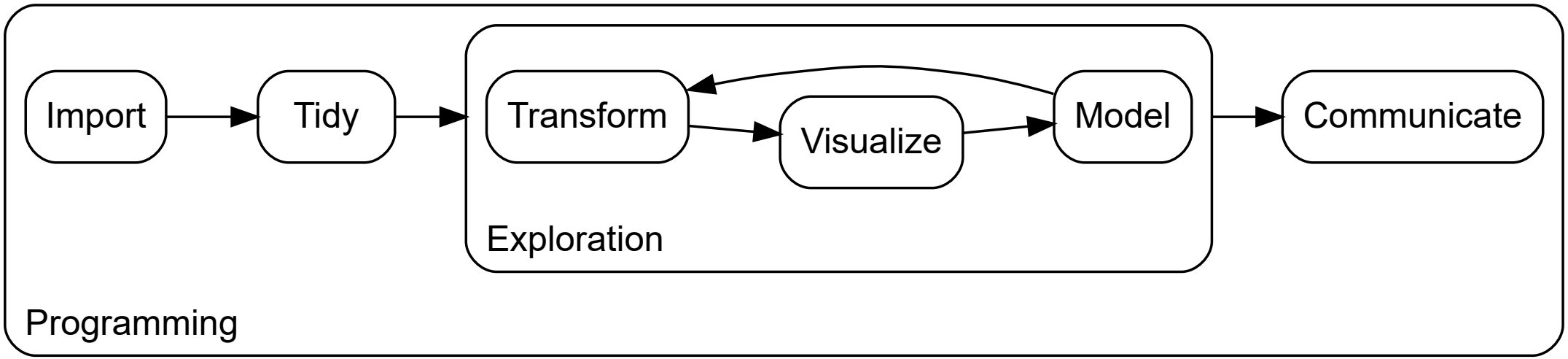
Data Analysis using R

Modeling

Sven Werenbeck-Ueding

09.12.2024



Source: Wickham and Grolemund (2016)

Fenced Out:

The Impact of Border Construction on U.S.-Mexico Migration

> "This paper estimates the **impact of the US-Mexico border fence on US-Mexico migration** by **exploiting variation in the timing and location of US government investment in fence construction**. Using Mexican survey data and data I collected on fence construction, I find that construction in a municipality reduces migration by 27 percent for municipality residents and 15 percent for residents of adjacent municipalities."
>
> Feigenberg (2020a)

Data

Mexican Survey

- Encuesta Nacional de Ocupación y Empleo (ENOE) from Q3 2003 to Q3 2013
- Quarterly rotating panel: Households included for 5 quarters
- Records whether any household member leaves to the US
- Potential migrants are restricted from ages 15 to 65
- Explanatory variables: age, gender, marital status and education for all household members

Fence Construction

- Data collected by identifying potential fence locations:
  - Documents from US authorities: Customs and Border Protection (CPB) and Government Accountability Office (GAO)
  - Local government reports
  - Published contracts with construction firms
- Cross-checked with an environmental organization tracking the impact of fence construction (Sierra Club)
- Provides quarterly information on fence construction on Mexican municipality level

Identification Strategy

- By exploiting temporal and spatial variation in fence construction, Feigenberg (2020a) estimates the impact of fence construction on the migration decision of potential migrants
- Exogenous variation in fence construction allows for a difference-in-difference estimator of the form

$$Pr(Y_{mti} = 1 \mid z_{mti}) = \frac{exp(\alpha + X_{mti}\beta + \delta \times fence_{mti} + \gamma_m + \tau_t)}{1 + exp(\alpha + X_{mti}\beta + \delta \times fence_{mti} + \gamma_m + \tau_t)}$$

  - $Y_{mti} \in [0, 1]$: 1 if individual i living in municipality m migrates to the US in year-quarter t
  - X_{mti}: Socio-economic characteristics of individual i living in municipality m and year-quarter t
  - $fence_{mti} \in [0, 1]$: 1 if fence construction started in municipality m in or before year-quarter t
  - γ_m, τ_t: Municipality and year-quarter fixed effects

Results

- Table 2 shows log-odds coefficients of different specifications for the estimation strategy
- Column (2) of panel A employs the specification on the previous slide
- Fence construction reduces probability of migrating by $1 - e^{\delta} = 37.87\%$
- Relative to baseline migration rate of 4.2 per $1,000$ respondents
- Parentheses show standard errors clustered by municipality
- Effect is highly significant: p-value is ≈ 0

TABLE 2—IMPACT OF FENCE CONSTRUCTION ON BORDER MUNICIPALITY MIGRATION

	Migrate to United States		
	(1)	(2)	(3)
Panel A			
Fence construction	−0.319	−0.476	−0.447
	(0.129)	(0.132)	(0.192)
Panel B			
Fence construction	−0.283	−0.398	−0.548
	(0.136)	(0.158)	(0.168)
Number of adjacent municipalities fenced	−0.164	−0.181	−0.318
	(0.0891)	(0.0909)	(0.110)
Number of fenced municipalities two away	0.0389	−0.0490	−0.0216
	(0.0708)	(0.0933)	(0.135)
Panel C			
Fence construction	−0.319	−0.438	−0.665
	(0.178)	(0.212)	(0.242)
Number of adjacent municipalities fenced	−0.192	−0.211	−0.401
	(0.120)	(0.129)	(0.193)
Number of fenced municipalities two away	0.0440	−0.0443	0.0164
	(0.0752)	(0.0961)	(0.153)
Fence construction × number of adjacent municipalities fenced	0.0520	0.0566	0.167
	(0.118)	(0.132)	(0.171)
Observations	330,503	316,591	316,591
Municipality fixed effects	X	X	X
Year-quarter fixed effects	X	X	X
Additional controls		X	X
State × year-quarter fixed effects			X
Mean of non-fenced		0.00420	
		[0.0647]	

Source: Feigenberg (2020a)

Prerequisites

```
library(tidyverse)

# Import the data set and convert municipality and period identifiers to
# factor variables (both have to be converted to character first). Set the
# first period (Q3 2003, i.e. 0.5) as the reference category for the period
# identifier
df_mig <- read_csv("data/processed/fence_migration.csv") %>%
  mutate(
    municipality = fct(as.character(municipality)),
    period = fct_relevel(as.character(period), "0.5")
  )
```

- For this course, a random 50% sample of the full sample was drawn from the data set provided by Feigenberg (2020b)
- Our results may differ

Regressions in R

Fit Linear Regressions in R

```
?lm
```

The `lm()` function fits linear models, including multivariate models. It can also be used to carry out single stratum analysis of variance and analysis of covariance.

- `formula`: An object of class `formula` that symbollically describes the model to be fitted
- `data`: An object of class `data.frame` (or coercible by `as.data.frame`) containing the model variables
- `subset`: An optional vector for subsetting observations in `data`
- `weights`: An optional numeric vector of weights, e. g. for weighted least squares

Formulas in R

Expressions such as `y ~ x1 + x2 + x3` use the `~` operator to specify that response `y` is modeled by a set of predictors (`x1`, `x2` and `x3`)

Operator	Meaning	Example
`:`	Interaction effect between two predictors	`x1:x2`
`*`	Main and interaction effects of predictors	`x1*x2` → `x1 + x2 + x1:x2`
`^`	Expands to a formula containing main effects and interactions up to the n^{th} order	`(x1 + x2 + x3)^2` → `x1 + x2 + x3 + x1:x2 + x1:x3 + x2:x3`
`/`	Terms on the LHS are nested within those on the right	`x1/x2` → `x1 + x1:x2`
`-`	Removes terms from the formula (e. g. the intercept)	`y ~ -1 + x1`
`.`	Usually interpreted as all `data` columns not otherwise in the formula	`y ~ .`

Binary Choice Models

Model	Functional Form	Command
LPM	$Pr(Y_i = 1 \mid X_i) = X_i\beta$	`lm(y ~ .)`
Logit	$Pr(Y_i = 1 \mid X_i) = \wedge(X_i\beta) = \frac{e^{X_i\beta}}{1 + e^{X_i\beta}}$	`glm(y ~ ., family = "binomial")`
Probit	$Pr(Y_i = 1 \mid X_i) = \phi(X_i\beta) = \int_{-\infty}^{X_i\beta} \phi(z)dz$	`glm(y ~ ., binomial(link = "probit"))`

Fitting a Linear Probability Model in base R

Task | Code | Summary | Summary (tidy)

Estimate the impact of border fence construction on Mexican-US migration using a linear probability model akin to the logit model employed by Feigenberg (2020a). Use the data provided in `data/processed/fence_migration.csv`.

$$Y_{mti} = \alpha + X_{mti}\beta + \delta \times fence_{mti} + \gamma_m + \tau_t$$

- $Y_{mti} \in [0, 1]$: 1 if individual i living in municipality m migrates to the US in year-quarter t
- X_{mti}: Socio-economic characteristics of individual i living in municipality m and year-quarter t
- $fence_{mti} \in [0, 1]$: 1 if fence construction started in municipality m in or before year-quarter t
- γ_m, τ_t: Municipality and year-quarter fixed effects

Fitting a Linear Probability Model in base R

Task | Code | Summary | Summary (tidy)

```
# Define the regression formula
full_formula <- formula(
  migrate ~ fence + female + age + educ + married +
    municipality + period
)

# Fit a linear probability model according to the formula above and set the data source to
# the `df_mig` data set
lp_model <- lm(full_formula, data = df_mig)
```

Fitting a Linear Probability Model in base R

Task | Code | **Summary** | Summary (tidy)

```
summary(lp_model)
```

```
##
## Call:
## lm(formula = full_formula, data = df_mig)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.04002 -0.00498 -0.00320 -0.00150  1.00246
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)         9.056e-03  1.257e-03   7.205 5.84e-13 ***
## fence              -1.785e-03  7.872e-04  -2.267  0.02338 *
## female             -1.567e-03  2.908e-04  -5.389 7.08e-08 ***
## age                -5.338e-05  1.184e-05  -4.509 6.51e-06 ***
```

Fitting a Linear Probability Model in base R

Task | Code | Summary | **Summary (tidy)**

```
# Print a tidy model summary to the console (remove FE and model fit from the table)
broom::tidy(lp_model) %>%
  filter(term %in% c("fence", "female", "age", "educ", "married"))
```

```
## # A tibble: 5 × 5
##   term      estimate std.error statistic  p.value
##   <chr>        <dbl>     <dbl>     <dbl>    <dbl>
## 1 fence   -0.00178   0.000787     -2.27  2.34e- 2
## 2 female  -0.00157   0.000291     -5.39  7.08e- 8
## 3 age     -0.0000534 0.0000118    -4.51  6.51e- 6
## 4 educ     0.0000278 0.0000366     0.760 4.47e- 1
## 5 married -0.00214   0.000319     -6.73  1.66e-11
```

The `lm` Class

- Models created through base R's `lm()` function are stored in list objects with the class `lm`
- An object of class `lm` stores several model details such as
  - `coefficients`: Named coefficient vector
  - `fitted.values`: Vector of predicted values from the regression (i.e. $\hat{Y}_{mti}$)
  - `residuals`: Vector of residuals from the regression (i.e. $Y_{mti} - \hat{Y}_{mti}$)
- Several generic functions, such as `summary()` and `print()`, work with the `lm` Class, e.g. the `tidy()` function from the `broom` package takes an `lm` object, extracts the coefficients and t-tests, and returns a tidy data set (see previous slide)

```
class(lp_model)
```

```
## [1] "lm"
```

```
# Alternatively, use the `coef()` function
lp_model$coefficients[1:3]
```

```
##  (Intercept)        fence       female
##  0.009055521 -0.001784673 -0.001567282
```

Adjusting Standard Errors

- In many cases, errors are heterogeneous, i.e. the variance of the error term is nonconstant
  - Estimates are unbiased but inefficient, leading to incorrect p-values
  - Can lead to rejecting a true / failing to reject a false null hypothesis
- Can be checked using a Breusch-Pagan test, e.g. via the `bgtest` function from the `lmtest`
- Often a visual inspection is sufficient to suspect a violation of the homoscedasticity assumption
- The `sandwich` package offers an easy way to create heteroscedasticity-consistent (robust) standard errors
- Using the `coeftest` function from the `lmtest`, we can adjust our t-tests for robust standard errors

Code | Output

```
library(lmtest)
library(sandwich)

# Specify covariance matrix computed by the
# sandwich package using the vcovHC() function
# and specifying "HC3" as the type of robust
# standard error (i.e. efficient covariance
# matrix estimator even in small samples)
lp_coeftest <- coeftest(
  lp_model,
  vcov. = vcovHC(lp_model, type = "HC3")
)

broom::tidy(lp_coeftest)
```

Adjusting Standard Errors

- In many cases, errors are heterogeneous, i.e. the variance of the error term is nonconstant
  - Estimates are unbiased but inefficient, leading to incorrect p-values
  - Can lead to rejecting a true / failing to reject a false null hypothesis
- Can be checked using a Breusch-Pagan test, e.g. via the `bgtest` function from the `lmtest`
- Often a visual inspection is sufficient to suspect a violation of the homoscedasticity assumption
- The `sandwich` package offers an easy way to create heteroscedasticity-consistent (robust) standard errors
- Using the `coeftest` function from the `lmtest`, we can adjust our t-tests for robust standard errors

Code | Output

```
## Lade nötiges Paket: zoo

##
## Attache Paket: 'zoo'

## Die folgenden Objekte sind maskiert von 'pac
##
##     as.Date, as.Date.numeric

## # A tibble: 68 × 5
##    term                estimate std.error sta
##    <chr>                  <dbl>     <dbl>
##  1 (Intercept)         0.00906   0.00152
##  2 fence              -0.00178   0.000888
##  3 female             -0.00157   0.000290
##  4 age                -0.0000534 0.0000110
```

Cluster-Robust Standard Errors

- A particular type of homoscedasticity violation is usually found when the data is structured hierarchically
- Observations may be correlated within groups (clusters) but independent across groups: Violates the assumption that errors are independent
- In Feigenberg (2020a): One-way clustering standard errors on municipality level
- Note that the code on the right uses the robust standard error `type` "HC1" as a default

Code | Output

```
library(lmtest)
library(sandwich)

# Use clustered standard errors; clustering
# variable can be provided in a formula
lp_coeftest_CL <- coeftest(
  lp_model,
  vcov. = vcovCL(
    lp_model,
    cluster = ~ municipality
  )
)

broom::tidy(lp_coeftest_CL)
```

Cluster-Robust Standard Errors

- A particular type of homoscedasticity violation is usually found when the data is structured hierarchically
- Observations may be correlated within groups (clusters) but independent across groups: Violates the assumption that errors are independent
- In Feigenberg (2020a): One-way clustering standard errors on municipality level
- Note that the code on the right uses the robust standard error `type` "HC1" as a default

Code | Output

```
## # A tibble: 68 × 5
##    term                estimate std.error sta
##    <chr>                  <dbl>     <dbl>
##  1 (Intercept)          0.00906   0.00275
##  2 fence               -0.00178   0.000921
##  3 female              -0.00157   0.000438
##  4 age                 -0.0000534 0.0000132
##  5 educ                 0.0000278 0.0000687
##  6 married             -0.00214   0.000529
##  7 municipality2002    -0.00196   0.0000786   -
##  8 municipality2003    -0.000351  0.0000392
##  9 municipality5025     0.000940  0.0000967
## 10 municipality5002    -0.000948  0.000159
## # ℹ 58 more rows
```

Fitting a Logit Model in base R

Task | Code | Summary | Summary (tidy)

Replicate the results of Feigenberg (2020a) using their logit model. Implement the model in base R.

$$Pr(Y_{mti} = 1 \mid z_{mti}) = \wedge(\alpha + X_{mti}\beta + \delta \times fence_{mti} + \gamma_m + \tau_t)$$

- $Y_{mti} \in [0, 1]$: 1 if individual i living in municipality m migrates to the US in year-quarter t
- X_{mti}: Socio-economic characteristics of individual i living in municipality m and year-quarter t
- $fence_{mti} \in [0, 1]$: 1 if fence construction started in municipality m in or before year-quarter t
- γ_m, τ_t: Municipality and year-quarter fixed effects

Fitting a Logit Model in base R

Task | Code | Summary | Summary (tidy)

```
# Fit a logit model as given by formula on the df_mig data set
logit_model <- glm(full_formula, data = df_mig, family = "binomial")
```

Fitting a Logit Model in base R

Task | Code | **Summary** | Summary (tidy)

```
summary(logit_model)
```

```
##
## Call:
## glm(formula = full_formula, family = "binomial", data = df_mig)
##
## Coefficients:
##                      Estimate Std. Error z value Pr(>|z|)
## (Intercept)        -4.282496   0.312529 -13.703  < 2e-16 ***
## fence              -0.470669   0.206658  -2.278  0.02275 *
## female             -0.472763   0.091037  -5.193 2.07e-07 ***
## age                -0.016581   0.003900  -4.251 2.13e-05 ***
## educ                0.011944   0.011479   1.041  0.29810
## married            -0.626415   0.101308  -6.183 6.28e-10 ***
## municipality2002   -0.832191   0.151951  -5.477 4.33e-08 ***
## municipality2003   -0.107081   0.267432  -0.400  0.68886
```

Fitting a Logit Model in base R

Task | Code | Summary | **Summary (tidy)**

```
# Print a tidy model summary to the console
broom::tidy(logit_model) %>%
  filter(term %in% c("fence", "female", "age", "educ", "married"))
```

```
## # A tibble: 5 × 5
##   term    estimate std.error statistic  p.value
##   <chr>      <dbl>     <dbl>     <dbl>    <dbl>
## 1 fence    -0.471    0.207       -2.28 2.28e- 2
## 2 female   -0.473    0.0910      -5.19 2.07e- 7
## 3 age      -0.0166   0.00390     -4.25 2.13e- 5
## 4 educ      0.0119   0.0115       1.04 2.98e- 1
## 5 married  -0.626    0.101       -6.18 6.28e-10
```

Computing Marginal Effects

- Coefficients from a Logit model are not directly interpretable: A one-unit change in the covariate corresponds to a change in *log odds*
- Odds ratios provide a more intuitive measure of the relative impact of covariates on the outcome, i.e.

$$\text{Odds Ratio} = e^{\beta}$$

  - $\text{Odds Ratio} > 1$: Odds for $P(Y = 1)$ increase with a one-unit increase in X
  - $\text{Odds Ratio} < 1$: Odds for $P(Y = 1)$ decrease with a one-unit increase in X

Computing Marginal Effects

- Unlike odds ratios, marginal effects directly tell the change in probability a one-unit change in the covariate has on the outcome, i.e.
  $\text{Marginal Effect} = \frac{\partial P(Y=1)}{\partial X}$
- We can use this measure for understanding the magnitude of impact on the outcome in terms of probabilities
- Due to the non-linearity in the outcome, it is important to correctly choose *where* to evaluate the marginal effect: For a central tendency, choose the sample mean, i.e. compute the **average marginal effect (AME)**
- The `marginaleffects` package offers several functions to help you with computing marginal effects; uses the `sandwich` package for robust covariance matrix estimation

Code | Output

```
library(marginaleffects)

# avg_slopes returns the AME
logit_me <- avg_slopes(logit_model)

broom::tidy(logit_me)
```

Computing Marginal Effects

- Unlike odds ratios, marginal effects directly tell the change in probability a one-unit change in the covariate has on the outcome, i.e.
  $\text{Marginal Effect} = \frac{\partial P(Y=1)}{\partial X}$
- We can use this measure for understanding the magnitude of impact on the outcome in terms of probabilities
- Due to the non-linearity in the outcome, it is important to correctly choose *where* to evaluate the marginal effect: For a central tendency, choose the sample mean, i.e. compute the **average marginal effect (AME)**
- The `marginaleffects` package offers several functions to help you with computing marginal effects; uses the `sandwich` package for robust covariance matrix estimation

Code | Output

```
## # A tibble: 67 × 3
##    term         contrast       estimate
##    <chr>        <chr>             <dbl>
##  1 age          dY/dX         -0.0000546
##  2 educ         dY/dX          0.0000393
##  3 female       1 - 0         -0.00153
##  4 fence        1 - 0         -0.00156
##  5 married      1 - 0         -0.00204
##  6 municipality 2002 - 2004   -0.00204
##  7 municipality 2003 - 2004   -0.000366
##  8 municipality 26002 - 2004   0.00326
##  9 municipality 26017 - 2004 -0.00246
## 10 municipality 26043 - 2004 -0.000595
## # ℹ 57 more rows
```

CRAN Task View: Econometrics

- Base R already provides lots of functions for (applied) econometrics, especially the `stats` package but
  - Many "advanced" estimators, however, are not readily available in base R
  - Some base R functions show poor performance when applied on large data sets
- The "CRAN Task View: Econometrics" (Zeilis, McDermott, and Tappe, 2024) provides a comprehensive overview of very useful packages that complement your analysis, e.g.
  - `fixest`: Fast implementation of high-dimensional fixed effects
  - `lmtest`: Tests, data sets, and examples for diagnostic checking in linear regression models
  - `sandwich`: Model-robust covariance matrix estimators for easy imputation in other packages
  - `modelsummary`: Customizable tables, plots etc. for several statistical models to summarize your results

➡ CRAN Task View: Econometrics

Speeding Up Computation

Fixed Effects Estimation

- Many microeconometric applications suffer from unobserved heterogeneity: Individuals are often correlated *within* groups of observations (e.g. firms, regions, years)
- Regional characteristics (local economic conditions, cultural norm towards migration, geographic proximity to border) might have a regionally constant, "baseline" effect on the migration decision
- Fixed effects let us focus on *within-region* variation by controlling for time-invariant regional characteristics and general time trends
- Since this is essentially including a (potentially large) set of region and/or year dummies, the estimation of fixed effects models can become computationally extensive even with moderate sample sizes
- The `fixest` package (Bergé, 2018) offers a fast implementation of fixed effects estimators, covering a wide range of (micro-)econometric applications

Fast LPM Implementation

Code | Summary

- Fixed effects are specified following | on the RHS of the formula
- The `vcov` argument let's you handle the computation of standard errors, e.g.
  - "iid" for homoscedastic SE
  - "hetero" for heteroscedastic White SE
  - "cluster" for clustered SE
- Generic function `summary()` handles `fixest` models (slightly) different

```
library(fixest)

fe_lp_model <- feols(
  migrate ~
    fence + female + age + educ + married |
    municipality + period,
  vcov = "cluster",
  data = df_mig
)

summary(fe_lp_model)
```

Fast LPM Implementation

Code | Summary

```
## OLS estimation, Dep. Var.: migrate
## Observations: 156,113
## Fixed-effects: municipality: 23,  period: 41
## Standard-errors: Clustered (municipality)
##             Estimate Std. Error   t value   Pr(>|t|)
## fence    -0.001785   0.000921 -1.937784 0.06559061 .
## female   -0.001567   0.000438 -3.578626 0.00167557 **
## age      -0.000053   0.000013 -4.052211 0.00053104 ***
## educ      0.000028   0.000069  0.404796 0.68953676
## married -0.002145   0.000529 -4.057874 0.00052376 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## RMSE: 0.057395     Adj. R2: 0.001553
##                  Within R2: 9.327e-4
```

Fast Logit Implementation

Code | Summary

- Logit models can be easily computed by specifying the corresponding function family ("binomial")
- Overall handling very similar to the base R solution but computationally much faster

```
fe_logit_model <- feglm(
  migrate ~
    fence + female + age + educ + married |
    municipality + period,
  family = "binomial",
  vcov = "cluster",
  data = df_mig
)

summary(fe_logit_model)
```

Fast Logit Implementation

Code | Summary

```
## GLM estimation, family = binomial, Dep. Var.: migrate
## Observations: 156,028
## Fixed-effects: municipality: 20,  period: 41
## Standard-errors: Clustered (municipality)
##          Estimate Std. Error   z value   Pr(>|z|)
## fence   -0.470669   0.211948 -2.220682 2.6373e-02 *
## female  -0.472763   0.098667 -4.791516 1.6553e-06 ***
## age     -0.016581   0.005254 -3.156070 1.5991e-03 **
## educ     0.011944   0.021102  0.566025 5.7138e-01
## married -0.626415   0.104565 -5.990698 2.0894e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## Log-Likelihood: -3,318.8   Adj. Pseudo R2: 0.024585
##            BIC:  7,414.9     Squared Cor.: 0.002821
```

Performing Multiple Estimations

Code | Summary

- Different model specifications serve different purposes, i.e. robustness checks, investigating effect heterogeneity, transforming variables
- `fixest` offers easy ways to handle mutliple LHS and/or RHS estimations
  - Wrap outcome variables inside a vector, e.g. `c(y1, y2) ~ x1`
  - Use the `sw()` function for stepwise replacement of the RHS, e.g. `y ~ x1 + sw(x2, x3)` evaluates to `y ~ x1 + x2` and `y ~ x1 + x3`
  - Use the `csw()` for stepwise adding of variables to the RHS, e.g. `y ~ csw(x1 + x2, x3)` evaluates to `y ~ x1 + x2` and `y ~ x1 + x2 + x3`

```
library(fixest)

fe_lp_models <- feols(
  migrate ~
    csw(fence, female + age + educ + married) |
    municipality + period,
  vcov = "cluster",
  data = df_mig
)

summary(fe_lp_models)
```

Performing Multiple Estimations

Code | Summary

```
## Standard-errors: Clustered (municipality)
## Expl. vars.: fence
##        Estimate Std. Error  t value Pr(>|t|)
## fence -0.001741   0.000919 -1.89585 0.071192 .
## ---
## Expl. vars.: fence + female + age + educ + married
##          Estimate Std. Error   t value   Pr(>|t|)
## fence   -0.001785   0.000921 -1.937784 0.06559061 .
## female  -0.001567   0.000438 -3.578626 0.00167557 **
## age     -0.000053   0.000013 -4.052211 0.00053104 ***
## educ     0.000028   0.000069  0.404796 0.68953676
## married -0.002145   0.000529 -4.057874 0.00052376 ***
```

Creating Regression Tables

Regression Tables

> A variety of packages offers solutions to create off-the-shelve tables for regression output. The `msummary()` from the `modelsummary` package takes an `lm` object – or a (named) list of `lm` objects – as input and returns a customizable regression table.

```
msummary(
  models,
  output = "default",
  vcov = NULL,
  stars = FALSE,
  title = NULL,
  notes = NULL,
  coef_map = NULL,
  gof_map = NULL,
  ...
)
```

Regression Tables

> A variety of packages offers solutions to create off-the-shelve tables for regression output. The `msummary()` from the `modelsummary` package takes an `lm` object – or a (named) list of `lm` objects – as input and returns a customizable regression table.

```
msummary(
  models,
  output = "default",
  vcov = NULL,
  stars = FALSE,
  title = NULL,
  notes = NULL,
  coef_map = NULL,
  gof_map = NULL,
  ...
)
```

`models`

A model object, such as `lm`, or a (named) list of models.

Regression Tables

> A variety of packages offers solutions to create off-the-shelve tables for regression output. The `msummary()` from the `modelsummary` package takes an `lm` object – or a (named) list of `lm` objects – as input and returns a customizable regression table.

```
msummary(
  models,
  output = "default",
  vcov = NULL,
  stars = FALSE,
  title = NULL,
  notes = NULL,
  coef_map = NULL,
  gof_map = NULL,
  ...
)
```

`output`

If the table should be saved directly, the filename can be specified here. If the table should be customized afterwards, this argument should be set to the desired object type, e. g. `"kableExtra"`.

Regression Tables

> A variety of packages offers solutions to create off-the-shelve tables for regression output. The `msummary()` from the `modelsummary` package takes an `lm` object – or a (named) list of `lm` objects – as input and returns a customizable regression table.

```
msummary(
  models,
  output = "default",
  vcov = NULL,
  stars = FALSE,
  title = NULL,
  notes = NULL,
  coef_map = NULL,
  gof_map = NULL,
  ...
)
```

`vcov`

Replace model standard errors with robust standard errors by setting this argument to `"HC3"` or other variants of heteroscedasticity-consistent standard errors. Including robust standard errors in the table requires the `sandwich` package.

Regression Tables

> A variety of packages offers solutions to create off-the-shelve tables for regression output. The `msummary()` from the `modelsummary` package takes an `lm` object – or a (named) list of `lm` objects – as input and returns a customizable regression table.

```
msummary(
  models,
  output = "default",
  vcov = NULL,
  stars = FALSE,
  title = NULL,
  notes = NULL,
  coef_map = NULL,
  gof_map = NULL,
  ...
)
```

`stars`

Show stars to indicate statistical significance by passing a named numeric vector to this argument. Most commonly, this would be set to:

```
c("*" = .1, "**" = .05, "***" = .01)
```

Regression Tables

> A variety of packages offers solutions to create off-the-shelve tables for regression output. The `msummary()` from the `modelsummary` package takes an `lm` object – or a (named) list of `lm` objects – as input and returns a customizable regression table.

```
msummary(
  models,
  output = "default",
  vcov = NULL,
  stars = FALSE,
  title = NULL,
  notes = NULL,
  coef_map = NULL,
  gof_map = NULL,
  ...
)
```

`title`

Title for the table given as a string.

Regression Tables

> A variety of packages offers solutions to create off-the-shelve tables for regression output. The `msummary()` from the `modelsummary` package takes an `lm` object – or a (named) list of `lm` objects – as input and returns a customizable regression table.

```
msummary(
  models,
  output = "default",
  vcov = NULL,
  stars = FALSE,
  title = NULL,
  notes = NULL,
  coef_map = NULL,
  gof_map = NULL,
  ...
)
```

`notes`

Pass a list or vector of strings to this arguments to show notes below the table.

Regression Tables

> A variety of packages offers solutions to create off-the-shelve tables for regression output. The `msummary()` from the `modelsummary` package takes an `lm` object – or a (named) list of `lm` objects – as input and returns a customizable regression table.

```
msummary(
  models,
  output = "default",
  vcov = NULL,
  stars = FALSE,
  title = NULL,
  notes = NULL,
  coef_map = NULL,
  gof_map = NULL,
  ...
)
```

`coef_map`

Use a named character vector to map model variable names to coefficient names, e. g. `c(female = "Female")`. Coefficients are shown in the order of the character vector. If a coefficient is not given in the vector, it will be omitted from the table.

Regression Tables

> A variety of packages offers solutions to create off-the-shelve tables for regression output. The `msummary()` from the `modelsummary` package takes an `lm` object – or a (named) list of `lm` objects – as input and returns a customizable regression table.

```
msummary(
  models,
  output = "default",
  vcov = NULL,
  stars = FALSE,
  title = NULL,
  notes = NULL,
  coef_map = NULL,
  gof_map = NULL,
  ...
)
```

`gof_map`

A character vector specifying goodness-of-fit statistics and other model information to show at the bottom of the table. Measures are reported in the order given in the vector. See `get_gof(<YOUR-MODEL>)` for a list of measures to choose from. Names of the `data.frame` correspond to measure names that have to be provided to show in the table.

If you want to use a custom name for the measures, you can pass a `data.frame` object with the columns `"raw"`, `"clean"` and `"fmt"` instead of a character vector.

Creating Regression Tables

Task | Code (models) | Code (table)

Use the `msummary()` function from the `modelsummary` package to create a table showing the results from the LPM and the logit models used so far.

Show only coefficients for your variable of interest and additional controls. Cluster standard errors on municipality level.

Creating Regression Tables

Task | Code (models) | Code (table)

```
library(modelsummary)

formula <- migrate ~
  csw(fence, female + age + educ + married) | municipality + period

lpm <- feols(formula, vcov = "cluster", data = df_mig)

logit <- feglm(formula, family = "binomial", vcov = "cluster", data = df_mig)
```

```
## NOTE: 3/0 fixed-effects (85 observations) removed because of only 0 (or only 1) outcomes.
```

Creating Regression Tables

Task | Code (models) | Code (table)

```
msummary(
  list(lpm, logit),
  output = "kableExtra",
  col.names = c("", "(1)", "(2)", "(3)", "(4)"),
  stars = c("*" = .1, "**" = .05, "***" = .01),
  title = "Impact of Fence Construction on Mexico-US Migration",
  notes = str_c(
    "Columns (1) and (2) show OLS results, columns (3) and (4) show Logit results. ",
    "Parantheses show clustered standard errors."
  ),
  coef_map = c(
    fence = "Fence Construction", age = "Age", female = "Female",
    educ = "Years of Schooling", married = "Married"
  ),
  gof_map = c(
    "r2", "r2.adjusted", "r2.within", "r2.within.adjusted", "nobs"
  )
)
```

Impact of Fence Construction on Mexico-US Migration

	(1)	(2)	(3)	(4)
Fence Construction	-0.002*	-0.002*	-0.449**	-0.471**
	(0.001)	(0.001)	(0.213)	(0.212)
Age		0.000***		-0.017***
		(0.000)		(0.005)
Female		-0.002***		-0.473***
		(0.000)		(0.099)
Years of Schooling		0.000		0.012
		(0.000)		(0.021)
Married		-0.002***		-0.626***
		(0.001)		(0.105)
R2 Within	0.000	0.001	0.001	0.021
R2 Within Adj.	0.000	0.001	0.001	0.020
Num.Obs.	156113	156113	156028	156028

* p < 0.1, ** p < 0.05, *** p < 0.01

Columns (1) and (2) show OLS results, columns (3) and (4) show Logit results. Parantheses show clustered standard errors.

References

Bergé, L. (2018). "Efficient estimation of maximum likelihood models with multiple fixed-effects: the R package FENmlm". In: *CREA Discussion Papers*.

Feigenberg, B. (2020a). "Fenced Out: The Impact of Border Construction on US-Mexico Migration". In: *American Economic Journal: Applied Economics* 12.3, pp. 106-39. DOI: 10.1257/app.20170231. URL: https://www.aeaweb.org/articles?id=10.1257/app.20170231.

Feigenberg, B. (2020b). *Replication package for: Fenced Out: The Impact of Border Construction on US-Mexico Migration.* American Economic Association. URL: https://www.aeaweb.org/journals/dataset?id=10.1257/app.20170231.

Wickham, H. and G. Grolemund (2016). *R for data science. import, tidy, transform, visualize, and model data*. O'Reilly. URL: https://r4ds.had.co.nz/.

Zeilis, A., G. McDermott, and K. Tappe (2024). *CRAN Task View: Econometrics*. Version 2024-06-03. URL: https://CRAN.R-project.org/view=Econometrics.