Formulas

The table below shows the correspondance between regression models in Stata and R

Stata	R
`y x1 x2`	`y ~ x1 + x2`
`y x1,nocons`	`y ~ 0 + x1`
`y i.x1`	`y ~ as.factor(x1)`
`y c.x1#c.x2`	`y ~ x1:x2`
`y c.x1##c.x2`	`y ~ x1*x2`
`y c.x1##i.x2`	`y ~ x1*as.factor(x2)`

Estimation commands

- The package `lfe` implements models with high dimensional fixed effects or/and instrumental variables

```
N <- 1e6
df <- tibble(
  id1 = sample(c("id01", "id02", "id03"), N, TRUE),
  id2 = sample(5, N, TRUE),
  y   = sample(round(runif(100, max = 100), 4), N, TRUE),
  x1  = sample(round(runif(100, max = 100), 4), N, TRUE),
  x2  = sample(round(runif(100, max = 100), 4), N, TRUE),
  x3  = sample(round(runif(100, max = 100), 4), N, TRUE)
)
```

You first need to convert categorical variables into factors:

```
df <- df %>% mutate(id1 = as.factor(id1))
df <- df %>% mutate(id2 = as.factor(id2))
```

To estimate a linear model:

Stata	`areg y x1 [w=x3], a(id1) cl(id1)`
lfe	`felm(y ~ x1 | id1 | 0 | id1, df, weight = x3))`

Stata	`reghdfe y x3 (x2 = x1), a(id1) cl(id1 id2)`
lfe	`felm(y ~ x3 | id1 | (x2 ~ x1) | id1 + id2, df)`

Stata	`reghdfe y x2, a(c.x3#i.id1 id1) cl(id1 id2)`
lfe	`felm(y ~ x2 | x3:id1 + id1, df)`

Errors reported by `felm` are similar to the ones given by `areg` and not `xtivreg` / `xtivreg2`. Manual adjustments can be done similarly to Gormley and Matsa.

- The package `gmm` implements GMM
- The package `rdd` implements regression discontinuity models.
- The package `matchit` implements matching procedures.

Post-estimation commands

An estimation function returns a list that contains the estimates, the covariance matrix, and in a lot of cases, the residuals, the predicted values, or the original variables used in the estimation. Apply the `names` function to examine the result:

```
result <- felm(y ~ x2, df)
names(result)
#>  [1] "coefficients"  "badconv"       "Pp"            "N"             "p"
#>  [6] "inv"           "beta"          "response"      "fitted.values" "residuals"
#> [11] "r.residuals"   "terms"         "cfactor"       "numrefs"       "df"
#> [16] "df.residual"   "rank"          "exactDOF"      "vcv"           "robustvcv"
#> [21] "clustervcv"    "cse"           "ctval"         "cpval"         "clustervar
"
#> [26] "se"            "tval"          "pval"          "rse"           "rtval"
#> [31] "rpval"         "xp"            "call"
pryr::object_size(result)
#> [1] 88 MB
```

Applying `summary` prints a table similar to Stata output

```
summary(result)
#> Call:
#>    felm(formula = y ~ x2, data = df)
#>
#> Residuals:
#>     Min      1Q  Median      3Q     Max
#> -48.834 -23.175  -5.028  25.222  50.939
#>
#> Coefficients:
#>              Estimate Std. Error t value Pr(>|t|)
#> (Intercept) 48.746112   0.064228 758.949   <2e-16 ***
#> x2           0.001997   0.001059   1.886   0.0593 .
#> ---
#> Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
#>
#> Residual standard error: 29.91 on 999998 degrees of freedom
#> Multiple R-squared: 3.556e-06   Adjusted R-squared: 1.556e-06
#> F-statistic:3.556 on 1 and 999998 DF, p-value: 0.05934
```

The package `stargazer` allows to combine several regression results in a table:

```
stargazer(result, type = "text")
#> ===============================================
#>                         Dependent variable:
#>                     ---------------------------
#>                                  y
#> -----------------------------------------------
#> x2                            -0.0004
#>                               (0.001)
#>
#> Constant                     50.315***
#>                               (0.064)
#>
#> -----------------------------------------------
#> Observations                 1,000,000
#> R2                            0.00000
#> Adjusted R2                  -0.00000
#> Residual Std. Error    29.707 (df = 999998)
#> ===============================================
#> Note:               *p<0.1; **p<0.05; ***p<0.01
```