

# Nonparametric Methods in Stata

Eduardo García Echeverri

Stata Webinar, 2023

# Outline

- 1 The appeal of nonparametric methods
- 2 Estimating probability densities
- 3 Estimating conditional expectations
  - Kernel Regression
  - Series Approximations
- 4 Advantages and Limitations of nonparametric methods

# What are a nonparametric methods?

Estimation method that is agnostic about:

- **Probability distributions** of outcomes and covariates.
- **Functional forms** relating outcomes and covariates.

Unlike **parametric methods**, which require us to specify these two.

## Example

What is the effect of **smoking during pregnancy** (`msmoke`) on the **babies weight** at birth (`bweight`)?

- **Parametric:**

$$\text{msmoke} = \beta_0 + \beta_1 \text{bweight} + \gamma \text{Controls} + \varepsilon$$

$$\varepsilon \sim N(0, \sigma^2)$$

- **Nonparametric:**

$$\text{msmoke} = g(\text{bweight}, \text{Controls}) + \varepsilon$$

$$\mathbb{E}[\varepsilon | \text{bweight}, \text{Controls}] = 0$$

# The Appeal of Nonparametric Methods

## Advantages:

1. Avoid the problems that arise with **misspecification**.
2. Improve predictions.
3. Knowing the functional form is not needed to answer important research questions.
4. Easily implementable in Stata.

## Disadvantages:

1. Data intensive (esp. with many covariates)
2. Computationally costly.

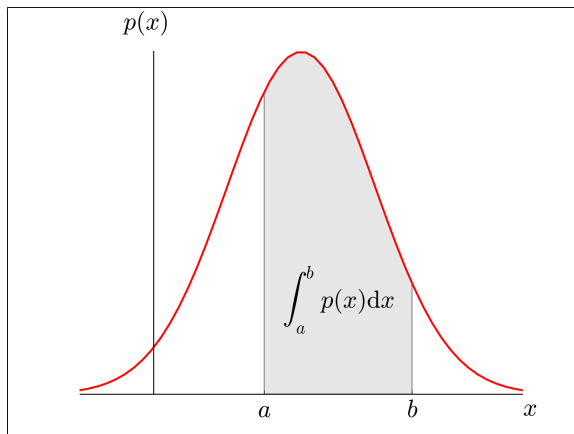
- 1 The appeal of nonparametric methods
- 2 Estimating probability densities
- 3 Estimating conditional expectations
  - Kernel Regression
  - Series Approximations
- 4 Advantages and Limitations of nonparametric methods

# Outline

- 1 The appeal of nonparametric methods
- 2 Estimating probability densities
- 3 Estimating conditional expectations
  - Kernel Regression
  - Series Approximations
- 4 Advantages and Limitations of nonparametric methods

# What is a probability density?

Probability densities tell us how a **random variable is distributed**.





# Why are they useful?

**Probability densities** can show us:

1. Which values of the variable are **likely/unlikely**.
2. The **mode(s)** of the distribution (**peaks** of the density)
3. The **range** of values of the variable
4. The probability of **extreme events** (tails of the distribution)

# Parametric estimation of densities

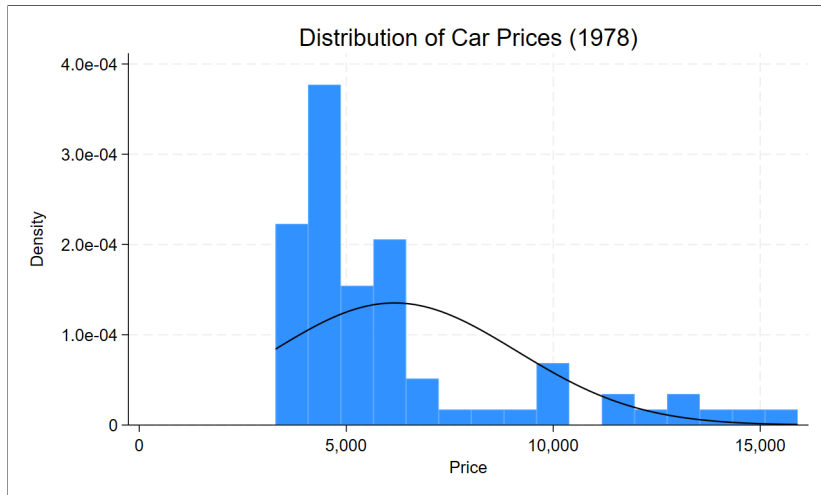
First, specify a **type** of probability density function:

1. **Normal** (most common choice)
2. Log-normal
3. ...

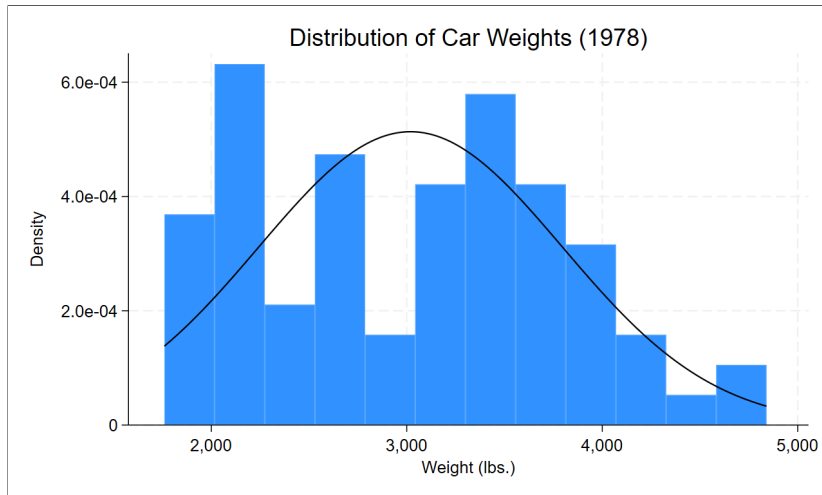
Second, based on **sample mean and variance**, **estimate** the density within the type that **best matches data**.

**Problem:** misspecification. Normal might be poor approximation.

## Normality may not hold – fat right tails



## Normality may not hold – multiple modes



# Kernel Density Estimators

Kernel density estimators **don't assume a functional form**.

To estimate a **density** at point  $x$ :

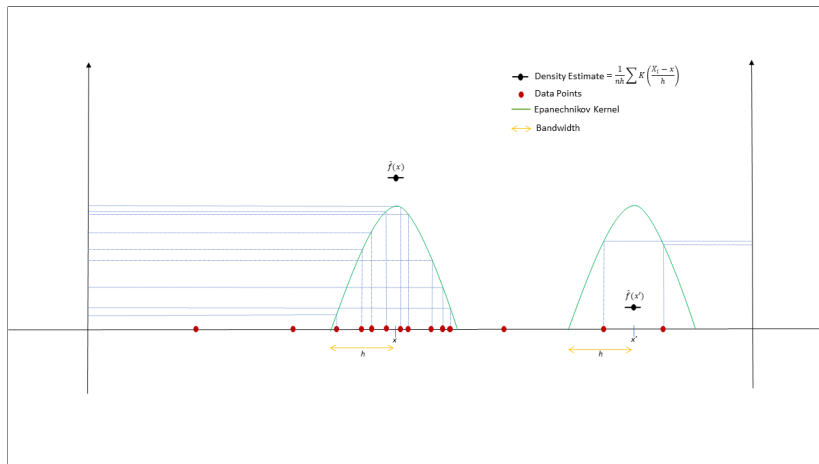
- **“Weighted count”** of how many data points are close to  $x$ ,

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^n K\left(\frac{x - X_i}{h}\right)$$

$h$ : **Bandwidth** – a positive number (very important!)

$K(\cdot)$ : **Kernel** – a function ( $K : \mathbb{R} \rightarrow \mathbb{R}$ )

# Illustration



# Implementation using Stata

```
kdensity varname [if] [in] [weight] [, options]
```

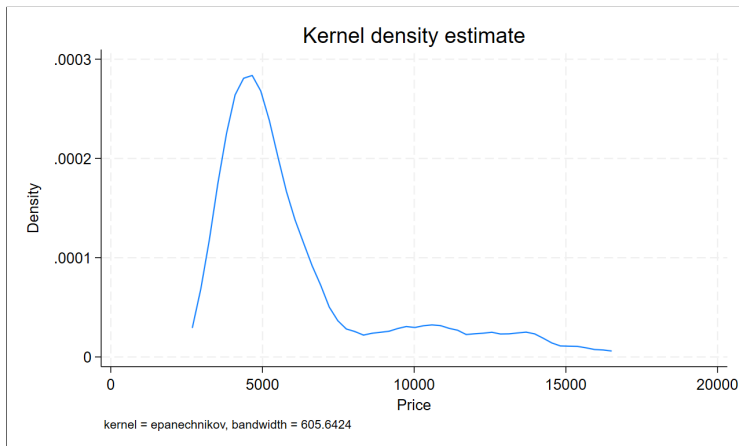
## Quick Start:

Graph of the **kernel density estimate** for variable  $x_1$

- `kdensity x1`

## Example 1: Graphing a probability density

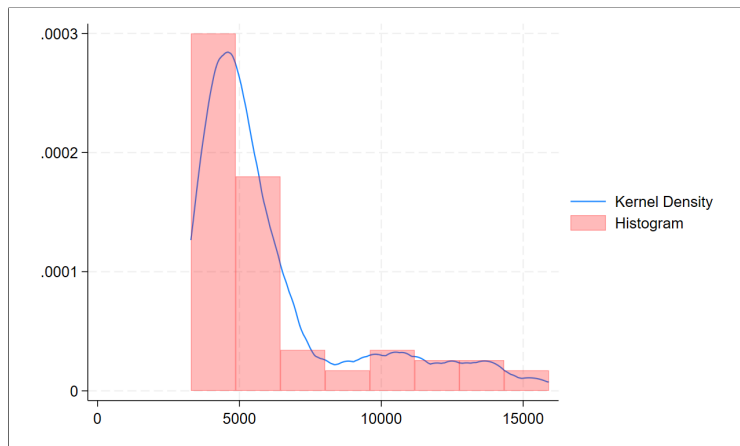
- `sysuse auto, clear`
- `kdensity price`





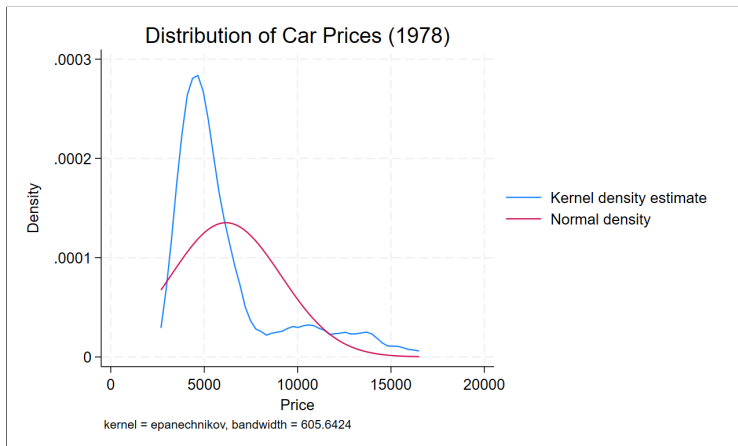
## Example 1: Better fit the data

- `twoway kdensity price || histogram price, legend(order(1 "Kernel Density" 2 "Histogram")) color(red%30)`



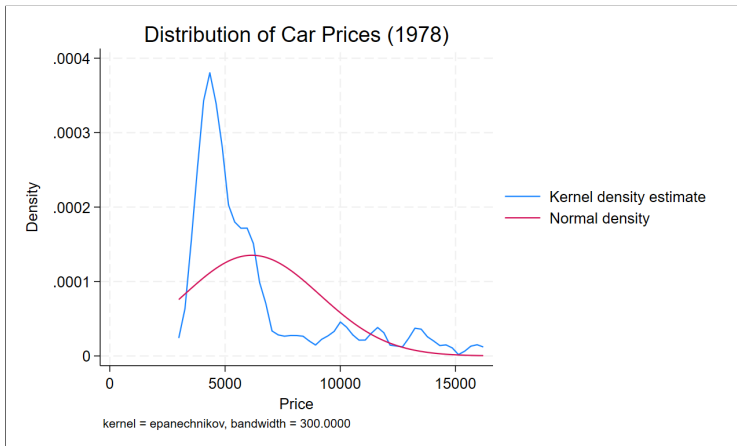
## Example 2: Comparing to a normal distribution

- `kdensity price, normal title("Distribution of Car Prices")`



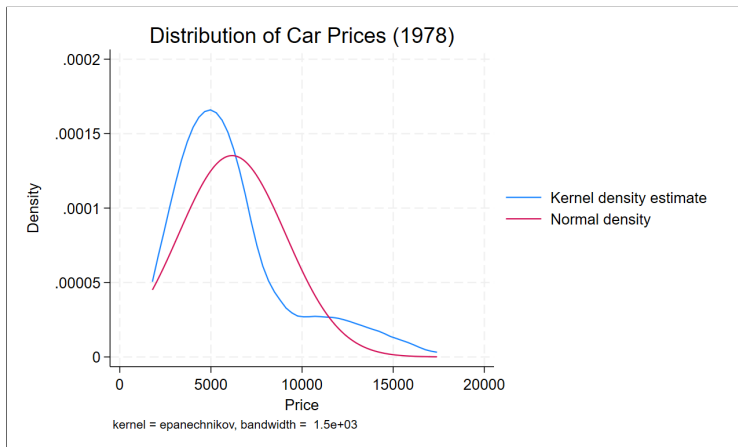
## Example 3: Using a smaller bandwidth (overfitting)

- `kdensity price, normal bwidth(300) title("Distribution of Car Weights")`



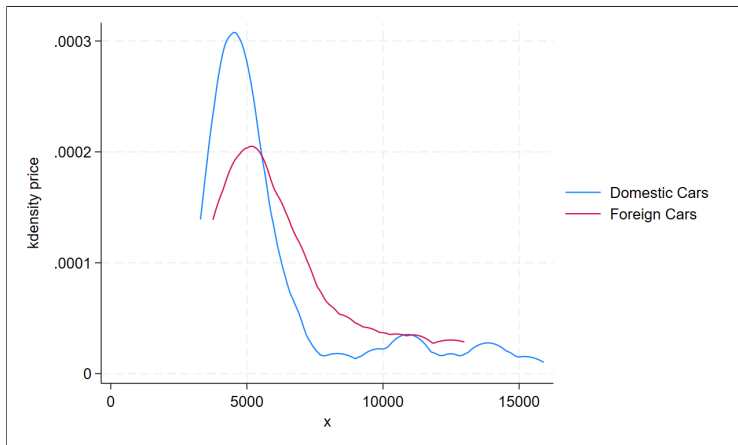
## Example 4: Using a larger bandwidth (oversmoothing)

- `kdensity price, normal bwidth(1500) title("Distribution of Car Prices")`



## Example 5: Comparing Price Distributions

- `twoway kdensity price if foreign==0 || kdensity price if foreign==1`



# Useful Options

`kernel`: specify the kernel function:

- `epanechnikov`; default
- `gaussian`
- `triangle...`

`generate(newvar1, newvar2)`: store estimation points in *newvar<sub>1</sub>* and density estimates in *newvar<sub>2</sub>*

`nograph`: Suppress the graph.

# Option generate

- `kdensity price, generate(pricepoint density) nograph`
- browse pricepoint density

The screenshot shows the Stata Data Editor window titled "2 - Data Editor (Browse) - [auto]". The main window displays a table with two columns: `density` and `pricepoint`. The `density` column contains values ranging from 0.00002919 to 0.0002250, and the `pricepoint` column contains values ranging from 2685.3578 to 11714.768. The right-hand pane shows the `Variables` list with `make` selected. The `Properties` pane shows details for the `make` variable, including its name, label, type, format, and value label.

density	pricepoint
0.00002919	2685.3578
0.0000687	2967.5267
0.00011843	3249.6958
0.00017546	3531.8648
0.00022515	3814.0339
0.00028398	4096.203
0.00034281	4378.3721
0.00040164	4660.5412
0.00046047	4942.7102
0.0005193	5224.8793
0.00057813	5507.0484
0.00063696	5789.2175
0.00069579	6071.3865
0.00075462	6353.5556
0.00081345	6635.7247
0.00087228	6917.8938
0.00093111	7200.0628
0.00098994	7482.2319
0.00104877	7764.401
0.0011076	8046.5701
0.00116643	8328.7392
0.00122526	8610.9082
0.00128409	8893.0773
0.00134292	9175.2464
0.00140175	9457.4155
0.00146058	9739.5845
0.00151941	10021.754
0.00157824	10303.923
0.00163707	10586.092
0.0016959	10868.261
0.00175473	11150.43
0.00181356	11432.599
0.00187239	11714.768

The right-hand pane shows the `Variables` list with `make` selected. The `Properties` pane shows details for the `make` variable, including its name, label, type, format, and value label.

Name	Label	Type	Format	Value l
<input checked="" type="checkbox"/> make	Make and model	str18	%-18s	
<input type="checkbox"/> price	Price	int	%8.0gc	
<input type="checkbox"/> mpg	Mileage (mpg)	int	%8.0g	
<input type="checkbox"/> rep78	Repair record 1978	int	%8.0g	
<input type="checkbox"/> headroom	Headroom (in.)	float	%6.1f	
<input type="checkbox"/> trunk	Trunk space (cu. ft.)	int	%8.0g	
<input type="checkbox"/> weight	Weight (lbs.)	int	%8.0gc	
<input type="checkbox"/> length	Length (in.)	int	%8.0g	
<input type="checkbox"/> turn	Turn circle (ft.)	int	%8.0g	

The `Properties` pane shows details for the `make` variable, including its name, label, type, format, and value label.

Property	Value
Name	make
Label	Make and model
Type	str18
Format	%-18s
Value label	
Notes	
Frame	default
Filename	auto.dta
Label	1978 automobile data
Notes	
Variables	14
Observations	74
Size	4.36K

The status bar at the bottom shows: `Vars: 2 of 14. Order: Dataset Obs: 74 Filter: Off Mode: Browse CAP NUM`

# Outline

- 1 The appeal of nonparametric methods
- 2 Estimating probability densities
- 3 Estimating conditional expectations
  - Kernel Regression
  - Series Approximations
- 4 Advantages and Limitations of nonparametric methods



# Conditional Expectations–Regression Function

## Objective:

Estimate the **conditional mean/regression function**  $g(\cdot)$ :

$$y = g(X) + \varepsilon$$

$$\mathbb{E}[\varepsilon|X] = 0$$

## Parametric approaches:

- Linear regression:  $g(X) = x\beta$
- Probit:  $g(X) = \Phi(x\beta)$
- Poisson:  $g(X) = \exp(x\beta)$

**Problems** may arise due to **misspecification**.

**Kernel regression** doesn't assume a functional form for  $g(\cdot)$

# Outline

- 1 The appeal of nonparametric methods
- 2 Estimating probability densities
- 3 Estimating conditional expectations
  - Kernel Regression
  - Series Approximations
- 4 Advantages and Limitations of nonparametric methods

# Kernel Regression – Local Linear

To estimate the **conditional expectation** at point  $x$ :

- “**Weighted regression**” of  $y$  on  $X$  using points close to  $x$ ,

$$\min_{\gamma_0, \gamma_1} \sum_{i=1}^n \left( y_i - \gamma_0 - \gamma_1(x_i - x) \right)^2 \cdot K\left(\frac{x - X_i}{h}\right)$$

$\gamma_0$  : Predicted **conditional expectation**  $\hat{g}(x)$

$\gamma_1$ : Predicted **derivative** of  $g(\cdot)$  at point  $x$ .

$h$ : **Bandwidth** – a positive number (very important!)

$K(\cdot)$ : **Kernel** – a function ( $K : \mathbb{R} \rightarrow \mathbb{R}$ )

# Kernel Regression – Local Constant

To estimate the **conditional expectation** at point  $x$ :

- “**Weighted regression**” of  $y$  on  $X$  using points close to  $x$ ,

$$\min_{\gamma_0, \gamma_1} \sum_{i=1}^n \left( y_i - \gamma_0 \right)^2 \cdot K\left( \frac{x - X_i}{h} \right)$$

$\gamma_0$  : Predicted **conditional expectation**  $\hat{g}(x)$

$h$ : **Bandwidth** – a positive number (very important!)

$K(\cdot)$ : **Kernel** – a function ( $K : \mathbb{R} \rightarrow \mathbb{R}$ )

# Implementation in Stata

```
npregress kernel depvar indepvars [if] [in] [, options]
```

## Quick Start:

Local-linear kernel regression of  $y$  on  $x$  and discrete covariate  $a$ :

- `npregress kernel y x i.a`

Local-const. kernel regression of  $y$  on  $x$  and discrete covariate  $a$ :

- `npregress kernel y x i.a, estimator(constant)  
noderivatives`

## Example 7: The effect of smoking on babies' weight

### Outcome:

- `bweight`: **Baby's weight** in grams.

### Treatment:

- `msmoke`: **Cigarettes** smoked during pregnancy (4 categories)

### Controls:

- `mage`: Mother's age.
- `medu`: Mother's educational attainment.
- `alcohol`: 1 if alcohol was consumed during pregnancy.
- `prenatal`: trimester of first prenatal visit.

# Example 7: Local-Linear Kernel Regression

```
. npregress kernel bweight mage medu i.msmove i.alcohol i.prenatal, nolog
```

Bandwidth

	Mean	Effect
mage	2.178108	3.09538
medu	.9284651	1.319472
msmove	.5	.5
alcohol	.5	.5
prenatal	.5	.5

Local-linear regression

Continuous kernel : epanechnikov

Discrete kernel : lracine

Bandwidth : cross-validation

Number of obs = 4,605

E(Kernel obs) = 4,605

R-squared = 0.0893

	bweight	Estimate
<b>Mean</b>	bweight	3357.97
<b>Effect</b>	mage	9.550276
	medu	4.95285
	msmove	
(1-5 daily vs 0 daily)		-114.3037
(6-10 daily vs 0 daily)		-214.6567
(11+ daily vs 0 daily)		-306.8657
	alcohol	
(1 vs 0)		-45.48752
	prenatal	
(1 vs 0)		31.2373
(2 vs 0)		10.41102
(3 vs 0)		-30.41237

Note: Effect estimates are averages of derivatives for continuous covariates and averages of contrasts for factor covariates.

Note: You may compute standard errors using `vce(bootstrap)` or `reps()`.

# Example 7: Generated Predictions

2 - Data Editor (Browse) - [lbw\_sim2]

File Edit View Data Tools

bweight[1] 3459

	bweight	alcohol	mage	medu	msmoke	prenatal	_Mean_bw...
1	3459	0	24	14 0 daly		1	3432.9464
2	3260	0	20	10 0 daly		1	3371.1014
3	3572	0	22	9 0 daly		1	3454.9649
4	2948	0	26	12 0 daly		1	3431.5499
5	2410	0	20	12 0 daly		1	3335.7757
6	3147	0	27	12 0 daly		1	3427.7775
7	3799	0	27	12 0 daly		1	3427.7775
8	3629	0	24	12 0 daly		1	3428.6979
9	2835	0	21	12 0 daly		1	3370.4283
10	3880	0	30	15 0 daly		1	3498.2042
11	3090	0	26	12 11+ daly		1	3063.5346
12	3345	0	20	12 0 daly		1	3335.7757
13	4013	0	34	14 0 daly		1	3496.9393
14	3771	0	21	8 0 daly		1	3399.5816
15	662	0	23	12 0 daly		2	3378.9015
16	3657	0	22	12 0 daly		1	3397.0289
17	3572	0	26	12 0 daly		1	3431.5499
18	3430	0	40	16 0 daly		1	3375.6734
19	4479	0	34	12 0 daly		1	3454.5931
20	3166	0	27	12 6-10 daly		2	3188.7895
21	4253	0	33	12 0 daly		1	3442.9508
22	4054	0	27	14 0 daly		1	3457.9927
23	3160	0	25	12 0 daly		1	3434.6629
24	2466	0	22	12 0 daly		1	3397.0289
25	3147	0	19	10 11+ daly		2	3093.1773
26	3232	0	33	14 0 daly		1	3491.6733
27	3005	0	19	12 0 daly		1	3290.9059
28	1899	0	36	17 0 daly		1	3470.78
29	4026	0	33	14 0 daly		1	3491.6733
30	3969	0	28	12 0 daly		2	3411.2734
31	4167	0	19	9 0 daly		1	3320.6182
32	4054	1	32	16 0 daly		1	3491.0059
33	3660	0	28	15 0 daly		1	3481.2317

**Variables**

Filter variables here

<input checked="" type="checkbox"/>	Name	Label	Type	Format	Value L
<input checked="" type="checkbox"/>	bweight	Infant birthweight (gra...	int	%9.0g	
<input type="checkbox"/>	mmarried	1 if mother married	byte	%11.0g	mmarried
<input type="checkbox"/>	mhispanic	1 if mother hispanic	byte	%9.0g	
<input type="checkbox"/>	fhispanic	1 if father hispanic	byte	%9.0g	
<input type="checkbox"/>	foreign	1 if mother born abroad	byte	%9.0g	
<input checked="" type="checkbox"/>	alcohol	1 if alcohol consumed d...	byte	%9.0g	
<input type="checkbox"/>	deadkids	Previous births where n...	byte	%9.0g	
<input checked="" type="checkbox"/>	mage	Mother's age	byte	%9.0g	
<input checked="" type="checkbox"/>	medu	Mother's education attai...	byte	%9.0g	

**Variables** **Snapshots**

**Properties**

**Variables**

Name	bweight
Label	Infant birthweight (grams)
Type	int
Format	%9.0g
Value label	
Notes	

**Data**

Frame	default
Filename	lbw_sim2.dta
Label	Excerpt from Cattaneo (2010) Journal
Notes	
Variables	34
Observations	4,605
Size	512.67K

Ready Vars: 7 of 34 Order: Dataset Obs: 4,605 Filter: Off Mode: Browse CAP NJM



# Example 8: Bootstrap Standard Errors

```
. npregress kernel bweight mage medu i.msmsoke i.alcohol i.prenatal, nolog reps(40) seed(1234)
(running npregress on estimation sample)
```

Bootstrap replications (40): .....10.....20.....30.....40 done

Bandwidth

	Mean	Effect
mage	2.178108	3.09538
medu	.9284651	1.319472
msmsoke	.5	.5
alcohol	.5	.5
prenatal	.5	.5

Local-linear regression                      Number of obs        =        4,605  
 Continuous kernel : epanechnikov           E(Kernel obs)        =        4,605  
 Discrete kernel : lracine                   R-squared             =        0.0893  
 Bandwidth                   : cross-validation

	bweight	Observed estimate	Bootstrap std. err.	z	P> z	Percentile [95% conf. interval]	
Mean	bweight	3357.97	10.23615	328.05	0.000	3341.884	3379.431
Effect							
	mage	9.550276	1.77653	5.38	0.000	6.437531	12.45403
	medu	4.95285	7.087289	0.70	0.485	-9.00919	15.87284
	msmsoke						
(1-5 daily vs 0 daily)		-114.3037	10.67519	-10.71	0.000	-135.9948	-95.59864
(6-10 daily vs 0 daily)		-214.6567	19.9158	-10.78	0.000	-257.2621	-179.2783
(11+ daily vs 0 daily)		-306.8657	29.30562	-10.47	0.000	-367.6013	-261.8348
	alcohol						
(1 vs 0)		-45.48752	42.34277	-1.07	0.283	-119.0151	39.6552
	prenatal						
(1 vs 0)		31.2373	23.23159	1.34	0.179	-16.08415	73.16318
(2 vs 0)		10.41102	41.17059	0.25	0.800	-79.54673	84.2395
(3 vs 0)		-30.41237	61.28031	-0.50	0.620	-158.3454	86.92535

Note: Effect estimates are averages of derivatives for continuous covariates and averages of contrasts for factor covariates.

## Example 9: Displaying Results Graphically

```
. npregress kernel bweight mage, nolog
```

Bandwidth

	Mean	Effect
mage	5.795516	3.695218

Local-linear regression

Number of obs = 4,605

Kernel : epanechnikov

E(Kernel obs) = 4,605

Bandwidth: cross-validation

R-squared = 0.0132

bweight	Estimate
Mean	
bweight	3354.357
Effect	
mage	11.41642

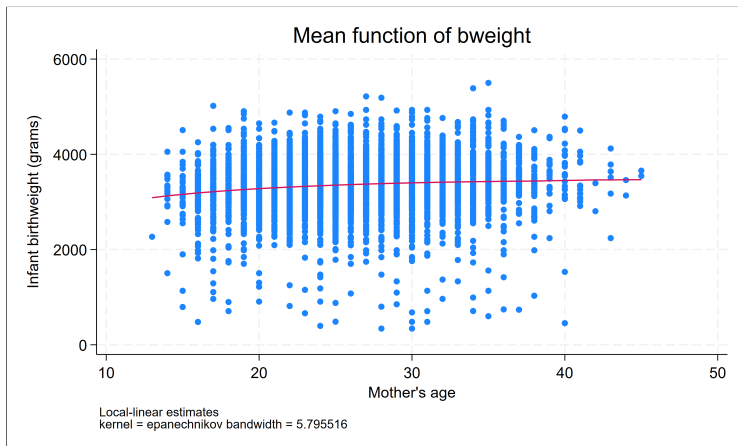
Note: Effect estimates are averages of derivatives.

Note: You may compute standard errors using `vce(bootstrap)` or `reps()`.

```
.
```

```
. npgraph
```

## Example 9: Displaying Results Graphically



# Outline

- 1 The appeal of nonparametric methods
- 2 Estimating probability densities
- 3 Estimating conditional expectations**
  - Kernel Regression
  - Series Approximations**
- 4 Advantages and Limitations of nonparametric methods

## Series Approximations – Introduction

We can also **approximate function**  $g(\cdot)$  **using series**. For instance, recall the Taylor expansion:

$$g(x) = g(0) + \frac{g'(0)}{1!} \cdot x + \frac{g''(0)}{2!} \cdot x^2 + \frac{g^{(3)}(0)}{3!} \cdot x^3 + \text{Remainder}$$

Thus, we can **estimate function**  $g(\cdot)$  as:

$$\hat{g}(x_i) = z(x_i)\hat{\beta}$$

Where  $z(x_i) = (1, x_i, x_i^2, x_i^3)$  and  $\hat{\beta} = (Z^\top Z)^{-1} Z^\top y$ .

# Basis Supported in Stata

## **Polynomial Basis:**

Function  $g(\cdot)$  is approximated with a polynomial.

## **Piecewise Polynomial Spline Basis:**

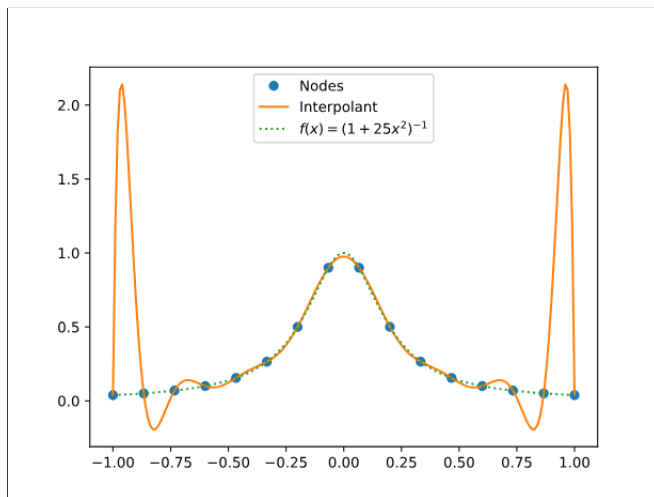
Function  $g(\cdot)$  is approximated with a piecewise polynomial.

## **B-Spline Basis (default):**

Function  $g(\cdot)$  is approximated with a spline function. A spline is a smoothed piecewise polynomial.

Piecewise polynomials and splines **alleviate Runge's phenomenon**.

# Runge's Phenomenon



# Implementation in Stata

```
npregress series depvar indepvars [if] [in] [weight] [, options]
```

## Quick Start:

Nonparametric regression of  $y$  on  $x$  and discrete covariate  $a$  (using B-spline basis):

- `npregress series y x i.a`

As above, but use a polynomial basis instead:

- `npregress series y x i.a, polynomial`



## Example 10: Effect of fines on the number of DUI citations

### Outcome:

- citations: **Annual DUI citations** in a county.

### Treatment:

- fines: **Fines** for drunk driving in the county

### Controls:

- csize: Size of the county (3 categories)
- college: 1 if there is a college in the county.

# Nonparametric regression in Stata using series

```
. npregress series citations fines i.csize i.college
```

Computing approximating function

Minimizing cross-validation criterion

Iteration 0: Cross-validation criterion = 30.26251

Computing average derivatives

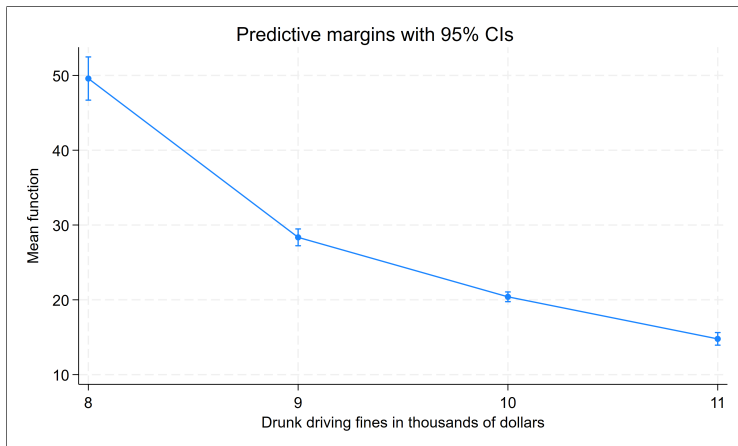
Cubic B-spline estimation	Number of obs	=	500
Criterion: cross-validation	Number of knots	=	1

citations	Effect	Robust std. err.	z	P> z	[95% conf. interval]	
fines	-7.787386	.2917941	-26.69	0.000	-8.359292	-7.215481
csz						
(Medium vs Small)	4.732592	.5087968	9.30	0.000	3.735368	5.729815
(Large vs Small)	10.91757	.5350892	20.40	0.000	9.868813	11.96632
col						
(College vs Not college)	6.514286	.5958949	10.93	0.000	5.346353	7.682218

Note: Effect estimates are averages of derivatives for continuous covariates and averages of contrasts for factor covariates.

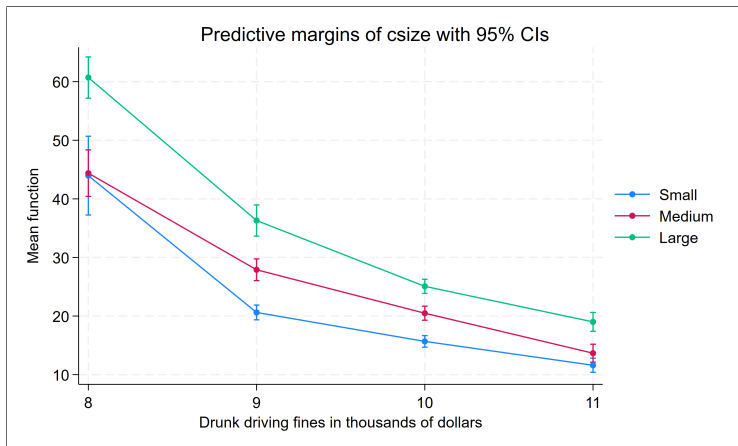
# Expected citations for different levels of fines

```
margins, at(fines=(8 9 10 11))
```



# The effect of fines for different levels of jurisdiction size

```
margins csize, at(fines=(8 9 10 11))
```



# makespline

Generates a set of variables that form a **prespecified basis**.

- Piecewise polynomial spline
- B-spline

You can then use these variables **directly in regressions**. Useful in **semiparametric methods**.

## Quick Start:

Generate a third-order B-spline basis from variables  $x_1$  and  $x_2$ ,

```
makespline bspline x1 x2
```

# Example 11: Creating a B-spline basis

```
sysuse auto
```

```
makespline bspline price
```

The screenshot shows the Stata Data Editor window with the 'auto' dataset loaded. The main window displays a list of variables: price, \_bsp\_1\_1, \_bsp\_1\_2, \_bsp\_1\_3, \_bsp\_1\_4, and \_bsp\_1\_5. The 'Variables' list on the right shows the following variables and their properties:

Variable	Label	Type	Format	Value Label
price	Price	int	%8.0gc	
_bsp_1_1	B-spline basis term 1 fo...	double	%10.0g	
_bsp_1_2	B-spline basis term 2 fo...	double	%10.0g	
_bsp_1_3	B-spline basis term 3 fo...	double	%10.0g	
_bsp_1_4	B-spline basis term 4 fo...	double	%10.0g	
_bsp_1_5	B-spline basis term 5 fo...	double	%10.0g	
make	Make and model	str18	%-18s	
mpg	Mileage (mpg)	int	%8.0g	
rep78	Repair record 1978	int	%8.0g	

The 'Properties' window on the right shows the following details for the variable 'price':

- Variables:** Name (price), Label (Price), Type (int), Format (%8.0gc), Value label, Notes.
- Data:** Frame (default), Filename (auto.dta), Label (1978 automobile data), Notes.
- Variables:** 17
- Observations:** 74
- Size:** 6.00K

## Using `makespline` in Semiparametric Methods

We will use simulated data for this example:

$$y = 3x_1 + 3 \sin(3(x_2 - x_3)) + \varepsilon$$

As researchers, we don't know this specific functional form.

However, we assume a semiparametric model structure:

$$y = \beta_1 x_1 + g(x_2, x_3) + \varepsilon$$

Our parameter of interest is  $\beta_1$ . Here,  $g(\cdot)$  is a nuisance parameter.

# Step 1: Create the B-spline basis function

- use <https://www.stata-press.com/data/r18/splines>
- makespline bspline x2 x3, knots(8)

Stata Data Editor (Browse) - [splines]

File Edit View Data Tools

1[1] - .6660838

	x1	x2	x3	gx	e	y	_bsp_1_1	_bsp_1_2	_bsp_1_3	_bsp_1_4	_bsp_1_5	_bsp_1_6
1	-.0098363	-.6303378	.2221587	-1.583013	-.4785688	-1.071334	0	0	.00282166	.39538540	.58851465	.05127825
2	.2034555	.1901777	1.035121	-1.440541	-.1022484	1.988577	0	0	0	0	0	.00221641
3	-.5158355	-.2234488	-.0244974	-1.68813	-.4402982	-.8739348	0	0	0	0	.03941367	.5851080
4	.9774078	1.382372	.3455514	.0603785	11.32719	11.48834	0	0	0	0	0	0
5	1.2355500	1.209281	-.5681133	2.537394	-1.763282	5.067212	.05980011	.42528647	.42834281	.08947062	0	0
6	.4593257	-.6045555	.225945	.5307749	2.487091	1.511623	0	.03353651	.24994128	.885213	.05128614	0
7	-.4852438	-.0628554	.1590182	-.4015542	2.85522	1.711306	0	.00553943	.1746567	.72935522	.09944885	0
8	-.8410513	1.300768	.5549493	1.972259	-.305399	2.056165	.0845715	.46405432	.38296594	.06780623	0	0
9	.0899468	1.207115	-.4323741	2.809523	1.542085	1.875412	0	0	0	0	0	0
10	2.182855	1.982184	.0334829	-1.430589	2.185232	1.184862	0	0	0	0	0	0
11	1.872006	-.357864	-.603853	2.017756	-.9681822	9.725593	0	0	0	.0055526	.2582525	.04414740
12	.3993822	1.007385	-.4867571	2.988879	7.728785	8.91699	0	.05388696	.88011864	.89528611	.00098209	0
13	1.239301	-.7547539	-.6805128	-.3895028	3.541662	4.774268	0	.03915666	.61772811	.33994181	.03317286	0
14	1.089707	1.019357	1.088197	-.670889	6.171591	8.893824	0	0	0	0	0	0
15	1.127881	1.122267	1.67718	2.586079	2.068117	2.848102	.00224405	.18296275	.55283127	.25282192	0	0
16	-.0031965	1.189125	.878285	1.652332	6.522084	8.503025	0	0	0	0	0	0
17	1.58669	.3262983	-1.048936	2.488534	3.210885	2.012272	0	0	0	0	0	0
18	1.231102	1.170859	-.289257	2.818295	2.822703	-.8966025	0	0	0	0	0	.00028853
19	7.148838	-.3047527	.0870202	2.835122	5.180694	7.496523	0	0	0	3.888607	.18083439	.06952104
20	-.3828277	.147826	-.453363	2.868893	3.542481	6.295881	0	0	0	0	0	.00371285
21	.5487259	0.048841	1.267137	2.462133	3.028167	10.14887	0	0	0	0	0	.05225891
22	.9489066	.0494317	2.159828	-.1774	2.119032	2.326562	0	.02738176	.4382182	.55545689	.00828652	0
23	-1.38989	.7007836	-.8688141	.8478326	2.065478	2.228716	0	0	.06162548	.67142788	.20935014	.00098649
24	1.135248	-.4805596	.8049548	.0347988	.3228181	6.822362	0	0	0	.02281852	.48510136	.46325747
25	.4162784	.895101	1.923957	-.1648838	2.313963	1.768677	0	0	0	0	0	0
26	-.7074443	.3545376	-.4044093	1.130014	2.646551	2.845889	0	0	0	0	0	0
27	1.888755	-.302955	-.6847363	2.868913	2.781032	14.4404	0	0	0	.00832665	.27153657	.63754555
28	-.3321472	1.49118	1.087834	2.532982	3.138788	3.384822	.45182113	.4893196	.08382408	.04833522	0	0
29	.0783834	.0254480	-.2010218	2.63820	2.286891	2.431711	0	.0121440	.3351729	.63188643	.02086567	0
30	.8759188	.2003304	-.3775734	2.722489	1.353158	1.742536	0	0	0	0	0	0
31	3.047812	.698922	.3200772	2.274573	7.104568	21.52258	0	0	0	0	0	0
32	-1.18735	-.18104	-1.158848	.6111985	3.778881	7.730725	0	0	0	0	.01482443	.48824888
33	-.3218362	.5309096	-.4620172	.8579376	4.216258	-1.224457	0	0	0	0	0	0

Variables

Filter variables here

Name	Label	Type	Format	Value l
x1		float	%9.0g	
x2		float	%9.0g	
x3		float	%9.0g	
gx		float	%9.0g	
e		float	%9.0g	
y		float	%9.0g	
_bsp_1_1	B-spline basis term 1 fo...	double	%10.0g	
_bsp_1_2	B-spline basis term 2 fo...	double	%10.0g	
_bsp_1_3	B-spline basis term 3 fo...	double	%10.0g	

Variables Snapshots

Properties

Variables

Name	Label	Type	Format	Value label	Notes
x1		float	%9.0g		

Data

File Name

Label

Format

Variables

Observations

Size

View: 30 Order: Dataset Filter: 5,000 Filter: Off Mode: Browse G-D 10.04

Elements of the base are stored in macro `r(regressors)`



## Step 2: Use LASSO to choose elements in the base

With all the interactions we have 168 new regressors. We'll use `poregress` to select from the 168 covariates using LASSO.

- `poregress y x1, controls('r(regressors)')`

```
. poregress y x1, controls('r(regressors)')
```

```
Estimating lasso for y using plugin
```

```
Estimating lasso for x1 using plugin
```

```
Partialing-out linear model      Number of obs      =      5,000
                                Number of controls      =      168
                                Number of selected controls =      19
                                Wald chi2(1)              =    3535.78
                                Prob > chi2               =      0.0000
```

		Robust				
	y	Coefficient	std. err.	z	P> z	[95% conf. interval]
	x1	2.951242	.049632	59.46	0.000	2.853965 3.048519

Note: Chi-squared test is a Wald test of the coefficients of the variables of interest jointly equal to zero. Lassos `select controls` for model estimation. Type `lassoinfo` to see number of selected variables in each lasso.

# Outline

- 1 The appeal of nonparametric methods
- 2 Estimating probability densities
- 3 Estimating conditional expectations
  - Kernel Regression
  - Series Approximations
- 4 Advantages and Limitations of nonparametric methods

# Limitations of Kernel Regression and Series Approximations

**1. Computational Burden:** CPU time rises fast with  $n$  and  $k$ .

Problem is acuter for series approximations.

- Kernel regression.
- Option `nointeract`.

**2. Curse of Dimensionality:** When  $k$  is large, we need a very big  $n$  to have informative CIs.

- Try to have a lot of data per covariate.
- Semiparametric methods.

## Conclusion: Nonparametric Methods in Stata

1. Powerful tool to explore the **relationships between variables**.
  - **No assumptions on functional form** required.
  - No risk of **misspecification**.
2. Easily implementable in Stata
  - `npregress`: Estimate  $g(\cdot)$ , effects, and make predictions.
  - `margins`: explore  $\hat{g}(\cdot)$  and ask interesting questions.
3. Be mindful of limitations:
  - **Computational cost**.
  - **Curse of dimensionality**.

## Where to learn more?

1. Stata documentation:  
<https://www.stata.com/features/documentation/>
2. YouTube channel: <https://www.youtube.com/user/statacorp>
3. Send an email to our tech support team:  
[tech-support@stata.com](mailto:tech-support@stata.com).
4. The `help` command.

Thank you!