

# replicate Function Guide

Project Replication Toolkit

August 12, 2025

## 1 Overview

The `replicate` function is the main entry point for running econometric replications in this project.

## 2 Function Signature

```
replicate(metadata, y, X, interest, endog_x=None, z=None, fe=None, elasticity
          =False,
          replicated=False, kwargs_estimator=None, kwargs_fit=None,
          kwargs_ols=None, kwargs_ppml=None, fit_full_model=False,
          output=False, output_dir=None, overwrite=False)
```

## 3 Arguments

### **metadata : dict**

Model info for tracking and saving. Must include `paper_id`, `table_id`, `panel_identifier`, `model_type`.

### **y : array-like**

Dependent variable (Series, ndarray, or DataFrame column). Note that this has to be the non-logged version, so exponentiate if author has logged variable.

### **X : array-like**

Independent variables matrix (DataFrame or ndarray).

### **interest : str or list**

Variable(s) of primary interest to highlight/report.

### **endog\_x : list[int**

or list[str], optional] Endogenous regressors referenced by **column indices** or **column names**. Requires `z`.

### **z : array-like, optional**

Instrumental variables matrix. Required if `endog_x` is provided.

### **fe : array-like, optional**

Fixed effects identifiers aligned with `y` (e.g., panel or group IDs). Must be in `X` matrix.

### **elasticity : bool, default False**

Compute/report elasticities if supported by the estimator.

### **replicated : bool, default False**

Set this to true after you've managed to replicate a result.

**kwargs\_estimator : dict, optional**

Extra keyword args for estimator init (e.g., {'estimator\_type': 'ols'}). (unused for now)

**kwargs\_fit : dict, optional**

Reserved for higher-level `.fit()` options (typically unused here).

**kwargs\_ols : dict, optional**

Options for OLS (e.g., {'cov\_type': 'HC3'}). Defaults to HC3 if not given.

**kwargs\_ppml : dict, optional**

Options for PPML (present for compatibility; not invoked here).

**fit\_full\_model : bool, default False**

Placeholder for future full-model fit (not implemented).

**output : bool, default False**

If True, save output bundle to `output_dir` (metadata + y/X/+z).

**output\_dir : str or Path, optional**

Directory to write outputs; required if `output=True`. Will be in your config file.

**overwrite : bool, default False**

Overwrite existing files in `output_dir` if they exist.

## 4 Returns

A dictionary with handles to useful objects:

- 'replicator' — the configured Replicator instance.
- 'ols\_results' — the statsmodels-like OLS results object.

Future versions may include PPML results when fixed-effects support is finalised.

## 5 Notes

- Only OLS is executed by this function. PPML code exists but is disabled here pending FE support.
- Saved bundle files (if `output=True`): `metadata.json`, `y.parquet`, `X.parquet`, and `z.parquet` (if instruments provided).
- All arrays must share the same number of rows and align with `y`.

## 6 Examples

### 6.1 Minimal OLS replication

```
from your_package import replicate

res = replicate(
    metadata={
        'paper_id': '001',
        'table_id': '3.2',
        'panel_identifier': 'district_year',
        'model_type': 'log-linear'
```

```

    },
    y=df['outcome'],
    X=df[['treatment', 'age', 'income']],
    interest='treatment',
    output=True,
    output_dir=OUTPUT_DIR,
    replicated=True,
    overwrite=False
)

# Access results
ols = res['ols_results']
print(ols.summary())

```

## 6.2 OLS with Fixed Effects

```

res = replicate(
    metadata={
        'paper_id': '002',
        'table_id': '4.1',
        'panel_identifier': 'firm_year',
        'model_type': 'log-linear'
    },
    y=np.exp(df['log_sales']),
    X=df[['policy', 'size', 'age', 'firm_id']],
    interest='policy',
    fe=['firm_id'], # fixed effects
    kwargs_ols={'cov_type': 'HC3'},
    output=True,
    output_dir=OUTPUT_DIR,
    replicated=True,
    overwrite=False
)

```

## 6.3 OLS with Endogenous Regressor + IV

```

# Suppose 'income' is endogenous; instrument with 'distance' and '
    legacy_index'
res = replicate(
    metadata={
        'paper_id': '003',
        'table_id': '5.3',
        'panel_identifier': 'region_year',
        'model_type': 'log-log'
    },
    y=df['outcome'],
    X=df[['treatment', 'income', 'age']],
    interest='treatment',
    endog_x=['income'], # could also be [1] if using index
    z=df[['distance', 'legacy_index']], # instruments
    kwargs_ols={'cov_type': 'HC3'},
    output=True,
    output_dir=OUTPUT_DIR,
    replicated=True,
    overwrite=False
)

```

## 7 Tips

- Use column names in `endog_x` to avoid errors when column order changes.
- Ensure all arrays (`y`, `X`, `z`, `fe`) are row-aligned.
- Prefer HC3 standard errors for small/medium samples.
- You can specify metadata, `y`, `X`, `z` as variables in the code, and then just call those variables when using function.