643840

119

85

0.0874663

0.1515410

119

86

0.0913066

0.1567710

119

87

0.0989659

0.1490940

123

81

0.0374067

0.4309330

123

82

0.0354991

0.4674050

123

83

0.0333204

0.4007780

123

84

0.0302041

0.4551920

123

85

0.0321128

0.4311930

123

86

0.0338801

0.3902120

123

87

0.0300184

0.4874300

125

81

0.0330362

0.3118990

125

82

0.0295613

0.2908370

125

83

0.0264993

0.3299120

125

84

0.0240551

0.2982890

125

85

0.0230652

0.3505320

125

86

0.0237337

0.2888540

125

87

0.0266287

0.2690430

127

81

0.0495042

0.5825040

127

82

0.0505650

0.4540350

127

83

0.0439535

0.4378320

127

84

0.0448501

0.3844160

127

85

0.0258142

0.1554500

127

86

0.0259867

0.2050560

127

87

0.0291496

0.1796160

129

81

0.0839218

0.2168070

129

82

0.0777385

0.2368740

129

83

0.0634987

0.2506260

129

84

0.0661003

0.2237920

129

85

0.0709140

0.2244740

129

86

0.0763696

0.2336990

129

87

0.0834982

0.2366010

131

81

0.0218448

0.5423730

131

82

0.0202951

0.4988240

131

83

0.0155024

0.5030490

131

84

0.0137993

0.5876290

131

85

0.0144598

0.7777780

131

86

0.0155081

0.6418440

131

87

0.0189848

0.6890240

133

81

0.0418210

0.2298510

133

82

0.0405609

0.2347420

133

83

0.0393081

0.2133480

133

84

0.0381837

0.2249010

133

85

0.0517775

0.2396680

133

86

0.0535404

0.2444570

133

87

0.0551287

0.2669600

135

81

0.0570824

0.1345160

135

82

0.0540804

0.1370990

135

83

0.0516532

0.1289460

135

84

0.0524006

0.1096840

135

85

0.0517818

0.1062410

135

86

0.0554268

0.0906305

135

87

0.0628972

0.0927700

137

81

0.0127292

0.2537310

137

82

0.0204753

0.2339450

137

83

0.0175680

0.1117320

137

84

0.0106549

0.2018350

137

85

0.0108666

0.2711860

137

86

0.0124818

0.2773720

137

87

0.0126662

0.2071430

139

81

0.0278244

0.6287670

139

82

0.0272460

0.5269710

139

83

0.0270391

0.5241380

139

84

0.0295938

0.5298850

139

85

0.0362156

0.4085710

139

86

0.0290230

0.4201880

139

87

0.0243470

0.5226960

141

81

0.0287834

0.2330250

141

82

0.0267457

0.2233170

141

83

0.0907109

0.2980770

141

84

0.0847914

0.3809520

141

85

0.0609911

0.2083330

141

86

0.1638350

0.1185190

141

87

0.0314610

0.2386360

143

81

0.0240664

0.5818970

143

82

0.0165128

0.4906830

143

83

0.0220282

0.4746540

143

84

0.0230636

0.4292040

143

85

0.0174132

0.4682080

143

86

0.0211692

0.2477060

143

87

0.0265806

0.3178570

145

81

0.0285040

0.2106510

145

82

0.0373545

0.3339290

145

83

0.0275623

0.3892220

145

84

0.0232703

0.4110170

145

85

0.0271005

0.3818850

145

86

0.0257495

0.3709880

145

87

0.0299856

0.3547330

147

81

0.0515935

0.1861970

147

82

0.0585298

0.2126580

147

83

0.0509100

0.2063870

147

84

0.0508713

0.2004260

147

85

0.0525701

0.1911070

147

86

0.0490622

0.2288210

147

87

0.0551686

0.2215420

149

81

0.0209532

0.0588235

149

82

0.0185088

0.1788620

149

83

0.0160101

0.3209300

149

84

0.0137135

0.1842110

149

85

0.0136124

0.2485550

149

86

0.0117459

0.2426040

149

87

0.0164987

0.2719670

151

81

0.0241835

0.2605790

151

82

0.0248051

0.2619150

151

83

0.0208599

0.2506320

151

84

0.0215881

0.2375180

151

85

0.0255308

0.2504080

151

86

0.0279068

0.2240680

151

87

0.0264557

0.2991980

153

81

0.0372807

0.2768430

153

82

0.0313830

0.4103610

153

83

0.0315881

0.3505360

153

84

0.0301159

0.3364620

153

85

0.0282947

0.3199070

153

86

0.0266522

0.3292680

153

87

0.0317563

0.3453680

155

81

0.0441616

0.2754050

155

82

0.0358587

0.3049550

155

83

0.0283191

0.3495710

155

84

0.0284567

0.3902520

155

85

0.0311808

0.4131750

155

86

0.0316701

0.4443780

155

87

0.0312279

0.4082000

157

81

0.0360251

0.2605700

157

82

0.0316020

0.3527890

157

83

0.0314258

0.2827210

157

84

0.0266499

0.3154010

157

85

0.0274601

0.3078240

157

86

0.0256915

0.2960770

157

87

0.0305908

0.2782870

159

81

0.0351730

0.2803260

159

82

0.0292244

0.3421900

159

83

0.0287771

0.2630820

159

84

0.0287943

0.2511690

159

85

0.0317770

0.2302610

159

86

0.0308711

0.2338530

159

87

0.0362330

0.2435900

161

81

0.0222742

0.3507030

161

82

0.0207294

0.3014060

161

83

0.0196418

0.2980390

161

84

0.0171226

0.3912570

161

85

0.0198354

0.3315410

161

86

0.0218999

0.2870890

161

87

0.0200070

0.4824250

163

81

0.0248116

0.2343100

163

82

0.0224186

0.2446910

163

83

0.0215107

0.3228570

163

84

0.0191925

0.3275680

163

85

0.0203012

0.3118280

163

86

0.0172874

0.3263650

163

87

0.0215728

0.3109870

165

81

0.0477432

0.3595580

165

82

0.0425763

0.3946250

165

83

0.0421413

0.4033070

165

84

0.0408157

0.3036110

165

85

0.0440056

0.3381440

165

86

0.0391644

0.4060560

165

87

0.0508341

0.3406790

167

81

0.0294232

0.4030470

167

82

0.0260284

0.3955110

167

83

0.0183428

0.3891300

167

84

0.0181909

0.4608600

167

85

0.0218960

0.4205690

167

86

0.0213432

0.4136320

167

87

0.0238285

0.3622700

169

81

0.0145053

0.2956880

169

82

0.0165503

0.2241990

169

83

0.0125903

0.2384260

169

84

0.0109940

0.3610390

169

85

0.0092151

0.3777090

169

86

0.0118263

0.2153110

169

87

0.0121033

0.3433870

171

81

0.0189838

0.1534440

171

82

0.0209131

0.1510170

171

83

0.0232080

0.1437540

171

84

0.0213532

0.1287360

171

85

0.0235205

0.1099440

171

86

0.0209747

0.1660140

171

87

0.0243954

0.1756490

173

81

0.0134337

0.1515150

173

82

0.0122786

0.1639340

173

83

0.0163379

0.5487800

173

84

0.0103730

0.4038460

173

85

0.0047703

0.2894740

173

86

0.0090203

0.2500000

173

87

0.0139937

0.5304350

175

81

0.0175623

0.3069540

175

82

0.0164474

0.3721520

175

83

0.0170197

0.3268770

175

84

0.0242597

0.1617650

175

85

0.0189785

0.3086680

175

86

0.0171221

0.2671230

175

87

0.0164932

0.3503480

179

81

0.0283660

0.4444440

179

82

0.0258779

0.3542480

179

83

0.0246299

0.2534640

179

84

0.0256946

0.3046040

179

85

0.0277412

0.3542070

179

86

0.0314961

0.3337540

179

87

0.0318720

0.3775430

181

81

0.0563677

0.2200380

181

82

0.0617686

0.1782010

181

83

0.0605235

0.1695060

181

84

0.0534929

0.2227120

181

85

0.0470632

0.2068770

181

86

0.0626827

0.1773520

181

87

0.0729479

0.1825900

183

81

0.0601894

0.1789080

183

82

0.0583432

0.1992970

183

83

0.0512834

0.1954750

183

84

0.0472032

0.2069690

183

85

0.0480699

0.1884730

183

86

0.0530244

0.2067920

183

87

0.0568423

0.2042160

185

81

0.0107527

0.7000000

185

82

0.0148936

1.0714300

185

83

0.0105374

1.2000000

185

84

0.0151844

1.2142900

185

85

0.0078125

0.2500000

185

86

0.0045815

0.2957750

185

87

0.0108703

0.1952660

187

81

0.0338507

0.2655600

187

82

0.0312457

0.2644440

187

83

0.0221291

0.2732920

187

84

0.0246110

0.2701150

187

85

0.0254524

0.2630060

187

86

0.0239377

0.3525840

187

87

0.0345231

0.3326690

189

81

0.0314674

0.1816390

189

82

0.0316963

0.1957360

189

83

0.0269621

0.2491540

189

84

0.0245914

0.2541770

189

85

0.0271844

0.1532610

189

86

0.0287322

0.1507690

189

87

0.0313130

0.1613810

191

81

0.0371162

0.2705590

191

82

0.0404994

0.2781920

191

83

0.0365370

0.2219480

191

84

0.0330351

0.2598230

191

85

0.0345692

0.2539680

191

86

0.0345793

0.2103540

191

87

0.0458895

0.1722570

193

81

0.0192107

0.3900090

193

82

0.0240456

0.3307750

193

83

0.0220034

0.3501890

193

84

0.0171735

0.3429670

193

85

0.0180945

0.3566820

193

86

0.0194505

0.3410060

193

87

0.0235277

0.2660550

195

81

0.0631212

0.2236710

195

82

0.0588861

0.2769070

195

83

0.0573422

0.2311030

195

84

0.0551157

0.1569010

195

85

0.0236432

0.2469700

195

86

0.0300950

0.2087380

195

87

0.0313973

0.2013970

197

81

0.0178621

0.1589150

197

82

0.0180711

0.2291670

197

83

0.0155747

0.2266670

197

84

0.0136619

0.2041880

197

85

0.0130857

0.1805560

197

86

0.0128740

0.1126760

197

87

0.0141928

0.2075950

9 rows out of 630. "Prob. of Arrest" is estimated probability of being arrested when you commit a crime

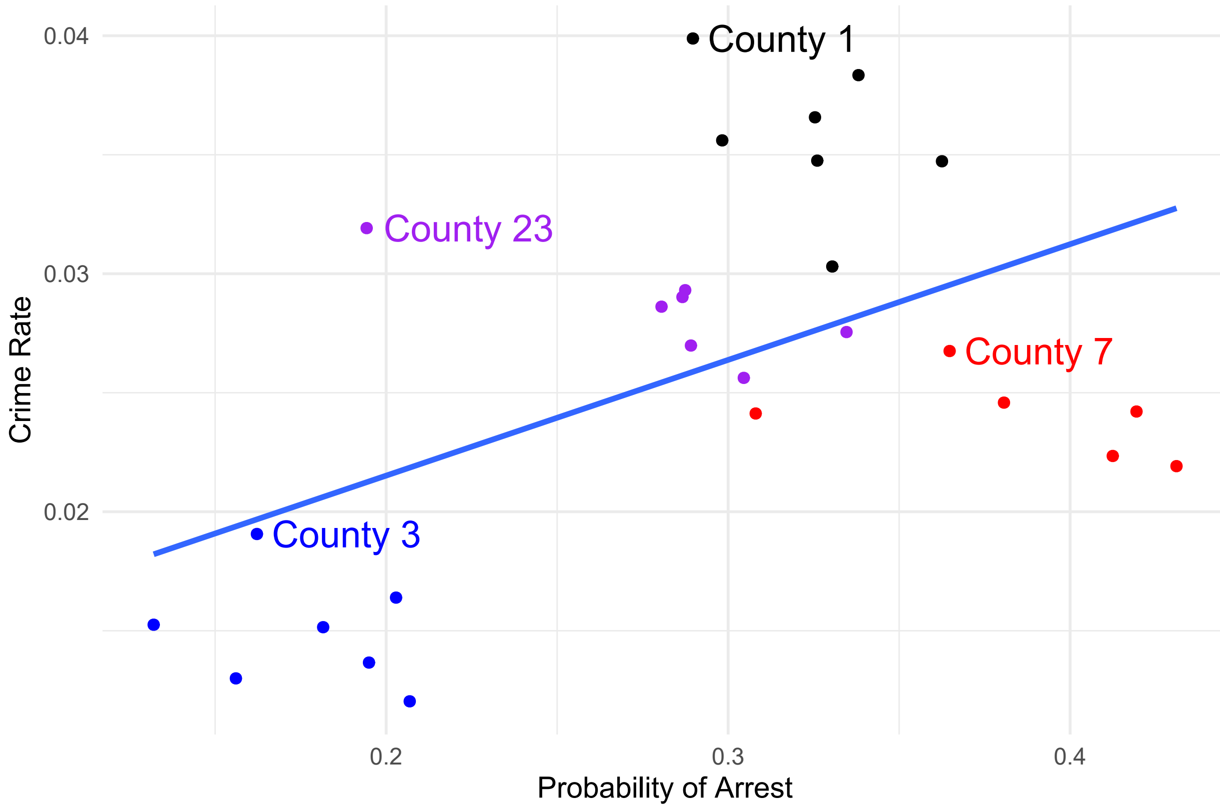
Let's visualize it

Below I visualize the data for just a few counties. Note the positive slope when pooling! Is that surprising?

```
crime4 %>%
  filter(county %in% c(1,3,7, 23),
         prbarr < .5) %>%
  group_by(county) %>%
  mutate(label = case_when(
    crmrte == max(crmrte) ~ paste('County',county),
    TRUE ~ NA_character_
  )) %>%
  ggplot(aes(x =  prbarr, y = crmrte, color = factor(county), label = label)) +
  geom_point() +
  geom_text(hjust = -.1, size = 14/.pt) +
  labs(x = 'Probability of Arrest',
       y = 'Crime Rate',
       caption = 'One outlier eliminated in County 7.') +
```

```r
  #scale_x_continuous(limits = c(.15, 2.5)) +
  guides(color = FALSE, label = FALSE) +
  scale_color_manual(values = c('black','blue','red','purple')) +
  geom_smooth(method = 'lm', aes(color = NULL, label = NULL), se = FALSE)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



One outlier eliminated in County 7.

Let's try the de-meaning approach

We can use `group_by` to get means-within-groups and subtract them out.

```r
crime4 <- crime4 %>%
  group_by(county) %>%
  mutate(mean_crime = mean(crmrte),
         mean_prob = mean(prbarr)) %>%
  mutate(demeaned_crime = crmrte - mean_crime,
         demeaned_prbarr = prbarr - mean_prob)
```

And Regress!

```r
orig_data <- lm(crmrte ~ prbarr, data = crime4)
de_mean <- lm(demeaned_crime ~ demeaned_prbarr, data = crime4)
msummary(list(orig_data, de_mean))
```

Note the coefficient has flipped!

	(1)	(2)
(Intercept)	0.043	0.000
	(0.001)	(0.000)
prbarr	−0.038	
	(0.004)	
demeaned_prbarr		−0.002
		(0.002)
Num.Obs.	630	630
R2	0.129	0.001
R2 Adj.	0.127	−0.001
AIC	−3347.3	−4549.6
BIC	−3334.0	−4536.3
Log.Lik.	1676.651	2277.823
F	92.646	
RMSE	0.02	0.01

Interpreting a Within Relationship

How can we interpret that slope of `-0.02`? This is all *within variation* so our interpretation must be *within-county*. So, "comparing a county in year A where its arrest probability is 1 (100 percentage points) higher than it is in year B, we expect the number of crimes per person to drop by .02." Or if we think we've causally identified it (and want to work on a more realistic scale), "raising the arrest probability by 1 percentage point in a county reduces the number of crimes per person in that county by .0002". We're basically "controlling for county" (and will do that explicitly in a moment). So your interpretation should think of it in that way - *holding county constant* i.e. *comparing two observations with the same value of county* i.e. *comparing a county to itself at a different point in time*.

Concept checks

- Do you think the model we've presented is sufficient to have a causal interpretation of the effect of arrest probability on crime?
- What assumptions would we need to make to have a causal interpretation?
- What potential confounders are there?
- Why does subtracting the within-individual mean of each variable "control for individual"?
- In a sentence, interpret the slope coefficient in the estimated model $(Y_{it} - \bar{Y}_i) = 2 + 3(X_{it} - \bar{X}_i)$ where Y is "blood pressure", X is "stress at work", and i is an individual person, and $\bar{Y}_i$ means average of Y_i
- Is this relationship causal? If not, what assumptions are required for it to be causal?

Can we do that all at once? Yes, with the Least Squares Dummy Variable Approach

De-meaning takes some steps which could get tedious to write out. Another way is to include a dummy or category variable for each county. This is called the Least Squares Dummy Variable approach.

You end up with the same results as if we de-meaned.

```
lsdv <- lm(crmrte ~ prbarr + factor(county), data = crime4)
msummary(list(orig_data, de_mean, lsdv), keep = c('prbarr', 'demeaned_prob'))
```

Hey look, the coefficient is the same!

Why LSDV?

- A benefit of the LSDV approach is that it calculates the fixed effects α_i for you
- We left those out of the table with the `coefs` argument of `export_summs` (we rarely want them) but here they are:

	(1)	(2)	(3)
prbarr	−0.038		−0.002
	(0.004)		(0.003)
demeaned_prbarr		−0.002	
		(0.002)	
Num.Obs.	630	630	630
R2	0.129	0.001	0.871
R2 Adj.	0.127	−0.001	0.849
AIC	−3347.3	−4549.6	−4371.6
BIC	−3334.0	−4536.3	−3962.6
Log.Lik.	1676.651	2277.823	2277.823
F	92.646		40.351
RMSE	0.02	0.01	0.01

```
lsdv
```

```
##
## Call:
## lm(formula = crmrte ~ prbarr + factor(county), data = crime4)
##
## Coefficients:
##       (Intercept)            prbarr    factor(county)3    factor(county)5
##         0.0363976        -0.0020232         -0.0211038         -0.0227439
##   factor(county)7   factor(county)9   factor(county)11   factor(county)13
##        -0.0125058        -0.0240486         -0.0183143         -0.0032912
##  factor(county)15  factor(county)17   factor(county)19   factor(county)21
##        -0.0179836        -0.0146255         -0.0185499          0.0035485
##  factor(county)23  factor(county)25   factor(county)27   factor(county)33
##        -0.0073943        -0.0034639         -0.0012558         -0.0198379
##  factor(county)35  factor(county)37   factor(county)39   factor(county)41
##         0.0070240        -0.0143802         -0.0212591         -0.0115589
##  factor(county)45  factor(county)47   factor(county)49   factor(county)51
##        -0.0008915        -0.0053747         -0.0015888          0.0318754
##  factor(county)53  factor(county)55   factor(county)57   factor(county)59
##        -0.0186603         0.0221664         -0.0063204         -0.0178825
##  factor(county)61  factor(county)63   factor(county)65   factor(county)67
##        -0.0149666         0.0381621          0.0198140          0.0214212
##  factor(county)69  factor(county)71   factor(county)77   factor(county)79
##        -0.0211463         0.0228639          0.0022599         -0.0215523
##  factor(county)81  factor(county)83   factor(county)85   factor(county)87
##         0.0205261        -0.0064776          0.0051594         -0.0078661
##  factor(county)89  factor(county)91   factor(county)93   factor(county)97
##        -0.0088413        -0.0040777         -0.0018436          0.0021169
##  factor(county)99 factor(county)101  factor(county)105  factor(county)107
##        -0.0192747        -0.0027612          0.0143055          0.0108018
## factor(county)109 factor(county)111  factor(county)113  factor(county)115
##        -0.0170930        -0.0187163         -0.0239391         -0.0301032
## factor(county)117 factor(county)119  factor(county)123  factor(county)125
##        -0.0169581         0.0526182         -0.0023063         -0.0091250
## factor(county)127 factor(county)129  factor(county)131  factor(county)133
##         0.0028419         0.0386488         -0.0179728          0.0098405
## factor(county)135 factor(county)137  factor(county)139  factor(county)141
```

```
##          0.0188796          -0.0220273          -0.0066127           0.0337109
## factor(county)143  factor(county)145  factor(county)147  factor(county)149
##         -0.0139798          -0.0071850           0.0166929          -0.0200991
## factor(county)151  factor(county)153  factor(county)155  factor(county)157
##         -0.0114062          -0.0047028          -0.0026681          -0.0058717
## factor(county)159  factor(county)161  factor(county)163  factor(county)165
##         -0.0043145          -0.0154759          -0.0147833           0.0082355
## factor(county)167  factor(county)169  factor(county)171  factor(county)173
##         -0.0128534          -0.0232628          -0.0141934          -0.0242636
## factor(county)175  factor(county)179  factor(county)181  factor(county)183
##         -0.0175234          -0.0077435           0.0232585           0.0175664
## factor(county)185  factor(county)187  factor(county)189  factor(county)191
##         -0.0243118          -0.0078490          -0.0071590           0.0015451
## factor(county)193  factor(county)195  factor(county)197
##         -0.0152095           0.0097064          -0.0209701
```

THe interpretation is exactly the same as with a categorical variable - we have an omitted county, and these show the difference relative to that omitted county

NOTE: See how I put `factor()` around county? That is to ensure it reads county, which is the county fips code as a categorical variable instead of as a numerical variable. If you don't do that, it will read it as a numerical variable and you'll get a different result:

```r
lm(crmrte ~ prbarr + county, data = crime4)
```

```
##
## Call:
## lm(formula = crmrte ~ prbarr + county, data = crime4)
##
## Coefficients:
## (Intercept)       prbarr       county
##   4.213e-02   -3.788e-02    1.094e-05
```

This is saying that as FIPS code increases by one, the crime rate increases by 0.000011… that's nonsense. There's an urban legend of an economist who took the log of the NAICS industry classification code for quite some time before realizing they meant to use a categorical variable. Correcting that mistake completely changed their results.

Why LSDV?

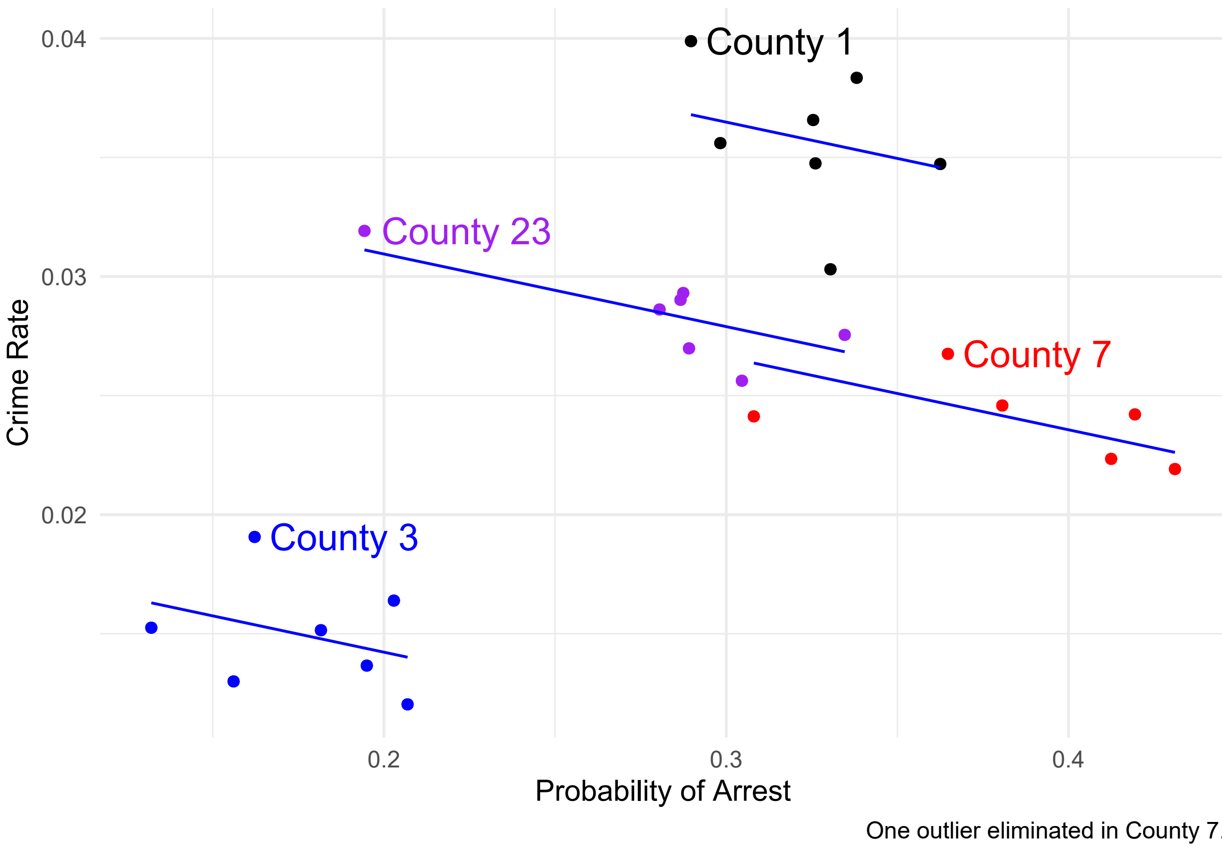
This also makes clear another element of what's happening! Just like with a categorical var, the line is moving *up and down* to meet the counties. Graphically, de-meaning moves all the points together in the middle to draw a line, while LSDV moves the line up and down to meet the points

```r
crime4_small <- crime4 %>%
  filter(county %in% c(1,3,7, 23), # filter down data points
         prbarr < .5) %>%
  ungroup()
# Make lsdv for this small dataframe
lsdv_small <- lm(crmrte ~ prbarr + factor(county),
  data = crime4_small)

crime4_small %>%
  mutate(pred = predict(lsdv_small)) %>%
  group_by(county) %>%
  mutate(label = case_when(
    crmrte == max(crmrte) ~ paste('County',county),
    TRUE ~ NA_character_
```

```r
)) %>%
ggplot(aes(x =  prbarr, y = crmrte, color = factor(county), label = label)) +
geom_point() +
geom_text(hjust = -.1, size = 14/.pt) +
geom_line(aes(y = pred, group = county), color = 'blue') +
labs(x = 'Probability of Arrest',
     y = 'Crime Rate',
     caption = 'One outlier eliminated in County 7.') +
#scale_x_continuous(limits = c(.15, 2.5)) +
guides(color = FALSE, label = FALSE) +
scale_color_manual(values = c('black','blue','red','purple'))
```

```
## Warning: Removed 23 rows containing missing values (`geom_text()`).
```



The "Pros" don't use LSDV

Most people do not use LSDB – it is computationally expensive. If you get too many fixed effects or too big of data, it just will not wrong. The professionally-written commands use de-meaning, like **fixest**, which is less computationally expensive. See for yourself! Look, we even used the **etable** function.

```r
pro <- feols(crmrte ~ prbarr | county, data = crime4)
de_mean <- feols(demeaned_crime ~ demeaned_prbarr, data = crime4)
etable(de_mean, pro)
```

```
##                              de_mean             pro
## Dependent Var.:       demeaned_crime          crmrte
##
## Constant         -1.01e-20 (0.0003)
```

```
## demeaned_prbarr   -0.0020 (0.0025)
## prbarr                               -0.0020 (0.0026)
## Fixed-Effects:  ------------------ ----------------
## county                          No              Yes
## _______________ __________________ ________________
## S.E. type                      IID      by: county
## Observations                   630             630
## R2                         0.00106         0.87076
## Within R2                       --         0.00106
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

To explain the **fixest** package, let's dive a bit deeper into the crime data. It has tons of variables we could use. We could account for variation by year for example.

```
crime_county_fe <- feols(crmrte ~ prbarr | county, data = crime4)
crime_year_fe  <-  feols(crmrte ~ prbarr | year, data = crime4)
crime_county_year_fe <- feols(crmrte ~ prbarr | county+year, data = crime4)

etable(list('County FE'=crime_county_fe,
            'Year FE'=crime_year_fe,
            'County and Year FE'=crime_county_year_fe))
```

```
##                          County FE            Year FE County and Yea..
## Dependent Var.:             crmrte             crmrte           crmrte
##
## prbarr          -0.0020 (0.0026) -0.0378** (0.0090) -0.0011 (0.0026)
## Fixed-Effects:  ---------------- ------------------ ----------------
## county                       Yes                 No              Yes
## year                          No                Yes              Yes
## _______________ ________________ __________________ ________________
## S.E.: Clustered       by: county            by: year      by: county
## Observations                 630                630              630
## R2                       0.87076            0.13347          0.87735
## Within R2                0.00106            0.12764          0.00034
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Pretty neat right? Just sticking something after the | allows you to residualize its fixed effect!

```
dict = c('prbarr'='Prob. Arrest',
  'avgsen'='Avg. Sentence',
  'county'='County',
  'year'= 'Year',
  'crmrte'='Crime Rate',
  'prbconv'='Prob. Conviction')

etable(list('County FE'=crime_county_fe,
            'Year FE'=crime_year_fe,
            'County and Year FE'=crime_county_year_fe),
       notes='Note: Estimates from various fixed effects regressions on the Crime Data',
       dict=dict
       )
```

```
##                          County FE            Year FE County and Yea..
## Dependent Var.:         Crime Rate         Crime Rate       Crime Rate
```

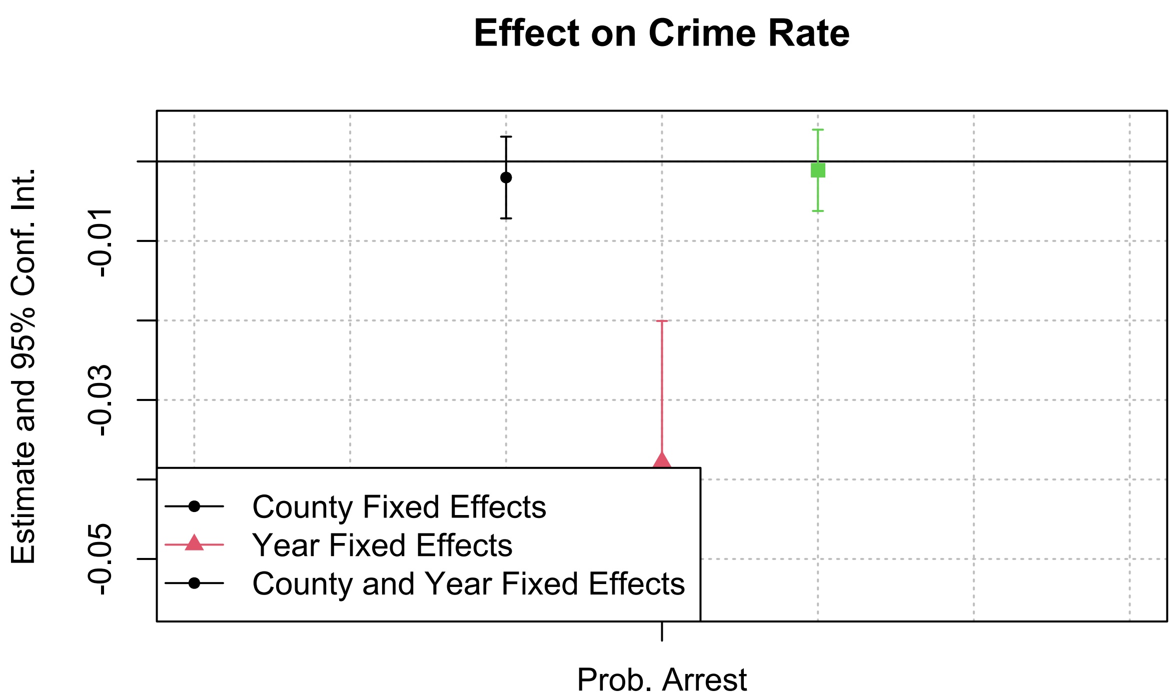
```
##
## Prob. Arrest    -0.0020 (0.0026) -0.0378** (0.0090) -0.0011 (0.0026)
## Fixed-Effects:  ---------------- ------------------ ----------------
## County                       Yes                 No              Yes
## Year                          No                Yes              Yes
## _______________ ________________ __________________ ________________
## S.E.: Clustered       by: County           by: Year       by: County
## Observations                 630                630              630
## R2                       0.87076            0.13347          0.87735
## Within R2                0.00106            0.12764          0.00034
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
# I don't want to keep writing in ,dict=dct. So I'll use setFixestDict
# This applies to every etable in the session
setFixest_dict(dict)
```

Visualization Similarly, the `fixest::coefplot()` function for plotting estimation results:

```r
coefplot(list(crime_county_fe, crime_year_fe, crime_county_year_fe))

## Add legend (optional)
legend("bottomleft", col = 1:2, lwd = 1, pch = c(20, 17),
       legend = c("County Fixed Effects", 'Year Fixed Effects', "County and Year Fixed Effects"))
```

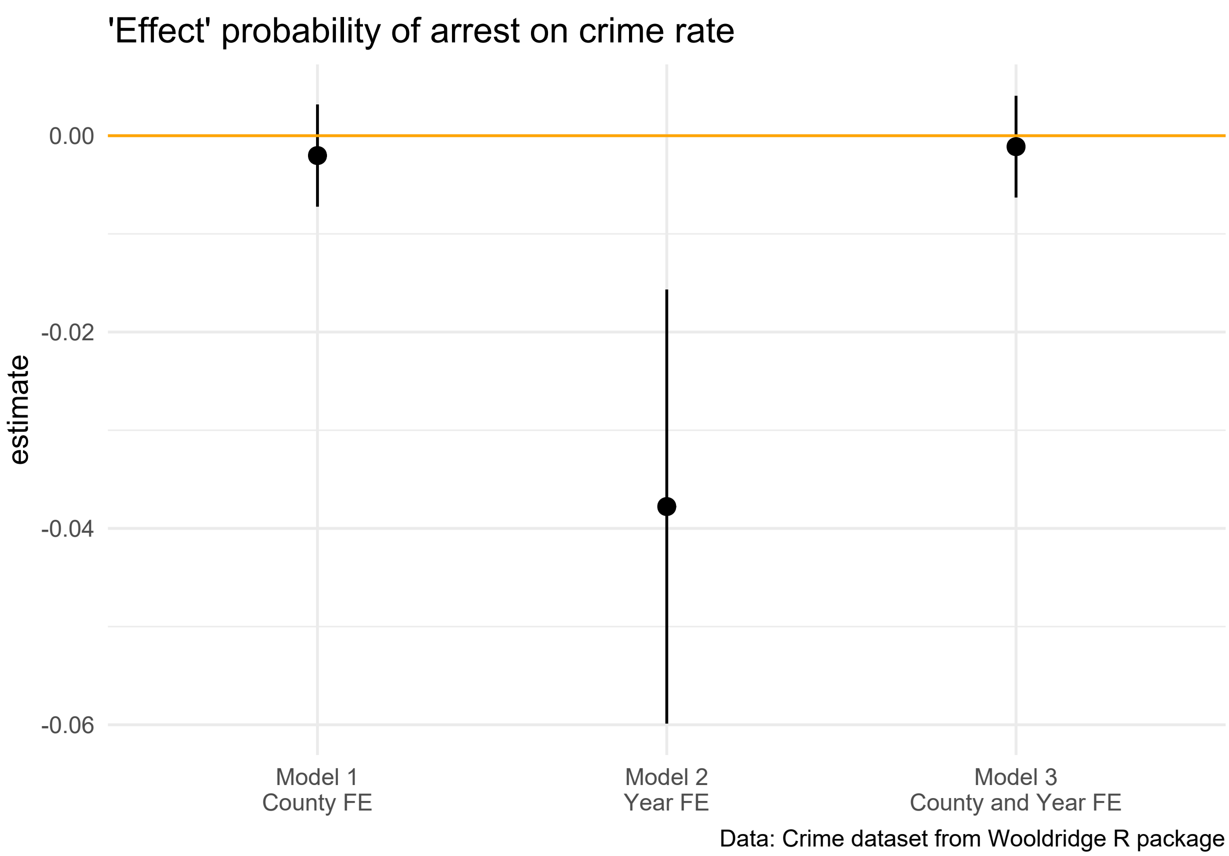


`coefplot()` is especially useful for tracing the evolution of treatment effects over time, as in a difference-in-differences setup (see Examples). However, I realise some people may find it a bit off-putting that it produces base R plots, rather than a **ggplot2** object. We'll get to an automated **ggplot2** coefficient plot solution further below with `modelsummary::modelplot()`. Nevertheless, let me close this out this section by demonstrating the relative ease with which you can do this "manually". Consider the below example, which leverages the fact that we have saved (or can save) regression models as data frames with `broom::tidy()`. As I suggested earlier, this makes it simple to construct our own bespoke coefficient plots.

```r
# library(ggplot2) ## Already loaded

## First get tidied output of the ols_hdfe object
coefs_crime_county_fe = tidy(crime_county_fe, conf.int = TRUE)
coefs_crime_year_fe = tidy(crime_year_fe, conf.int = TRUE)
coefs_crime_county_year_fe = tidy(crime_county_year_fe, conf.int = TRUE)

bind_rows(
  coefs_crime_county_fe %>% mutate(reg = "Model 1\nCounty FE"),
  coefs_crime_year_fe %>% mutate(reg = "Model 2\nYear FE"),
  coefs_crime_county_year_fe %>% mutate(reg="Model 3\nCounty and Year FE")
  ) %>%
  ggplot(aes(x=reg, y=estimate, ymin=conf.low, ymax=conf.high)) +
  geom_pointrange() +
  labs(Title = "Marginal effect of probability of arrest on crime rate") +
  geom_hline(yintercept = 0, col = "orange") +
  labs(
    title = "'Effect' probability of arrest on crime rate",
    caption = "Data: Crime dataset from Wooldridge R package"
    ) +
  theme(axis.title.x = element_blank())
```



What if we wanted to change the clustering of the standard errors? Did you notice the S.E. type above? It auto-clustered by the fixed effects – specifically the fixed effect with the most levels. **fixest** does that by default, but maybe you disagree!

Sometimes you want to cluster standard errors a new way. Well that is something you can do with **fixest** and its delightfully well-designed `etable()` function. You can specify the cluster variable with `cluster()` or the type of standard errors you want with `se()` and get different types of standard errors. Below I specify standard errors clustered by state and then an assumption of independent and identically distributed errors. (The most vanilla standard errors you can assume and rarely the ones we believe explain real world phenomena.)

```
# IID standard errors
etable(list('County FE'=crime_county_fe,
            'Year FE'=crime_year_fe,
            'County and Year FE'=crime_county_year_fe),
        se='IID')
```

```
##                         County FE            Year FE County and Yea..
## Dependent Var.:        Crime Rate          Crime Rate      Crime Rate
##
## Prob. Arrest    -0.0020 (0.0027) -0.0378*** (0.0040) -0.0011 (0.0026)
## Fixed-Effects:  ---------------- ------------------- ----------------
## County                       Yes                  No              Yes
## Year                          No                 Yes              Yes
## _______________ ________________ ___________________ ________________
## S.E. type                    IID                 IID              IID
## Observations                 630                 630              630
## R2                       0.87076             0.13347          0.87735
## Within R2                0.00106             0.12764          0.00034
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
etable(list('County FE'=crime_county_fe,
            'Year FE'=crime_year_fe,
            'County and Year FE'=crime_county_year_fe),
        cluster='county')
```

```
##                         County FE            Year FE County and Yea..
## Dependent Var.:        Crime Rate          Crime Rate      Crime Rate
##
## Prob. Arrest    -0.0020 (0.0026) -0.0378*** (0.0103) -0.0011 (0.0026)
## Fixed-Effects:  ---------------- ------------------- ----------------
## County                       Yes                  No              Yes
## Year                          No                 Yes              Yes
## _______________ ________________ ___________________ ________________
## S.E.: Clustered       by: County          by: County       by: County
## Observations                 630                 630              630
## R2                       0.87076             0.13347          0.87735
## Within R2                0.00106             0.12764          0.00034
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We'd normally expect our standard errors to blow up with clustering and we see something similar here. Why is that?

Yes, I know this is a lot on stuff you've only barely experienced before. But you're going to come across these terms when you read papers and I want you to know how to play with them when you're trying to learn by doing.

Aside on standard errors We've now seen the various options that **fixest** has for specifying different standard error structures. In short, you invoke either of the `se` or `cluster` arguments. Moreover, you can choose to do so either at estimation time, or by adjusting the standard errors for an existing model post-estimation (e.g. with `summary.fixest(mod, cluster = ...)`). There are two additional points that I want to draw your attention to.

First, if you're coming from another statistical language, adjusting the standard errors post-estimation (rather than always at estimation time) may seem slightly odd. But this behaviour is actually extremely powerful, because it allows us to analyse the effect of different error structures *on-the-fly* without having to rerun the entire model again. **fixest** is already the fastest game in town, but just think about the implied time savings for really large models.[1] I'm a huge fan of the flexibility, safety, and speed that on-the-fly standard error adjustment offers us. I even wrote a whole blog post about it if you'd like to read more.

Second, reconciling standard errors across different software is a much more complicated process than you may realise. There are a number of unresolved theoretical issues to consider — especially when it comes to multiway clustering — and package maintainers have to make a number of arbitrary decisions about the best way to account for these. See here for a detailed discussion. Luckily, Laurent (the **fixest** package author) has taken the time to write out a detailed vignette about how to replicate standard errors from other methods and software packages.[2]

Multiple estimations But won't it get tedious writing out all these variations of fixed effects over and over with the `feols()` repeated? Sure will. That's where the **fixest** package comes in handy.

fixest allows you to do multiple estimations in one command and it does is it fast! Why is it so fast? It leverages the de-meaning trick mentioned above. If a fixed effect is used in multiple estimations, it saves the outcome variable de-meaned of that fixed effect to use in all the other estimations. That saves a bunch of time!

This is also a really smart big data technique we'll get into more later in the course. It does a task once instead of multiple times to save time and processing power.

Here's a demo using the stepwise `sw0()` function, which adds fixed effects – starting with none step-by-step:

```
crime_many_fes <- feols(crmrte ~ prbarr |
  sw0(county,year,county+year),
  data=crime4)
etable(crime_many_fes)
```

```
##                    crime_many_fes.1 crime_many_fes.2   crime_many_fes.3
## Dependent Var.:          Crime Rate       Crime Rate         Crime Rate
##
## Constant          0.0432*** (0.0014)
## Prob. Arrest    -0.0379*** (0.0039) -0.0020 (0.0026) -0.0378** (0.0090)
## Fixed-Effects:  ------------------- ---------------- ------------------
## County                           No              Yes                 No
## Year                             No               No                Yes
## _______________ ___________________ ________________ __________________
## S.E. type                       IID      by: County           by: Year
## Observations                    630              630                630
## R2                          0.12856          0.87076            0.13347
## Within R2                        --          0.00106            0.12764
##
##                 crime_many_fes.4
## Dependent Var.:       Crime Rate
##
## Constant
## Prob. Arrest    -0.0011 (0.0026)
## Fixed-Effects:  ----------------
## County                       Yes
## Year                         Yes
## _______________ ________________
## S.E. type             by: County
```

[1] To be clear, adjusting the standard errors via, say, `summary.fixest()` completes instantaneously.

[2] If you want a deep dive into the theory with even more simulations, then this paper by the authors of the **sandwich** paper is another excellent resource.

```
## Observations                        630
## R2                              0.87735
## Within R2                       0.00034
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

These results are the same as above. Oh and guess what? You can get a lot more complicated than that!

Wouldnt it be nice to have better names of our variables? We can do that uing a dict, which is just a fancy vector with names.

Here's the basics of how it works.[3] You can specify:

1. One or more rhs variable using `c(var1,var2,var3)`
2. One or more fixed effects using the stepwise functions `sw()`, `sw0()`, `csw()`, and `csw0()`.
3. One or more independent variable using the stepwise functions `sw()`, `sw0()`, `csw()`, and `csw0()`.
4. Different samples using the `split` or `fsplit` option.

And here's multiple estimations used to their "fuller" potential:

```
crime_many_estimations <- feols(c(crmrte,prbconv) ~ csw(prbarr, avgsen, polpc) |
  sw0(county,year,county+year),
  data=crime4,
  fsplit=~urban)

etable(crime_many_estimations[lhs='crmrte',sample=1],title='Crime Rate',notes='Note: Estimates from var
```

```
##                 crime_many_estim..1 crime_many_estim..2 crime_many_estim..3
## Sample (urban)          Full sample         Full sample         Full sample
## Dependent Var.:          Crime Rate          Crime Rate          Crime Rate
##
## Constant          0.0432*** (0.0014)  0.0406*** (0.0026)  0.0397*** (0.0025)
## Prob. Arrest     -0.0379*** (0.0039) -0.0381*** (0.0039) -0.0478*** (0.0039)
## Avg. Sentence                            0.0003 (0.0003)     0.0003 (0.0002)
## polpc                                                       2.089*** (0.2442)
## Fixed-Effects:   ------------------- ------------------- -------------------
## County                            No                  No                  No
## Year                              No                  No                  No
## _______________  ___________________ ___________________ ___________________
## S.E. type                        IID                 IID                 IID
## Observations                     630                 630                 630
## R2                           0.12856             0.13055             0.22159
## Within R2                         --                  --                  --
##
##                 crime_many_es..4 crime_many_es..5 crime_many_est..6
## Sample (urban)       Full sample      Full sample       Full sample
## Dependent Var.:       Crime Rate       Crime Rate        Crime Rate
##
## Constant
## Prob. Arrest    -0.0020 (0.0026) -0.0019 (0.0027)  -0.0043 (0.0028)
## Avg. Sentence                     7.12e-5 (0.0001)  0.0002* (0.0001)
## polpc                                             1.735*** (0.3191)
## Fixed-Effects:  ---------------- ---------------- -----------------
## County                       Yes              Yes               Yes
## Year                          No               No                No
## _______________ ________________ ________________ _________________
```

[3]You can find a more in-depth explanation at the Multiple Estimation vignette.

```
## S.E. type             by: County       by: County        by: County
## Observations                 630              630               630
## R2                       0.87076          0.87084           0.89669
## Within R2                0.00106          0.00164           0.20150
##
##                   crime_many_esti..7 crime_many_esti..8 crime_many_esti..9
## Sample (urban)           Full sample        Full sample        Full sample
## Dependent Var.:           Crime Rate         Crime Rate         Crime Rate
##
## Constant
## Prob. Arrest     -0.0378** (0.0090) -0.0379** (0.0088) -0.0478** (0.0086)
## Avg. Sentence                          0.0002 (0.0002)    0.0002 (0.0002)
## polpc                                                     2.134* (0.7683)
## Fixed-Effects:   ------------------ ------------------ ------------------
## County                           No                 No                 No
## Year                            Yes                Yes                Yes
## ________________ __________________ __________________ __________________
## S.E. type                  by: Year           by: Year           by: Year
## Observations                    630                630                630
## R2                          0.13347            0.13463            0.22896
## Within R2                   0.12764            0.12881            0.22377
##
##                   crime_many_e..10 crime_many_e..11 crime_many_es..12
## Sample (urban)         Full sample      Full sample       Full sample
## Dependent Var.:         Crime Rate       Crime Rate        Crime Rate
##
## Constant
## Prob. Arrest     -0.0011 (0.0026) -0.0012 (0.0026)  -0.0038 (0.0027)
## Avg. Sentence                     -9.4e-5 (0.0001)  5.05e-5 (0.0001)
## polpc                                              1.821*** (0.3223)
## Fixed-Effects:   ---------------- ---------------- -----------------
## County                        Yes              Yes               Yes
## Year                          Yes              Yes               Yes
## ________________ ________________ ________________ _________________
## S.E. type             by: County       by: County        by: County
## Observations                 630              630               630
## R2                       0.87735          0.87746           0.90563
## Within R2                0.00034          0.00125           0.23086
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
etable(crime_many_estimations[lhs='prbconv',sample=1],title='Probability of Conviction',notes='Note: Es
```

```
##                   crime_many_esti..1 crime_many_es..2 crime_many_est..3
## Sample (urban)           Full sample      Full sample       Full sample
## Dependent Var.:     Prob. Conviction Prob. Conviction  Prob. Conviction
##
## Constant         0.5807*** (0.1385) 0.5018. (0.2628)   0.3759 (0.2338)
## Prob. Arrest        0.3512 (0.3937)  0.3464 (0.3942) -1.029** (0.3662)
## Avg. Sentence                        0.0090 (0.0254)   0.0068 (0.0226)
## polpc                                                296.5*** (22.92)
## Fixed-Effects:   ------------------ ---------------- -----------------
## County                           No               No                No
## Year                             No               No                No
## ________________ __________________ ________________ _________________
```

```
## S.E. type                         IID              IID                IID
## Observations                      630              630                630
## R2                            0.00127          0.00146            0.21216
## Within R2                          --               --                 --
##
##                   crime_many_es..4 crime_many_es..5 crime_many_es..6
## Sample (urban)         Full sample      Full sample      Full sample
## Dependent Var.: Prob. Conviction Prob. Conviction Prob. Conviction
##
## Constant
## Prob. Arrest      -2.941 (2.064)   -2.940 (2.074)   -3.394 (2.559)
## Avg. Sentence                     0.0008 (0.0342)  0.0301 (0.0299)
## polpc                                               328.8* (142.5)
## Fixed-Effects:  ---------------- ---------------- ----------------
## County                       Yes              Yes              Yes
## Year                          No               No               No
## _______________ ________________ ________________ ________________
## S.E. type             by: County       by: County       by: County
## Observations                 630              630              630
## R2                       0.33114          0.33114          0.43784
## Within R2                0.04762          0.04762          0.19955
##
##                   crime_many_es..7 crime_many_es..8 crime_many_es..9
## Sample (urban)         Full sample      Full sample      Full sample
## Dependent Var.: Prob. Conviction Prob. Conviction Prob. Conviction
##
## Constant
## Prob. Arrest     0.3665 (0.3999)  0.3571 (0.4001)  -1.008 (0.7845)
## Avg. Sentence                     0.0138 (0.0358)  0.0074 (0.0274)
## polpc                                               294.5. (125.8)
## Fixed-Effects:  ---------------- ---------------- ----------------
## County                        No               No               No
## Year                         Yes              Yes              Yes
## _______________ ________________ ________________ ________________
## S.E. type               by: Year         by: Year         by: Year
## Observations                 630              630              630
## R2                       0.01355          0.01398          0.22043
## Within R2                0.00139          0.00182          0.21082
##
##                   crime_many_e..10 crime_many_e..11 crime_many_e..12
## Sample (urban)         Full sample      Full sample      Full sample
## Dependent Var.: Prob. Conviction Prob. Conviction Prob. Conviction
##
## Constant
## Prob. Arrest      -2.939 (2.077)   -2.931 (2.079)   -3.388 (2.552)
## Avg. Sentence                     0.0104 (0.0277)  0.0361 (0.0269)
## polpc                                               324.6* (139.9)
## Fixed-Effects:  ---------------- ---------------- ----------------
## County                       Yes              Yes              Yes
## Year                         Yes              Yes              Yes
## _______________ ________________ ________________ ________________
## S.E. type             by: County       by: County       by: County
## Observations                 630              630              630
## R2                       0.34289          0.34305          0.44589
```

```
## Within R2                0.04784         0.04807         0.19709
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
etable(crime_many_estimations[lhs='crmrte',sample=2],title='Crime Rate in Urban Areas',notes='Note: Est
```

```
##                   crime_many_estim..1 crime_many_estim..2 crime_many_estim..3
## Sample (urban)                      0                   0                   0
## Dependent Var.:            Crime Rate          Crime Rate          Crime Rate
##
## Constant          0.0370*** (0.0012)  0.0392*** (0.0023)  0.0385*** (0.0021)
## Prob. Arrest     -0.0270*** (0.0034) -0.0267*** (0.0034) -0.0356*** (0.0033)
## Avg. Sentence                           -0.0003 (0.0002)    -0.0003 (0.0002)
## polpc                                                        1.819*** (0.2036)
## Fixed-Effects:   ------------------- ------------------- -------------------
## County                             No                  No                  No
## Year                               No                  No                  No
## ________________ ___________________ ___________________ ___________________
## S.E. type                         IID                 IID                 IID
## Observations                      574                 574                 574
## R2                            0.10023             0.10240             0.21267
## Within R2                          --                  --                  --
##
##                   crime_many_es..4 crime_many_es..5 crime_many_est..6
## Sample (urban)                   0                0                 0
## Dependent Var.:         Crime Rate       Crime Rate        Crime Rate
##
## Constant
## Prob. Arrest     -0.0017 (0.0026) -0.0017 (0.0026)  -0.0041 (0.0027)
## Avg. Sentence                     1.95e-5 (0.0001)  0.0002. (0.0001)
## polpc                                               1.722*** (0.3264)
## Fixed-Effects:   ---------------- ---------------- -----------------
## County                        Yes              Yes               Yes
## Year                           No               No                No
## ________________ ________________ ________________ _________________
## S.E. type              by: County       by: County        by: County
## Observations                  574              574               574
## R2                        0.80689          0.80690           0.84780
## Within R2                 0.00084          0.00088           0.21251
##
##                   crime_many_esti..7 crime_many_esti..8 crime_many_esti..9
## Sample (urban)                     0                  0                  0
## Dependent Var.:           Crime Rate         Crime Rate         Crime Rate
##
## Constant
## Prob. Arrest     -0.0268** (0.0058) -0.0264** (0.0059) -0.0355** (0.0064)
## Avg. Sentence                        -0.0004* (0.0001)  -0.0004. (0.0002)
## polpc                                                    1.865* (0.7238)
## Fixed-Effects:   ------------------ ------------------ ------------------
## County                          No                 No                 No
## Year                           Yes                Yes                Yes
## ________________ __________________ __________________ __________________
## S.E. type                  by: Year           by: Year           by: Year
## Observations                    574                574                574
## R2                          0.10602            0.10988            0.22501
```

```
## Within R2                  0.09934           0.10323           0.21922
##
##                  crime_many_e..10 crime_many_e..11 crime_many_es..12
## Sample (urban)                  0                0                 0
## Dependent Var.:        Crime Rate       Crime Rate        Crime Rate
##
## Constant
## Prob. Arrest    -0.0010 (0.0026) -0.0011 (0.0026)  -0.0036 (0.0026)
## Avg. Sentence                    -0.0001 (0.0001)  3.91e-5 (0.0001)
## polpc                                             1.805*** (0.3295)
## Fixed-Effects:  ---------------- ---------------- -----------------
## County                       Yes              Yes               Yes
## Year                         Yes              Yes               Yes
## _______________ ________________ ________________ _________________
## S.E. type              by: County       by: County        by: County
## Observations                 574              574               574
## R2                       0.81420          0.81446           0.85886
## Within R2                0.00030          0.00168           0.24061
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
etable(crime_many_estimations[lhs='crmrte',sample=3],title='Crime Rate in Rural Areas',notes='Note: Est
```

```
##                  crime_many_estim..1 crime_many_estim..2 crime_many_estim..3
## Sample (urban)                     1                   1                   1
## Dependent Var.:           Crime Rate          Crime Rate          Crime Rate
##
## Constant          0.1055*** (0.0078)  0.0990*** (0.0100)  0.0775*** (0.0138)
## Prob. Arrest    -0.1995*** (0.0368) -0.2033*** (0.0369) -0.2014*** (0.0357)
## Avg. Sentence                           0.0007 (0.0007)     0.0004 (0.0007)
## polpc                                                         11.94* (5.472)
## Fixed-Effects:  ------------------- ------------------- -------------------
## County                            No                  No                  No
## Year                              No                  No                  No
## _______________ ___________________ ___________________ ___________________
## S.E. type                        IID                 IID                 IID
## Observations                      56                  56                  56
## R2                           0.35269             0.36588             0.41909
## Within R2                         --                  --                  --
##
##                  crime_many_est..4 crime_many_est..5 crime_many_est..6
## Sample (urban)                   1                 1                 1
## Dependent Var.:         Crime Rate        Crime Rate        Crime Rate
##
## Constant
## Prob. Arrest    -0.1903. (0.0823) -0.1871. (0.0801) -0.1811* (0.0741)
## Avg. Sentence                        0.0008 (0.0005)   0.0007 (0.0005)
## polpc                                                  8.647 (6.132)
## Fixed-Effects:  ----------------- ----------------- -----------------
## County                        Yes               Yes               Yes
## Year                           No                No                No
## _______________ _________________ _________________ _________________
## S.E. type              by: County        by: County        by: County
## Observations                   56                56                56
## R2                        0.88722           0.89616           0.90231
```

```
## Within R2                  0.20282            0.26602            0.30946
##
##                   crime_many_estim..7 crime_many_estim..8 crime_many_estim..9
## Sample (urban)                      1                   1                   1
## Dependent Var.:            Crime Rate          Crime Rate          Crime Rate
##
## Constant
## Prob. Arrest    -0.1979*** (0.0149) -0.2011*** (0.0175) -0.1982*** (0.0137)
## Avg. Sentence                           0.0005 (0.0007)   -4.93e-5 (0.0007)
## polpc                                                       13.42** (3.493)
## Fixed-Effects:  ------------------- ------------------- -------------------
## County                             No                  No                  No
## Year                              Yes                 Yes                 Yes
## _______________ ___________________ ___________________ ___________________
## S.E. type                    by: Year            by: Year            by: Year
## Observations                       56                  56                  56
## R2                            0.39994             0.40426             0.46292
## Within R2                     0.36403             0.36861             0.43078
##
##                   crime_many_es..10 crime_many_es..11 crime_many_es..12
## Sample (urban)                    1                 1                 1
## Dependent Var.:          Crime Rate        Crime Rate        Crime Rate
##
## Constant
## Prob. Arrest    -0.1694. (0.0771) -0.1723. (0.0733) -0.1709* (0.0703)
## Avg. Sentence                       0.0002 (0.0005)  3.84e-5 (0.0006)
## polpc                                                    11.27 (6.369)
## Fixed-Effects:  ----------------- ----------------- -----------------
## County                          Yes               Yes               Yes
## Year                            Yes               Yes               Yes
## _______________ _________________ _________________ _________________
## S.E. type                by: County        by: County        by: County
## Observations                     56                56                56
## R2                          0.93501           0.93554           0.94120
## Within R2                   0.23564           0.24188           0.30840
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Concept check In our second table, the probability of conviction regressed on probability of arrest is almost certainly not causal. It is a pretty bogus regresion since both that are heavily affected by government decisions.

Can we say any of the above are causal? What would we need to assume?

Difference-in-differences

One of the most popular uses of fixed effects is to implement difference-in-difference designs we've discussed. Here's a quick visualization. Let's walk through an example that uses the National Supported Work Demonstration dataset that Lalonde (1986) published on.[4]

Lalonde (1986)

The neat thing about these data is Lalonde (and a follow-up by Dehejia and Wahba (2022)) compare experimental to non-experimental data. The experimental data is from a randomized control trial (RCT) of a job training program. The non-experimental data is a random sample of US households.

[4]I take this example from an activity devised by Scott Cunningham and Kyle Butts.

Earned Income Tax Credit

The Earned Income Tax Credit (EITC) was increased for parents in 1993. The EITC is a tax credit for low-income workers. It is a refundable tax credit, meaning that if the credit exceeds the amount of taxes owed, the excess is returned to the taxpayer. The EITC is designed to supplement wages for low-to-moderate income workers. The amount of the credit depends on income and number of children.

The EITC is also designed to incentivize work. It initially increases as earnings increase, before leveling off and falling once earnings reach a threshold level and the worker transitions out of "low-income."

Effectively at low-income levels, the EITC increases the dollars earned from working – either on the intensive margin (one more hour) or extensive margin (working vs. not working). But does it effect labor supply?

Let's focus on how this affects labor supply of single mothers who are the primary beneficiaries of . This example is borrowed from Nick Huntington-Klein and pulled from work by Bruce Meyer (2002).

We walked through this example in the lecture, but let's do it again.

Diff-in-diff with data

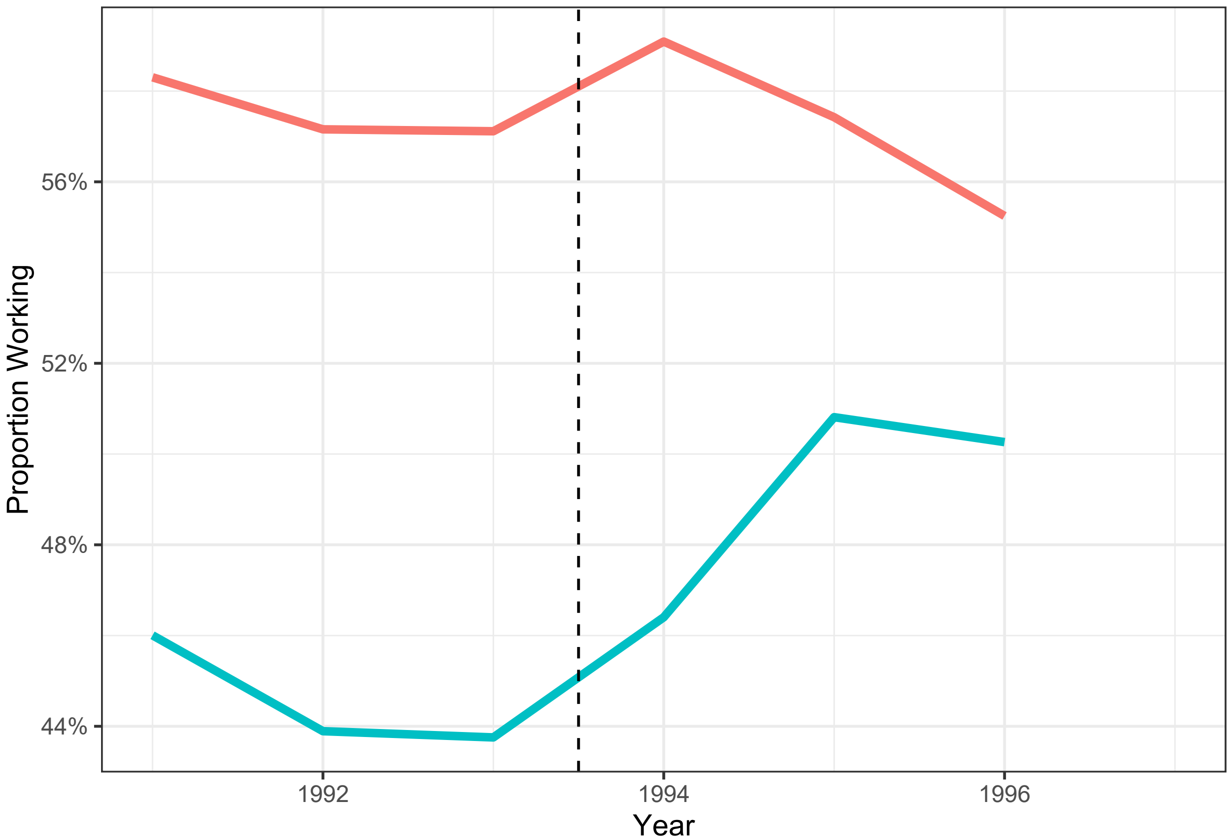
Let's load in the data.

We do not have an individual identifier in these data, so we can't add an individual fixed effect. We can add other fixed effects if we believe there is endogenous variation in the treatment between the groups of the fixed effect.

Still, let's work through how to visualize the data to check for no pre-trends and treatment effects change over time. We checked averages for our two groups before – not bad!

```
## `summarise()` has grouped output by 'year', 'treated'. You can override using
## the `.groups` argument.
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



But the lines are a little far apart, so it makes it tricky to visualize the difference. And we don't know the confident interval on the difference between these. Let's try to get that!

Introducing the `i()` function. This handy little guy is a function that creates an interaction term. It's a little tricky to use, but it's worth it. Basically, what you do is you feed it a factor variable, an interacted variable, then a reference value of the factor variable – all coefficients will relative to the level when the factor variable equals the reference value.

```
## OLS estimation, Dep. Var.: work
## Observations: 13,746
## Fixed-effects: year: 6,  treated: 2
## Standard-errors: Heteroskedasticity-robust
##                     Estimate Std. Error  t value  Pr(>|t|)
## year::1991:treated 0.010618   0.028518 0.372334 0.7096501
## year::1992:treated 0.000952   0.028960 0.032880 0.9737710
## year::1994:treated 0.006720   0.029484 0.227919 0.8197125
## year::1995:treated 0.067488   0.030154 2.238086 0.0252314 *
## year::1996:treated 0.083754   0.030552 2.741312 0.0061274 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## RMSE: 0.496477      Adj. R2: 0.012584
##                   Within R2: 0.001075
```

So what does this output mean? Well it tells us the difference between the treated and untreated groups over time! But relative to when? It is all relative to the reference value, when year=1993. That is often called the "omitted" year. I chose the period just before the EITC expansion.

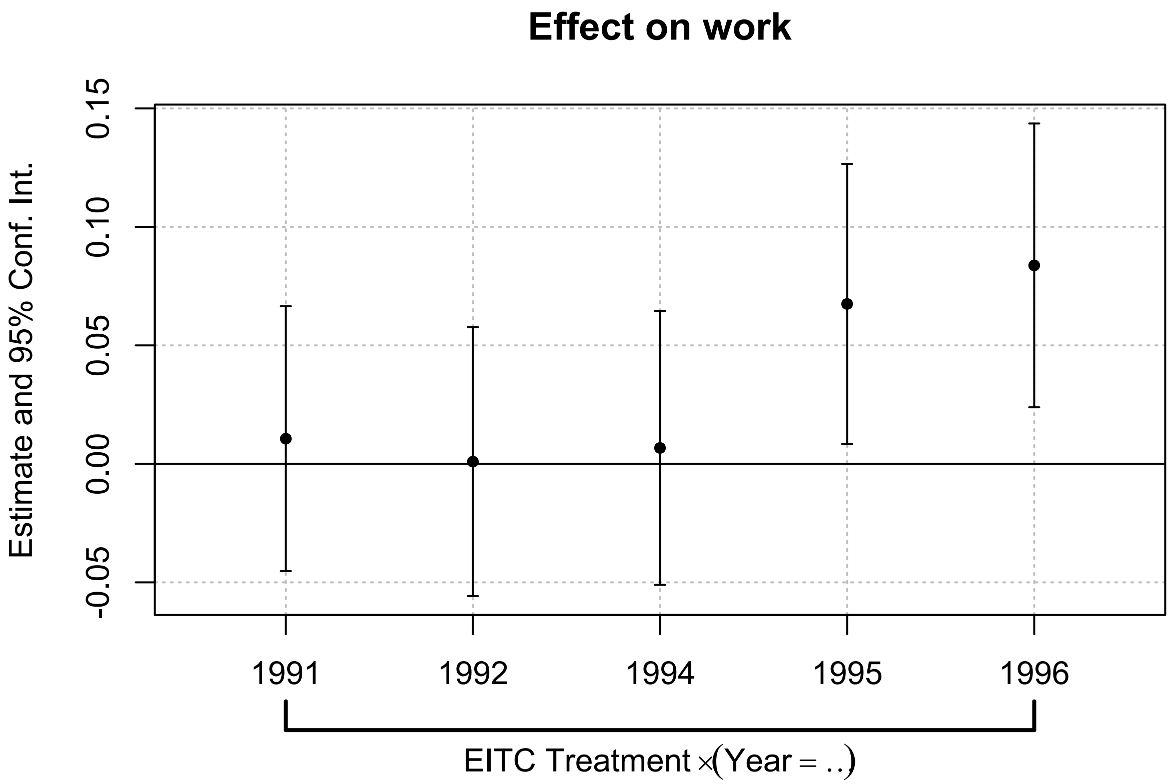
Challenge: What regression did we just run? Write it out. We have a year fixed effect and a treated fixed effect. Note the treated fixed effect is defined across individuals because we do not have an individual identifier!

But how do we visualize this? We have a few options. They both work the same way as the examples with `coefplot()`

and ggplot() above though. Note, I introduce a dict to improve the labels.

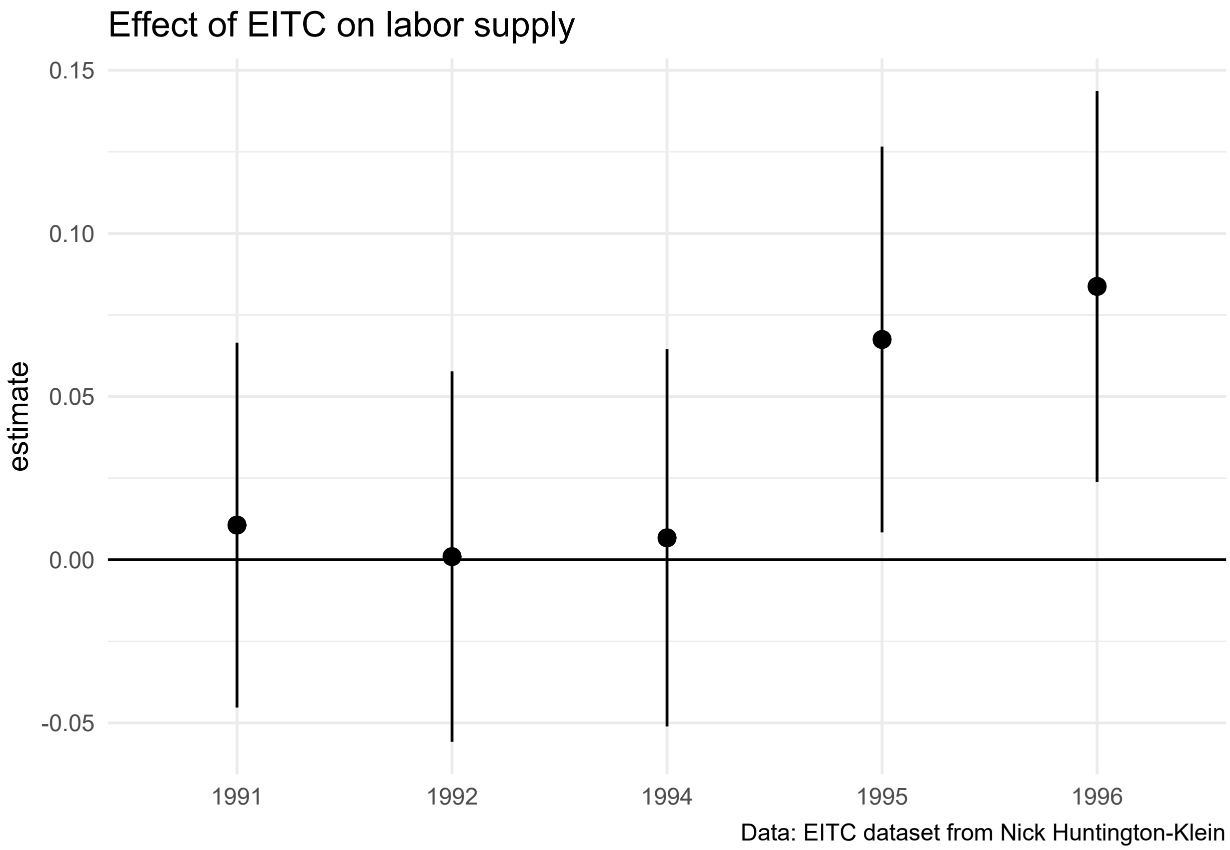
The plots show that prior to 1994, the labor supply decisions of women with and without children were on a similar trend (though it is a fairly short trend).

```
coefplot(eitc_did,dict=c('treated'='EITC Treatment','year'='Year'))
```



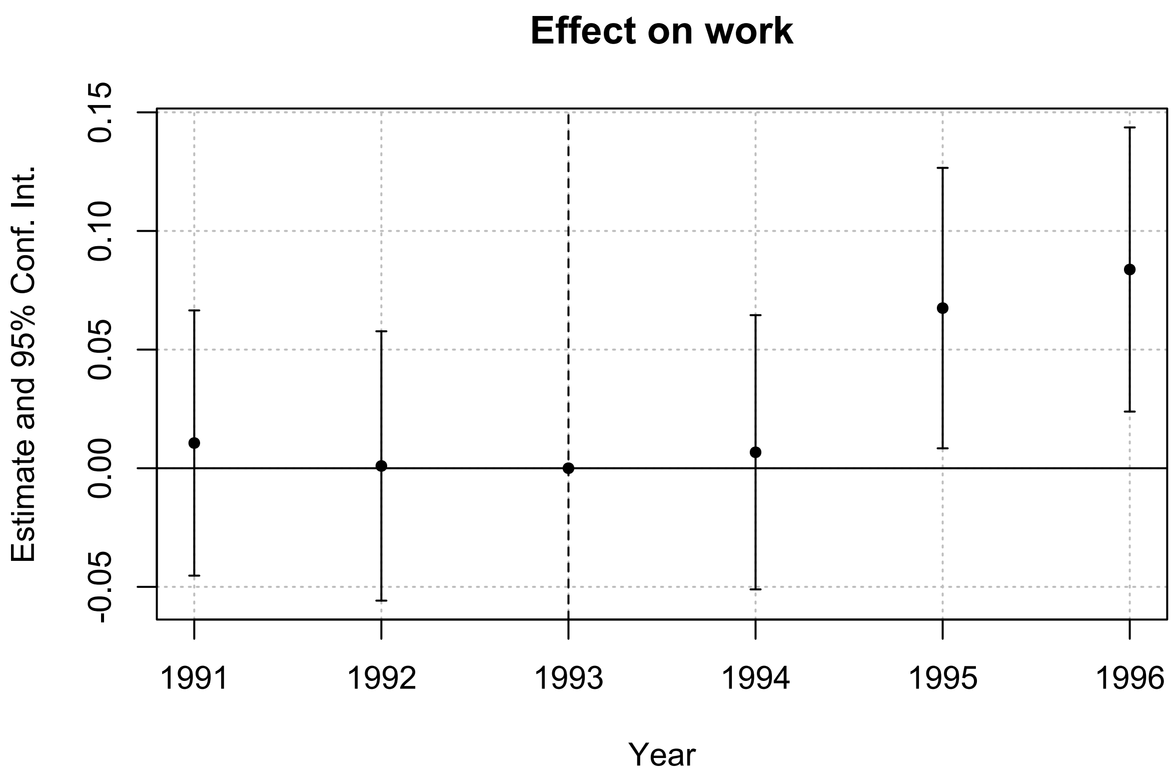

```
coef_eitc_did <- tidy(eitc_did, conf.int = TRUE) %>%
  mutate(year=str_extract(term,'\\d{4}')) # Regular expressions to extract year

ggplot(coef_eitc_did, aes(x=year, y=estimate, ymin=conf.low, ymax=conf.high)) +
  geom_pointrange() +
  geom_hline(yintercept = 0, col = "black") +
  labs(
    title = "Effect of EITC on labor supply",
    caption = "Data: EITC dataset from Nick Huntington-Klein"
    ) +
  theme(axis.title.x = element_blank())
```



And then we also have `iplot()`, which works directly with `i()`. It works well for quick visualization, but it can be a little clunky to make as beautiful plots as you can with ggplot.

```
iplot(eitc_did,dict=c('treated'='EITC Treatment','year'='Year'))
```



"'

Further resources

- Ed Rubin has outstanding teaching notes for econometrics with R on his website. This includes both undergrad- and graduate-level courses. Seriously, check them out.
- Several introductory texts are freely available, including *Introduction to Econometrics with R* (Christoph Hanck *et al.*), *Using R for Introductory Econometrics* (Florian Heiss), and *Modern Dive* (Chester Ismay and Albert Kim).
- Tyler Ransom has a nice cheat sheet for common regression tasks and specifications.
- Itamar Caspi has written a neat unofficial appendix to this lecture, *recipes for Dummies*. The title might be a little inscrutable if you haven't heard of the `recipes` package before, but basically it handles "tidy" data preprocessing, which is an especially important topic for machine learning methods. We'll get to that later in course, but check out Itamar's post for a good introduction.
- I promised to provide some links to time series analysis. The good news is that R's support for time series is very, very good. The Time Series Analysis task view on CRAN offers an excellent overview of available packages and their functionality.
- Lastly, for more on visualizing regression output, I highly encourage you to look over Chapter 6 of Kieran Healy's *Data Visualization: A Practical Guide*. Not only will learn how to produce beautiful and effective model visualizations, but you'll also pick up a variety of technical tips.