

ACTUALIZACIÓN DE PROGRAMAS MEJORADOS DE STATA PARA LA SUAVIZACIÓN NO PARAMÉTRICA DE DATOS.

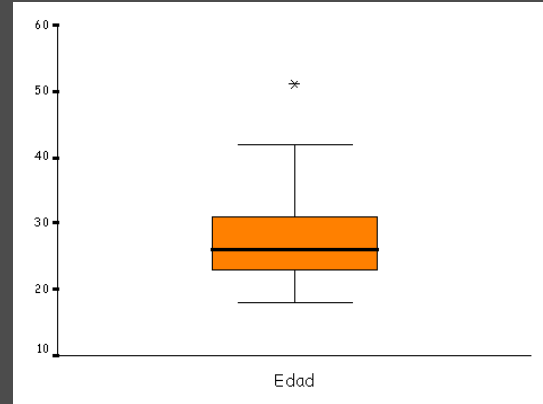
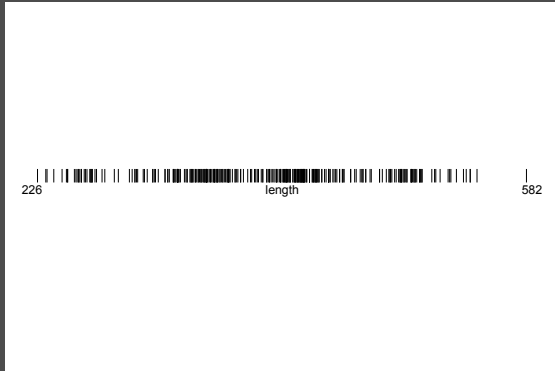
Mosqueda Romo Néstor A.

Salgado Ugarte Isaías H.

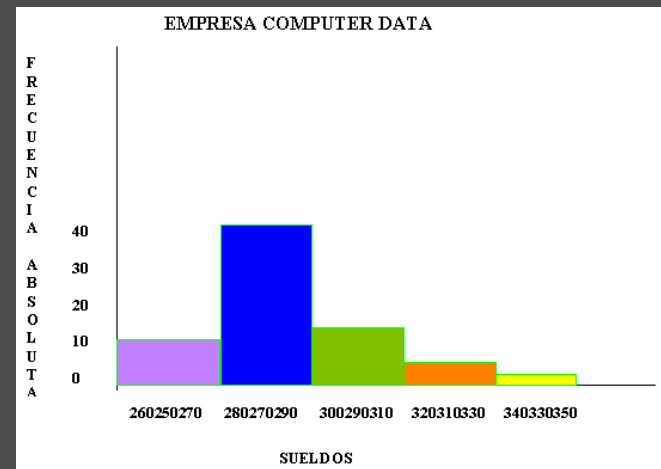
Laboratorio de Biometría y Biología Pesquera

FES Zaragoza, UNAM

Existen varios procedimientos estadísticos para mostrar la distribución de un lote de datos univariado.



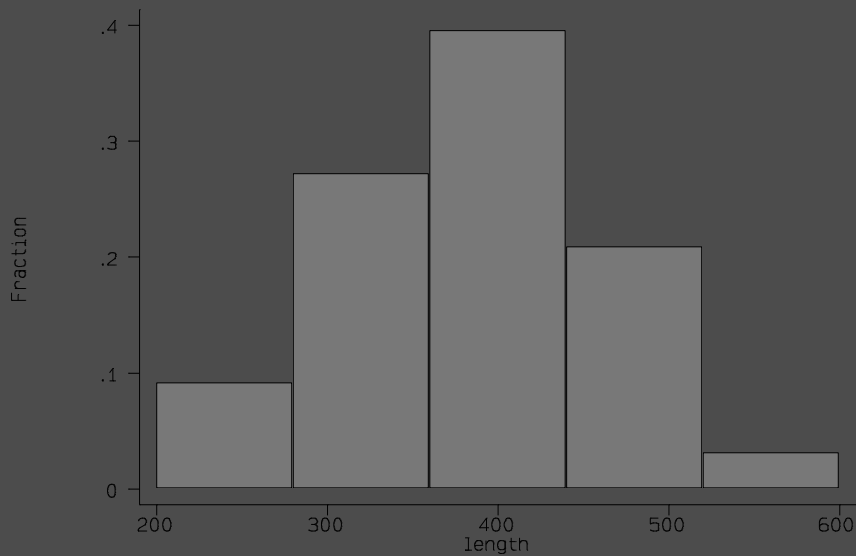
| Tallo | Hoja |
|-------|-----------|
| 3 | |
| 3 | 5 6 8 |
| 4 | 0 2 2 4 |
| 4 | 5 5 7 8 9 |
| 5 | 0 0 0 |
| 5 | |



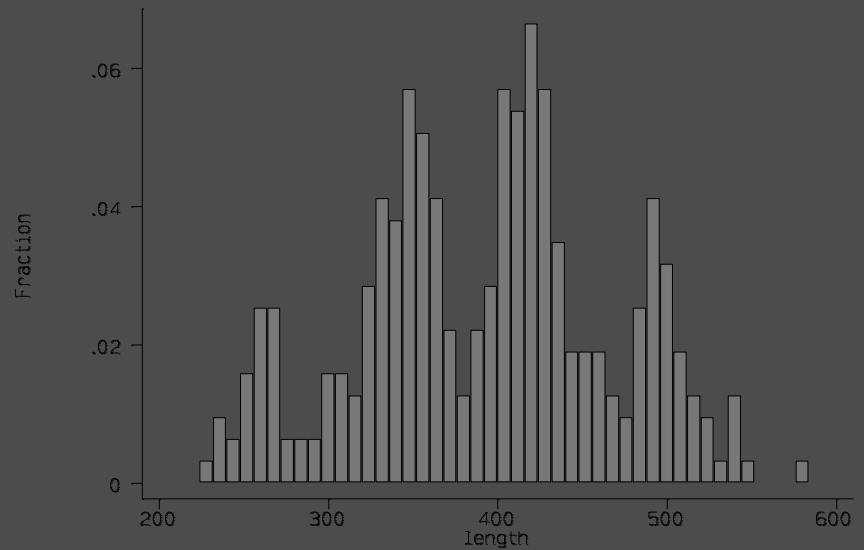
Histograma

- ⦿ El método mas ampliamente utilizado para representar la forma de una función de densidad de probabilidad es el histograma.

Histogramas



Histograma con 5 intervalos fig. 1.1



Histograma con 50 intervalos fig. 1.2

Desventajas de los histogramas

- ⦿ Dependen del punto de origen de los intervalos
- ⦿ Dependen de la amplitud y el número de intervalos
- ⦿ Discontinuidad
- ⦿ Amplitud fija de intervalo

Lo anterior ha motivado el desarrollo de otros métodos (estimadores de densidad por kernel)

Ventajas de los estimadores de densidad por kernel (EDK's)

- No dependen del punto de origen (estimación centrada en cada valor)
- Elimina la discontinuidad (estimación centrada de cada valor y usa un cambio gradual en la función ponderal en lugar de la función rectangular).
- Se puede aplicar una amplitud variable de intervalo

Desventajas de los EDK's

- Gran número de operaciones algebraicas

Enfoques para solucionar este problema

- ⦿ Estimación discretizada
- ⦿ Estimación por el método HDP-PPPR

Introducción I

- A continuación voy a presentar versiones actualizadas a Stata 11 de programas ado con rutinas mejoradas para estimación de densidad por kernel.
- Se incluyen:
 - Trazas de densidad
 - Reglas prácticas para el número y amplitud de intervalo en histogramas y polígonos de frecuencia y amplitud de banda en la estimación de densidad por kernel
 - Estimadores de densidad por kernel con amplitud de banda fijo y variable directos y discretizados
 - Buscador de amplitud crítica de banda
 - Prueba bootstrap para evaluación no paramétrica de multimodalidad

Introducción II

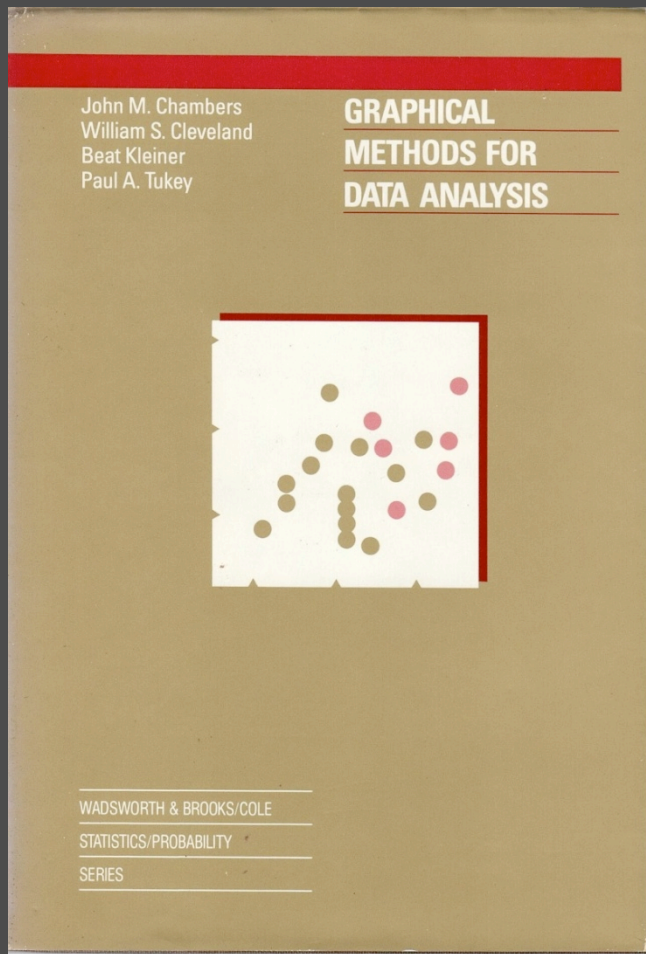
- Estas versiones actualizadas son de programas ya mejorados, simples, y más versátiles presentados anteriormente para el cálculo de diversos estimadores de densidad por kernel.

Trazas de densidad I

Trazas de densidad presentadas en:

- Chambers, J.M., W.S. Cleveland, B. Kleiner y P.A. Tukey (1983) Graphical Methods for Data Analysis. Wadsworth & Brooks/Cole, Cap. 2: 9-46.

Trazas de densidad II



Back row: W. S. Cleveland, Paul A. Tukey
Front row: Beat Kleiner, John M. Chambers

Trazas de densidad III

- ◎ **Los programas ado incluyen:**
 - **boxdetr1 (función ponderal cuadrada) usando un algoritmo directo**
 - **boxdet21 (función ponderal cuadrada) usando un algoritmo discretizado considerando 50 puntos uniformemente espaciados desde el valor mínimo hasta el máximo**
 - **cosdetr1 (función ponderal coseno) usando un algoritmo discretizado**
 - **dentrac1 (boxcar y función ponderal coseno) implementado con un procedimiento discretizado**

Las actualizaciones para el primer programa se presentan a continuación

- 5.- if `_rc~=0` {
- 6.- di "syntax: boxdetra
varname iwidth detravar"
- 7.- exit}
-
- 11.- gen `nuobs'=_result(1)
- 17.- noi di "WORKING WITH
EACH VALUE. PLEASE BE
PATIENT"
- 21.- replace `trace'=_result(1)/
(`2'*`nuobs') if `_n==`count'`
- 26.- di "DONE. THANKS FOR
YOUR PATIENCE"

boxdetra.ado versión 3.0

- 5.- if `_rc!=0` {
- 6.- di **as error** "syntax: boxdetr1
varname iwidth detravar"
- 7.- exit
- }**
- 11.- gen `nuobs'=**r(N)**
- 17.- noi di **as result** "WORKING
WITH EACH VALUE. PLEASE
BE PATIENT"
- 21.- replace `trace'=**r(N)**/
(`2'*`nuobs') if `_n==`count'`
- 26.- di **as result** "DONE.
THANKS FOR YOUR
PATIENCE"

boxdetr1.ado versión 11.0

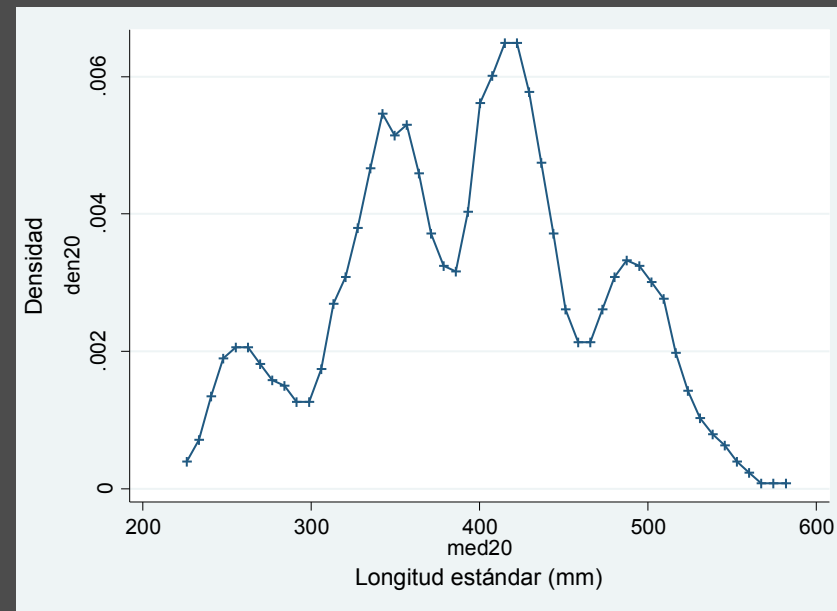
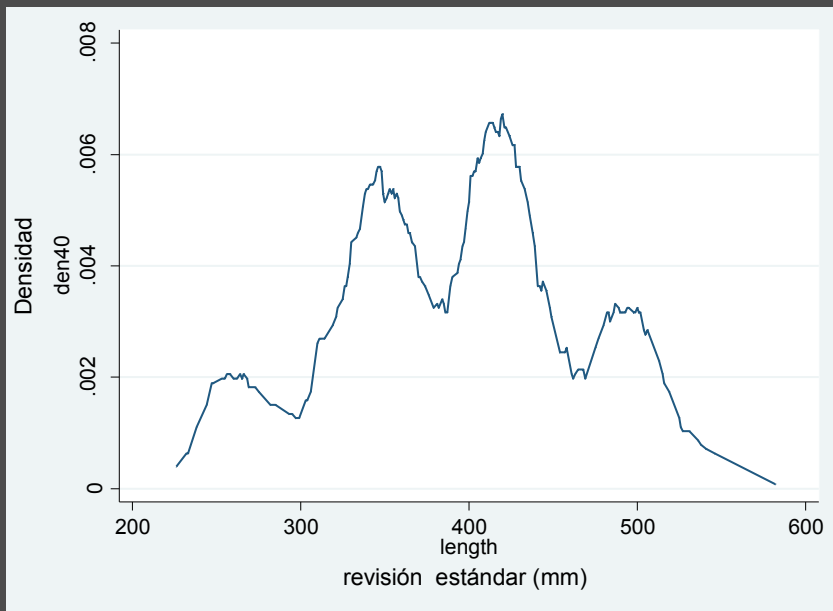
Las actualizaciones para el segundo programa se presentan a continuación

- 2.- capture confirm existence `1'
 - 5.- if _rc~=0 {
 - 6.- di "syntax: boxdetra varname
iwidth detravar midpoivar"
 - 7.- exit}
 - 10.- gen `nuobs'=_N
 - 11.- gen `nuobs'=_result(1)
 - 17.- noi di "WORKING WITH EACH
VALUE. PLEASE BE PATIENT"
 - 21.- replace `trace'=_result(1)/
(`2'*`nuobs') if _n==`count'
 - 26.- di "DONE. THANKS FOR
YOUR PATIENCE"
 - 41.-noi di "DONE. THANKS FOR
YOUR PATIENCE"
- 2.- capture {
 - 5.- if _rc!=0 {
 - 6.- di **as error** "syntax: boxdet21
varname iwidth detravar midpoivar"
 - 7.- exit
 - 10.- **summ `1'**
 - 11.- gen `nuobs'= **r(N)**
 - 17.- noi di **as result** "WORKING WITH
EACH VALUE. PLEASE BE PATIENT"
 - 21.- replace `trace'= **r(N)**/(`2'*`nuobs') if
_n==`count'
 - 26.- di **as result** "DONE. THANKS FOR
YOUR PATIENCE"
 - 41.-noi di **as result** "DONE. THANKS
FOR YOUR PATIENCE"

boxdetr2.ado versión 3.0

boxdet21.ado versión 11.0

Trazas de densidad IV



boxdetr1.ado

boxdet21.ado

Trazas de densidad VI

- Estos estimadores simples de la densidad eliminan la discontinuidad local de los intervalos en los histogramas, pero siguen teniendo un poco de ruido. Una razón para esta rugosidad es la forma rectangular de la función de peso con forma cuadrada. Para suavizar aún más la estimación de densidad podemos considerar una función ponderal coseno.

Las actualizaciones para este tercer programa se presentan a continuación

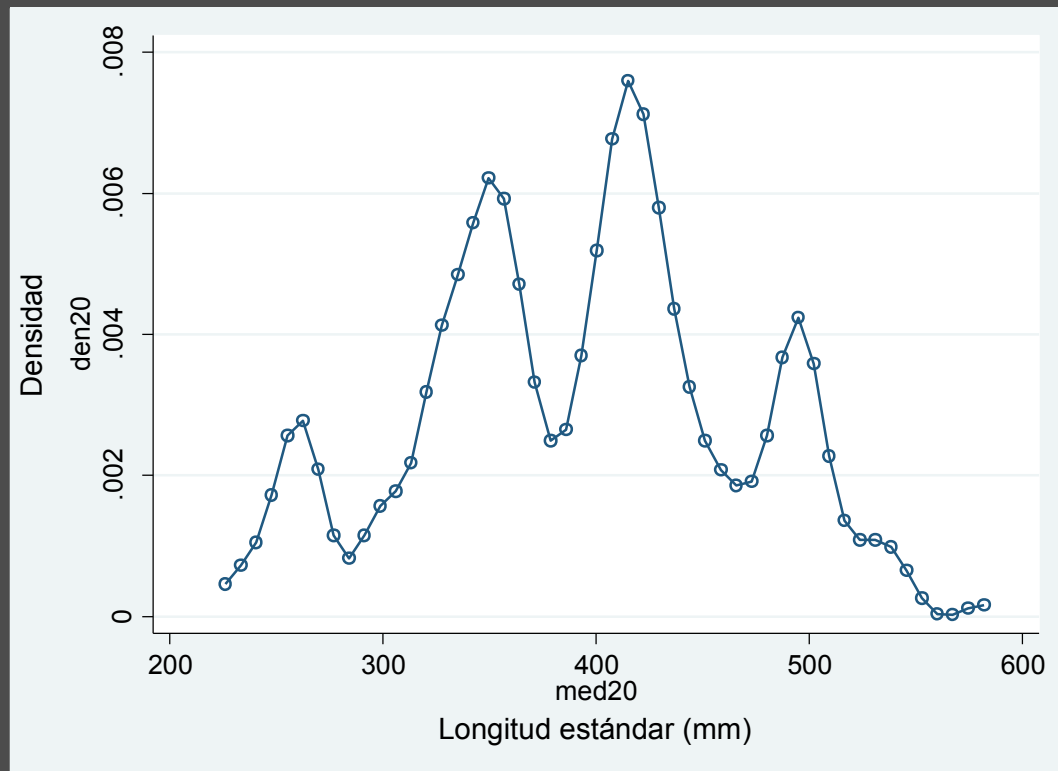
- 5.- if `_rc~=0` {
- 6.- di "syntax: cosdetra varname iwidth detravar midpoivar"
- 7.- exit}
- 10.- gen ``nuobs'=_N`
- 11.- gen ``nuobs'=_result(1)`
- 17.- noi di "WORKING WITH EACH VALUE. PLEASE BE PATIENT"
- 21.- replace ``trace'=_result(1)/(`2'*`nuobs')` if `_n==`count'`
- 26.- di "DONE. THANKS FOR YOUR PATIENCE"
- 41.- noi di "DONE. THANKS FOR YOUR PATIENCE"

cosdetra.ado versión 3.0

- 5.- if `_rc!=0` {
- 6.- di **as error** "syntax: cosdetr1 varname iwidth detravar midpoivar"
- 7.- exit
- 10.- **summ `1'**
- 11.- gen ``nuobs'= r(N)`
- 17.- noi di **as result** "WORKING WITH EACH VALUE. PLEASE BE PATIENT"
- 21.- replace ``trace'= r(N)/(`2'*`nuobs')` if `_n==`count'`
- 26.- di **as result** "DONE. THANKS FOR YOUR PATIENCE"
- 41.- noi di **as result** "DONE. THANKS FOR YOUR PATIENCE"

cosdetr1.ado versión 11.0

Trazas de densidad VII



Las actualizaciones para el cuarto programa se presentan a continuación

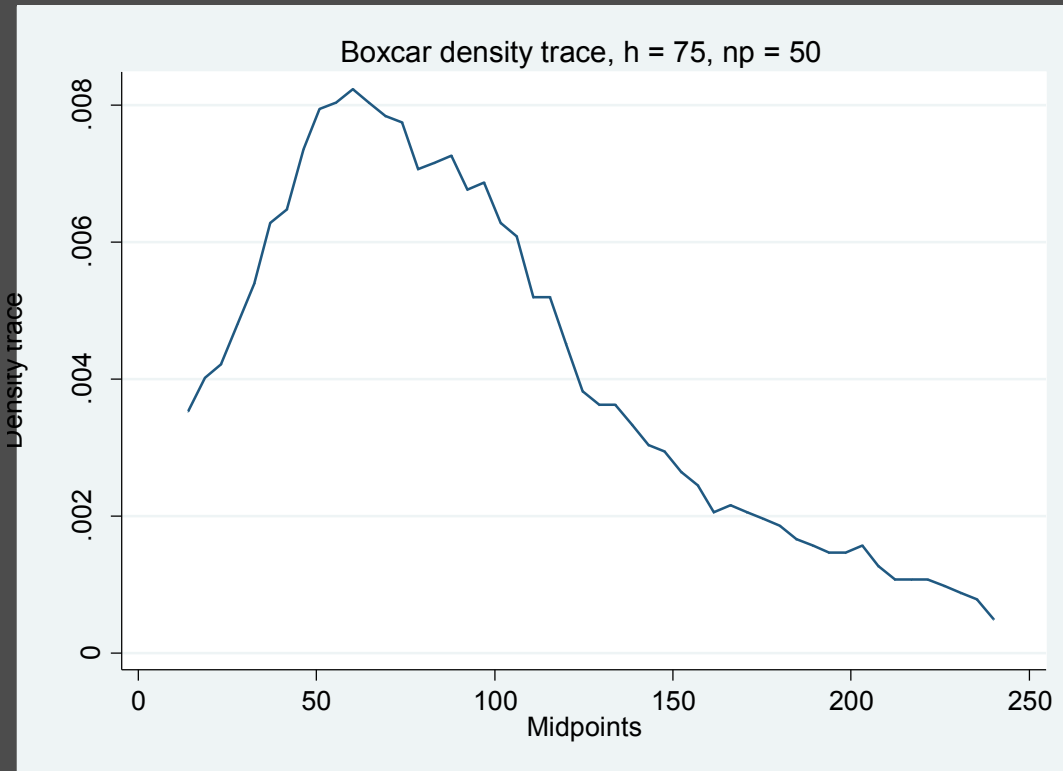
- 2.- syntax varlist(min=1 max=1) [if] [in] , Hval(real) Fcode(int) [NPoint(int 50) Gen(str) noGraph T1title(str) Symbol(str) Connect(str) *]
- 17.- if r(N) == 0 {error 2000}
- 57.- if "`graph'" ~= "nograph" {
- 67.- if "`symbol'" == "" { local symbol "." }
- 71.- graph `fwy' `midval', `options' t1("`t1title'") s(`symbol') c(`connect')
- 73.- if "`gen'" ~= "" {

dentrace.ado versión 6.0

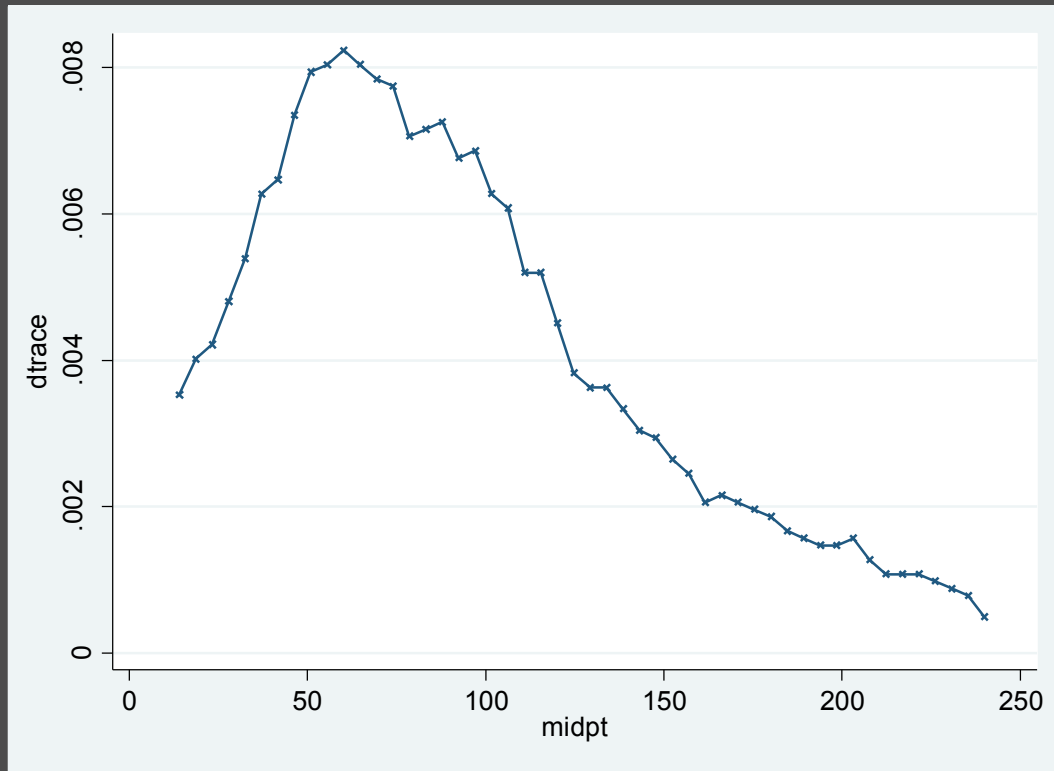
- 2.- syntax varlist(min=1 max=1) [if] [in] , Hval(real) Fcode(int) [NPoint(int 50) Gen(str) noGraph T1title(str) **mSymbol(str)** Connect(str) *]
- 17.- if r(N) == 0 {
- error 2000**
- }**
- 57.- if "`graph'" != "nograph" {
- 67.- if "**msymbol**" == "" {
- 71.- **scatter** `fwy' `midval', `options' t1("`t1title'") **ms(`msymbol')** c(`connect')
- 73.-if "`gen'" != "" {

dentrac1.ado versión 11.0

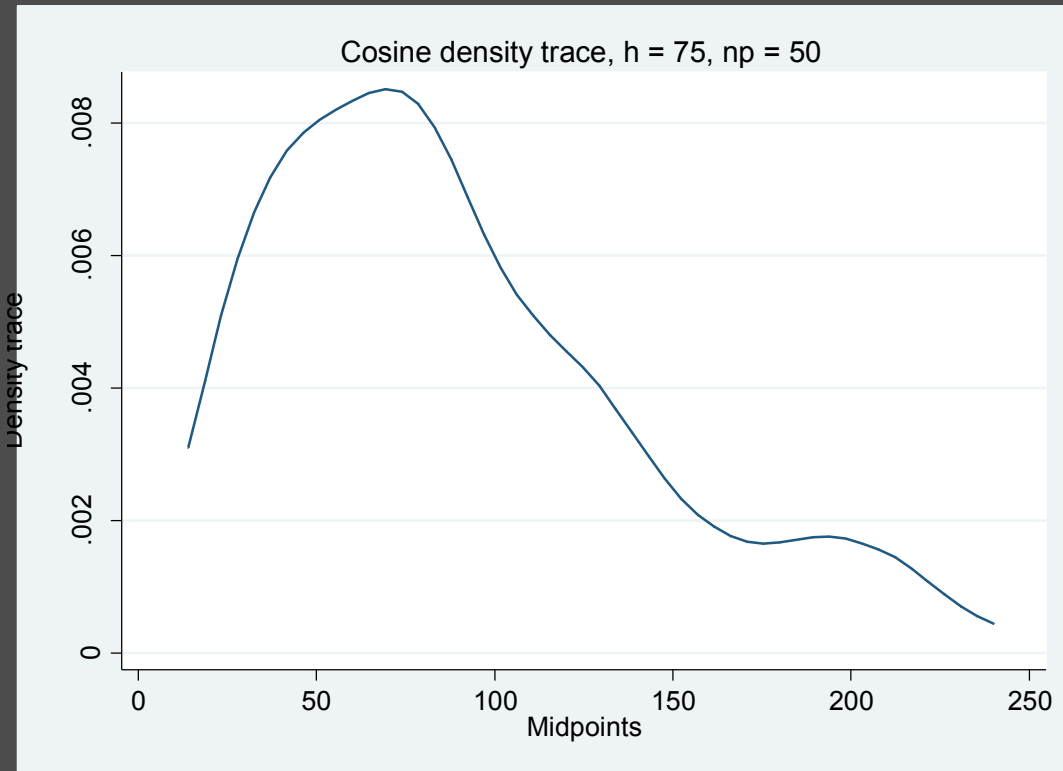
dentrac1 ozone, h(75) f(1) gen(dtrace midpt)



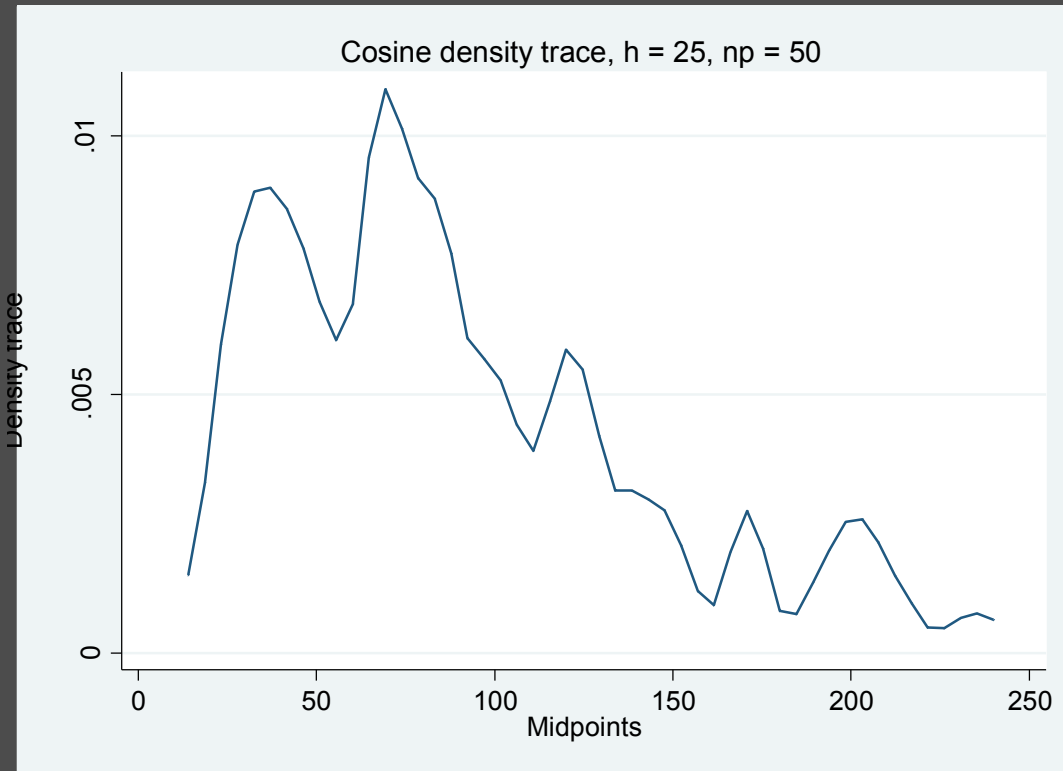
scatter dtrace midpt , c(l) ms(x)



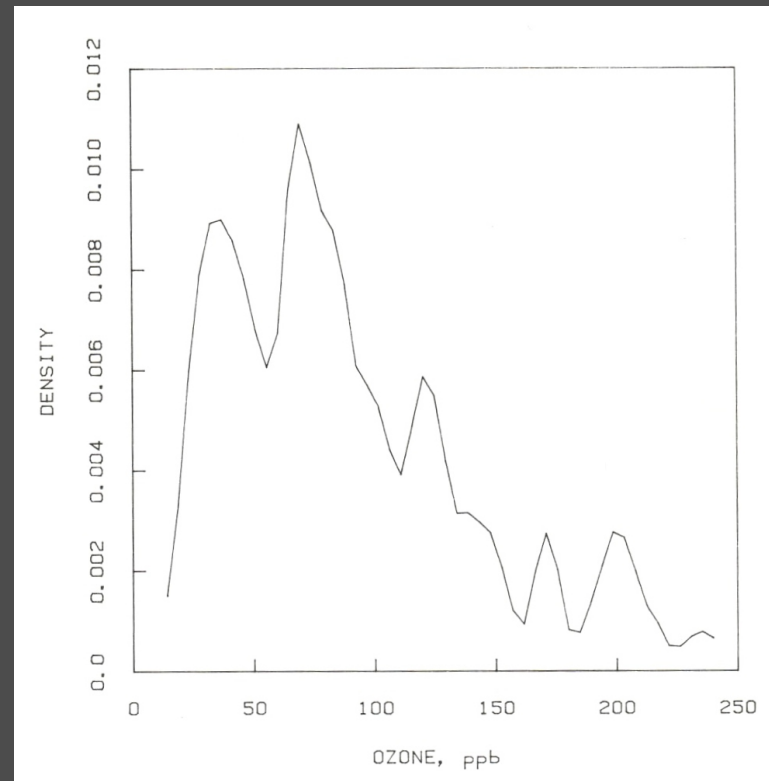
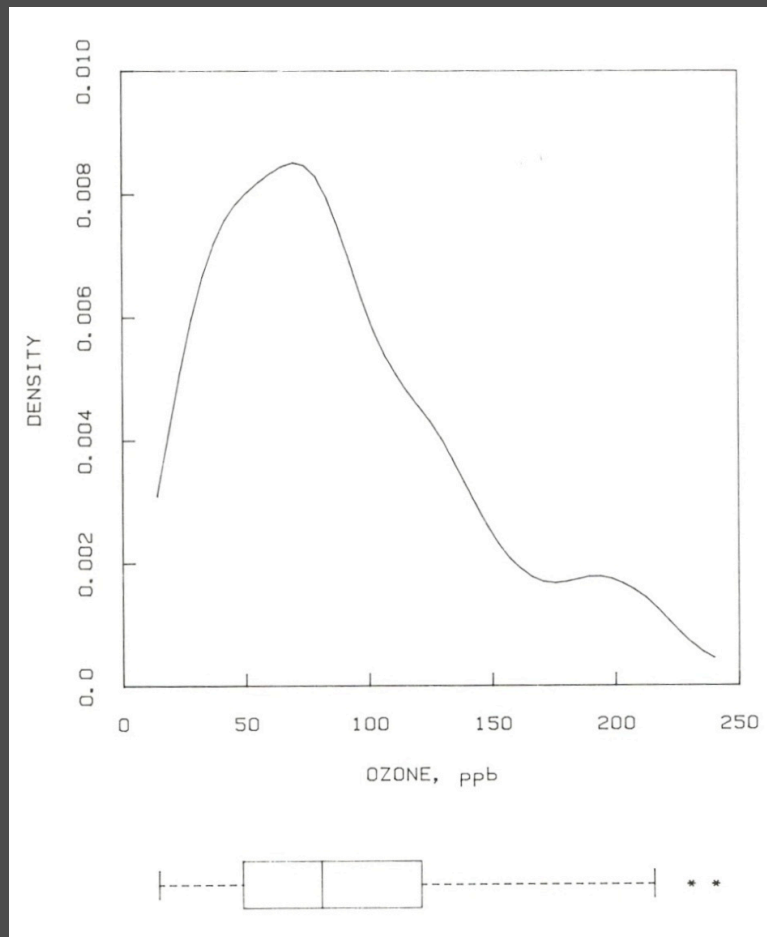
dentrac1 ozone, h(75) f(2) gen(dtracec midptc)



dentrac1 ozone, h(25) f(2) gen(dtraced midptd)



Figuras. 2.20 y 2.21 Chambers, et al. 1983



Elección de la amplitud de banda de banda I

- En la estimación de densidad por kernel. Algo muy importante son los pasos para elegir la amplitud de banda. El programa `bandw1.ado`, calcula mediante un conjunto de reglas la elección adecuada del número o ancho de intervalo (histogramas o polígonos de frecuencia) o de la amplitud de banda (estimadores de densidad por kernel).

Las actualizaciones para el programa bandw se presentan a continuación

- 2.- syntax varlist(min=1 max=1) [if] [in][, Kercode(integer 6)]

- 8.- if r(N) == 0 {error 2000}

bandw.ado versión 7.0

- 2.- syntax varlist(max=1 numeric) [if] [in][, Kercode(integer 6)]

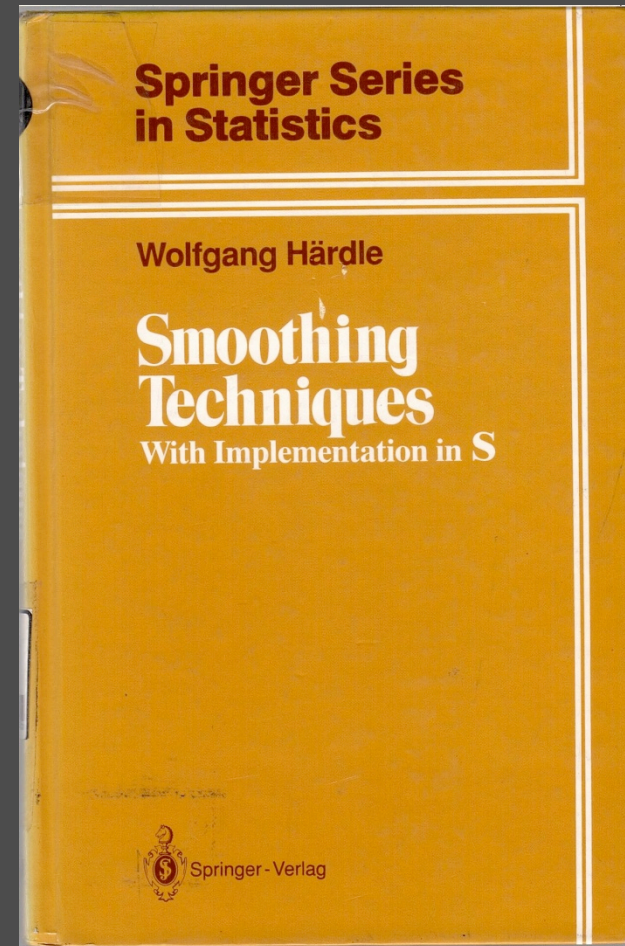
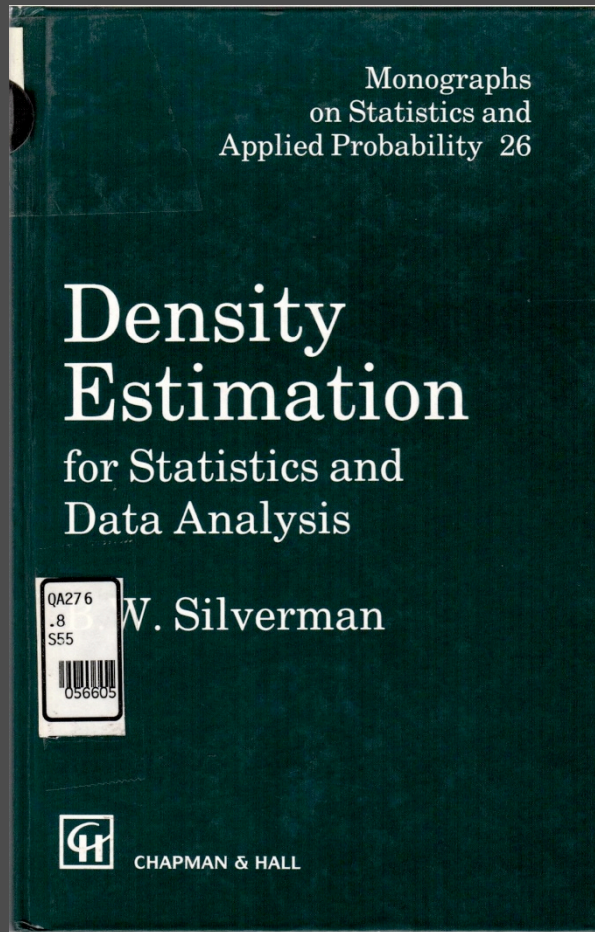
- 8.- if r(N) == 0 {
 - error 2000
 - }

bandw1.ado versión 11.0

Elección del ancho de banda II

- Esta versión mejorada de `bandw1.ado` permite la elección del kernel y el cálculo del ancho de banda sobresuavizado y óptimo utilizando los factores de conversión incluidos en Härdle (1991), Scott (1992) y Salgado-Ugarte et al. (1995b).
- Todas las reglas se basan en ecuaciones incluidas en Silverman (1986), Fox (1990), Härdle (1991), Scott (1992) y Salgado-Ugarte (2002).

Elección del ancho de banda III



Elección del ancho de banda IV

Algunos factores de conversión para kernels sobreesuavizados

| A fila/desde columna | Uniforme | Triangular | Epanech. | Cuártico | Triponderado | Coseno | Gaussiano |
|----------------------|----------|------------|----------|----------|--------------|--------|-----------|
| Uniforme | 1.000 | 0.715 | 0.786 | 0.663 | 0.584 | 0.761 | 1.740 |
| Triangular | 1.398 | 1.000 | 1.099 | 0.927 | 0.817 | 1.063 | 2.432 |
| Epanech. | 1.272 | 0.910 | 1.000 | 0.844 | 0.743 | 0.968 | 2.214 |
| Cuartico | 1.507 | 1.078 | 1.185 | 1.000 | 0.881 | 1.146 | 2.623 |
| Triponderado | 1.711 | 1.225 | 1.345 | 1.136 | 1.000 | 1.302 | 2.978 |
| Coseno | 1.315 | 0.941 | 1.033 | 0.872 | 0.768 | 1.000 | 2.288 |
| Gaussiano | 0.575 | 0.411 | 0.452 | 0.381 | 0.336 | 0.437 | 1.000 |

Transformación de kernel en fila a kernel en columna

Elección del ancho de banda (por default)

```
. use catfilen  
. bandw1 bodlen
```

Some practical number of bins and binwidth-bandwidth rules
for univariate density estimation using histograms,
frequency polygons (FP) and kernel density estimators

```
Sturges' number of bins = 10.3242  
Oversmoothed number of bins <= 10.8633
```

```
-----  
FP oversmoothed number of bins <= 8.6026
```

```
Scott's optimal Gaussian binwidth = 20.1301  
Freedman-Diaconis optimal robust binwidth = 14.8454  
Terrell-Scott's oversmoothed binwidth >= 15.5759  
Oversmoothed homoscedastic binwidth >= 21.4472  
Oversmoothed robust binwidth >= 19.3212
```

```
-----  
FP optimal Gaussian binwidth = 29.2728  
FP oversmoothed binwidth >= 31.7236
```

Gaussian kernel (6)

```
Silverman's optimal bandwidth = 11.7230  
Haerdle's 'better' optimal bandwidth = 13.8071  
Scott's oversmoothed bandwidth = 15.5759
```

Elección del ancho de banda (cuártico)

```
. bandwl bodlen, k(4)
```

```
Some practical number of bins and binwidth-bandwidth rules  
for univariate density estimation using histograms,  
frequency polygons (FP) and kernel density estimators
```

```
Sturges' number of bins = 10.3242  
Oversmoothed number of bins <= 10.8633
```

```
FP oversmoothed number of bins <= 8.6026
```

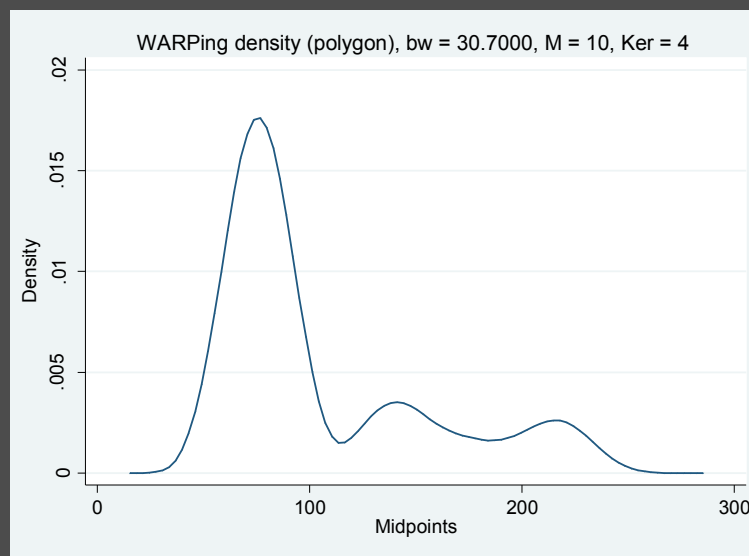
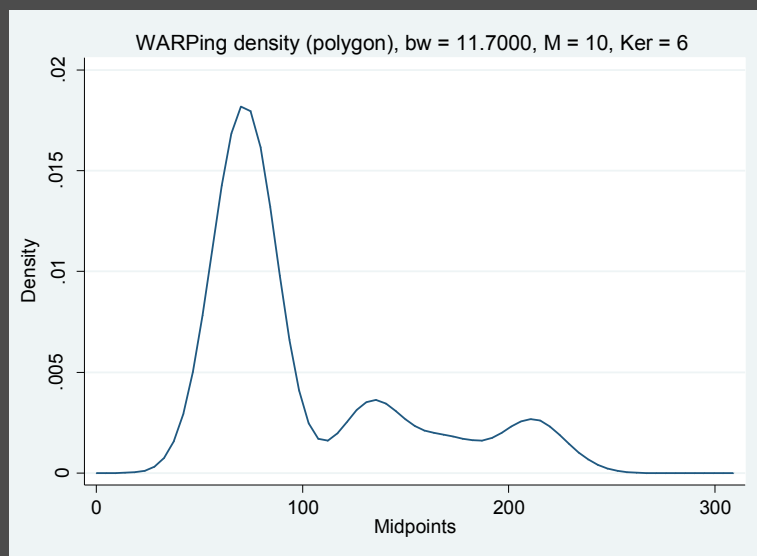
```
Scott's optimal Gaussian binwidth = 20.1301  
Freedman-Diaconis optimal robust binwidth = 14.8454  
Terrell-Scott's oversmoothed binwidth >= 40.8555  
Oversmoothed homoscedastic binwidth >= 21.4472  
Oversmoothed robust binwidth >= 19.3212
```

```
FP optimal Gaussian binwidth = 29.2728  
FP oversmoothed binwidth >= 31.7236
```

```
Quartic kernel (4)
```

```
Silverman's optimal bandwidth = 30.7494  
Haerdle's 'better' optimal bandwidth = 36.2160  
Scott's oversmoothed bandwidth = 40.8555
```

Elección de ancho de banda, estimadores óptimos, kerneles gaussiano (bw = 11.7) y cuártico (bw= 30.7)



Estimadores de densidad por kernel con ancho de banda fijo y variable directos y discretizados

⦿ Los programas ado incluyen:

- kernsi1 (función ponderal rectangular) usando un algoritmo discretizado
- kernep1 (función por kernel epanechnikov) usando un algoritmo discretizado considerando 50 puntos
- kerngau1 (función por kernel gaussiano) usando un algoritmo discretizado considerando 50 puntos
- adgaker1 (función por kernel gaussiano) con amplitud variable usando un algoritmo directo
- kerneld1 permite escoger la función ponderal (kernel) usando un algoritmo discretizado o interpolado

Las actualizaciones para el programa kernsim se presentan a continuación

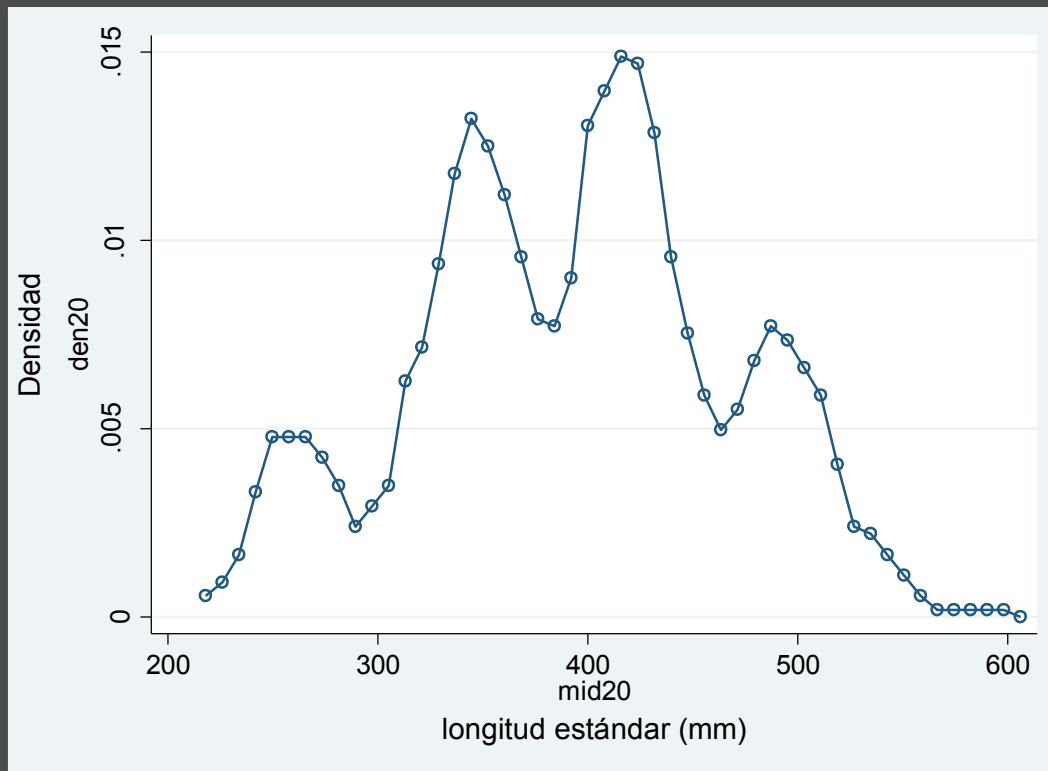
- 5.- if `_rc~=0` {
- 6.- di "syntax: kernsim
varname halfwidth densivar
midpoivar"
- 7.- exit}
- 10.- gen `nuobs'=_result(1)
- 20.- gen
`maxval'=_result(6)+`h'
- 21.- gen `minval'=_result(5)-
`h'
- 26.- di "DONE. THANKS FOR
YOUR PATIENCE"

- 5.- if `_rc!=0` {
- 6.- di **as error** "syntax: kernsi1
varname halfwidth densivar
midpoivar"
- 7.- exit
- }
- 10.- **summ `1'**
- 10.- gen `nuobs'=**r(N)**
- 20.- gen `maxval'=**r(max)**+`h'
- 21.- gen `minval'=**r(min)**-`h'
- 26.- di **as result** "DONE.
THANKS FOR YOUR
PATIENCE"

kernsim.ado versión 3.0

kernsi1.ado versión 11.0

kernsi1 length 20 den20 mid20
scatter den20 mid20, c(l) ms(Oh)



Las actualizaciones para el programa kernepa se presentan a continuación

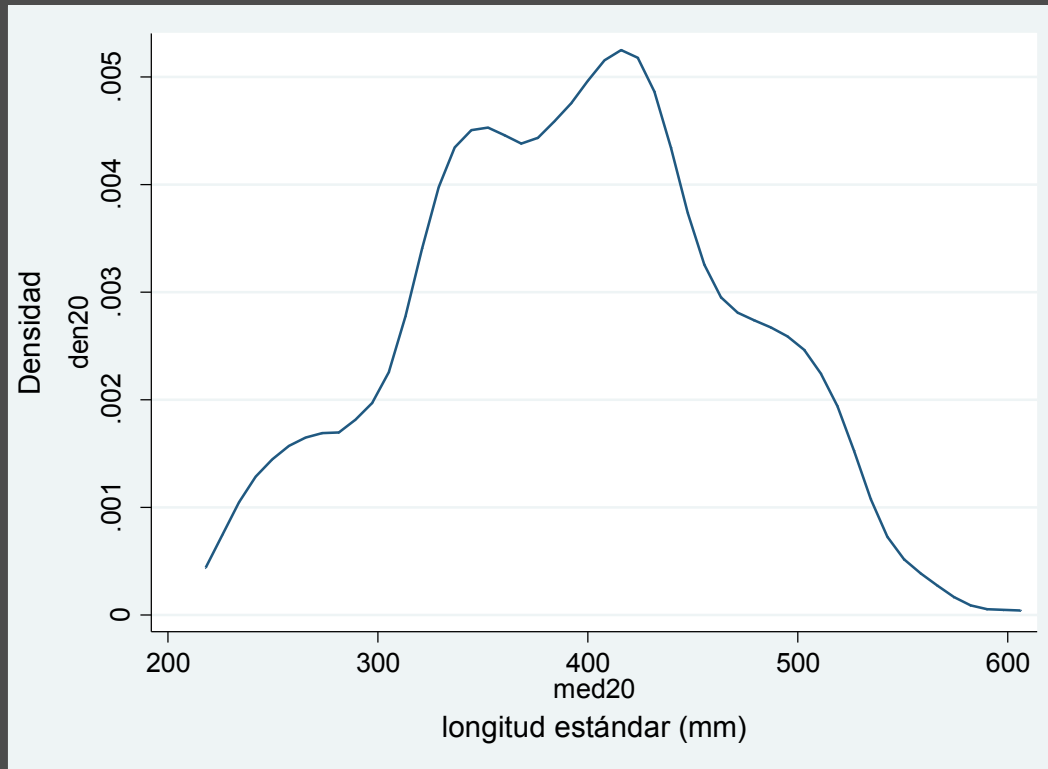
- 5.- if _rc~=0 {
- 6.- di "syntax: kernepa varname halfwidth densivar midpoivar"
- 7.- exit}
- 10.- gen `nuobs'=_result(1)
- 20.- gen `maxval'=_result(6)+`h'
- 21.- gen `minval'=_result(5)-`h'
- 26.- di "DONE. THANKS FOR YOUR PATIENCE"

kernepa.ado versión 3.0

- 5.- if _rc!=0 {
- 6.- di **as error** "syntax: kernep1 varname halfwidth densivar midpoivar"
- 7.- exit
- 10.- **summ `1'**
- 10.- gen `nuobs'= **r(N)**
- 20.- gen `maxval'= **r(max)**+`h'
- 21.- gen `maxval'= **r(min)**-`h'
- 26.- di **as result** "DONE. THANKS FOR YOUR PATIENCE"

kernep1.ado versión 11.0

kernel1 length 20 den20 mid20
scatter den20 mid20, c(l) ms(p)



Las actualizaciones para el programa kerngaus se presentan a continuación

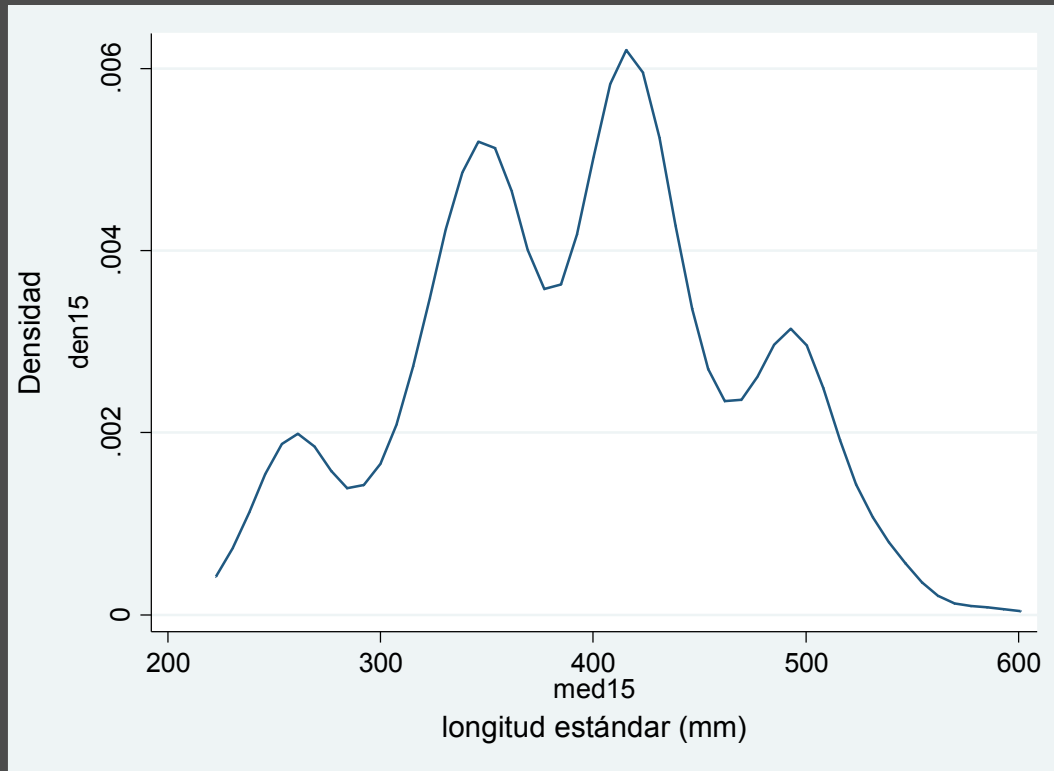
- 5.- if _rc~=0 {
- 6.- di "syntax: kerngaus varname halfwidth densivar midpoivar"
- 7.- exit}
- 10.- gen `nuobs'=_result(1)
- 20.- gen `maxval'=_result(6)+`h'
- 21.- gen `minval'=_result(5)-`h'
- 26.- di "DONE. THANKS FOR YOUR PATIENCE"

kerngaus.ado versión 3.0

- 5.- if _rc!=0 {
- 6.- di **as error** "syntax: kerngau1 varname halfwidth densivar midpoivar"
- 7.- exit
- 10.- **summ `1'**
- 10.- gen `nuobs'= **r(N)**
- 20.- gen `maxval'= **r(max)**+`h'
- 21.- gen `maxval'= **r(min)**-`h'
- 26.- di **as result** "DONE. THANKS FOR YOUR PATIENCE"

kerngau1.ado versión 11.0

```
kerngau1 length 15 den15 mid15  
scatter den15 med15, c(l) ms(p)
```



Las actualizaciones para el programa adgakern se presentan a continuación

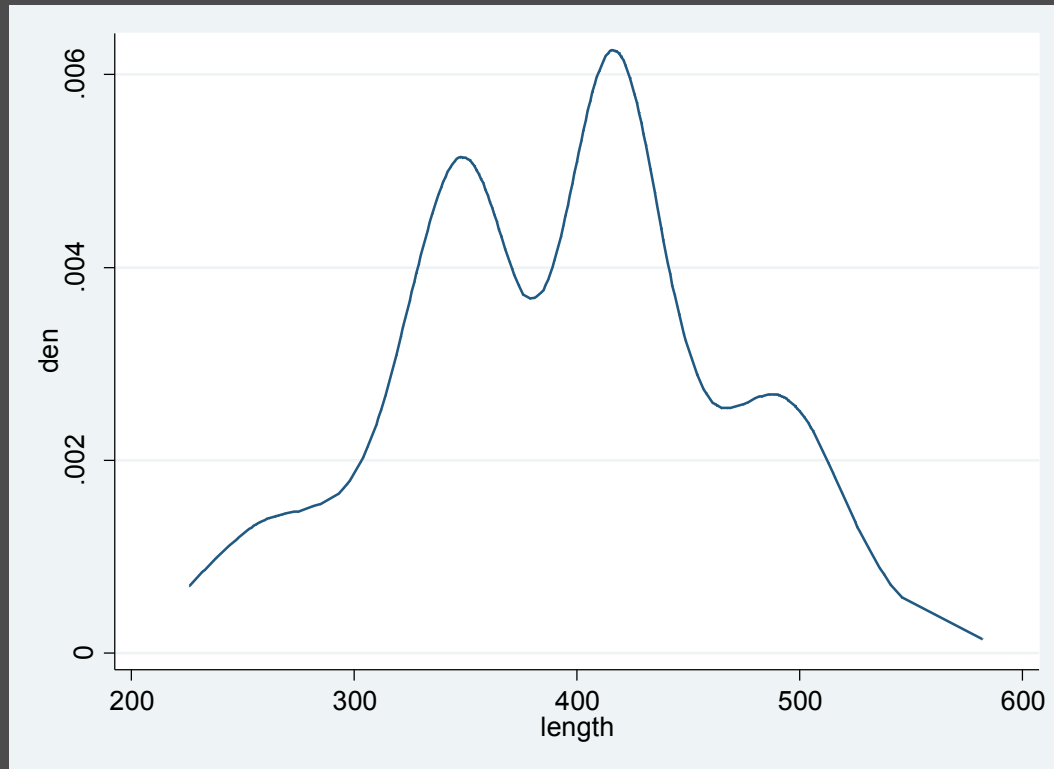
- 5.- if _rc~=0 {
- 6.- di "syntax: boxdetra
varname halfwidth densivar"
- 7.- exit}
- 10.- gen `nuobs'=_result(1)
- 19.- noi di "WORKING WITH
EACH VALUE. PLEASE BE
PATIENT"
- 43.- noi di "CALCULATING
ADAPTIVE VALUES. DON'T
DESPAIR"
- 55.- noi di "DONE. THANK FOR
YOUR PATIENCE"

adgakern.ado versión 3.0

- 5.- if _rc!=0 {
- 6.- di **as error** "syntax: boxdetra
varname halfwidth densivar"
- 7.- exit
- }**
- 10.- gen `nuobs'= **r(N)**
- 19.- noi di **as result** "WORKING
WITH EACH VALUE. PLEASE BE
PATIENT"
- 43.- noi di **as result**
"CALCULATING ADAPTIVE
VALUES. DON'T DESPAIR"
- 55.- noi di **as result** "DONE. THANK
FOR YOUR PATIENCE"

adgaker1.ado versión 11.0

adgaker1 length 20.18 den
scatter den length, c(l) ms(p)



Las actualizaciones para el programa kerneld se presentan a continuación

- 5.- local options "Bwidth(real 0) Kercode(integer 0) NPoint(integer 0) Gen(string) noGraph T1title(string) Symbol(string) Connect(string) *"
- 6.- di "syntax: boxdetra varname iwidth detravar"
- 7.- exit}
- 10.- gen `nuobs'=_result(1)
- 19.- noi di "WORKING WITH EACH VALUE. PLEASE BE PATIENT"
- 43.- noi di "CALCULATING ADAPTIVE VALUES. DON'T DESPAIR"
- 55.- noi di "DONE. THANK FOR YOUR PATIENCE"

kerneld.ado versión 3.0

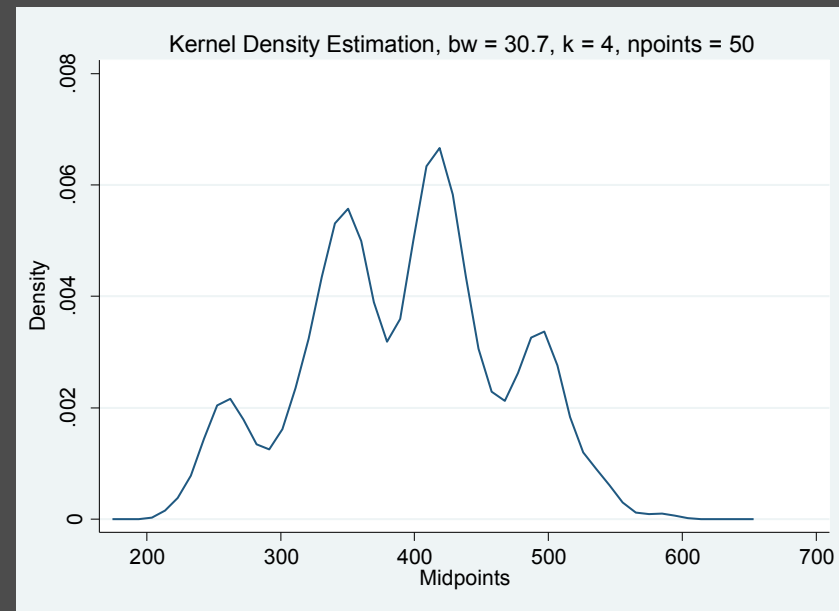
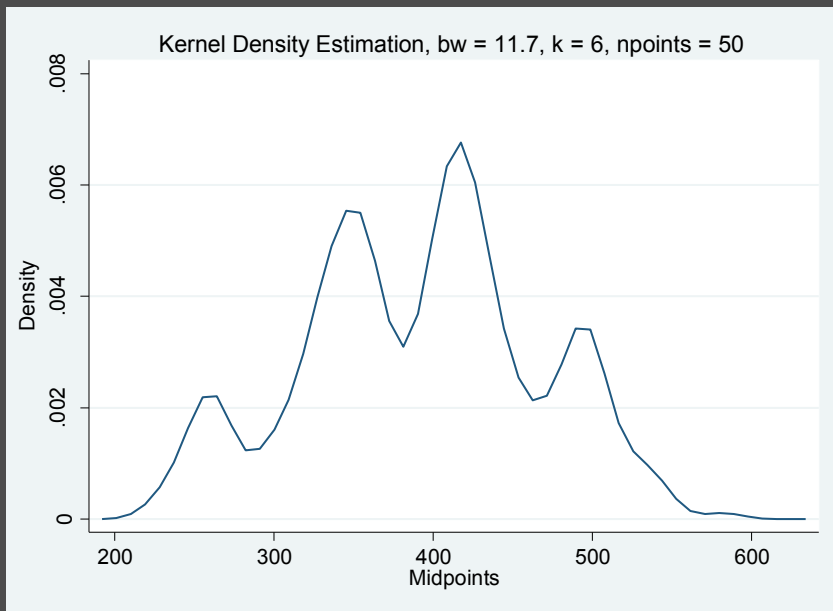
- 5.- local options "Bwidth(real 0) Kercode(integer 0) NPoint(integer 0) Gen(string) noGraph T1title(string) **M**Symbol(string) Connect(string) *"
- 6.- di **as error** "syntax: boxdetra varname iwidth detravar"
- 7.- exit
- 10.- gen `nuobs'=**r(N)**
- 19.- noi di **as result** "WORKING WITH EACH VALUE. PLEASE BE PATIENT"
- 43.- noi di **as result** "CALCULATING ADAPTIVE VALUES. DON'T DESPAIR"
- 55.- noi di **as result** "DONE. THANK FOR YOUR PATIENCE"

kerneld1.ado versión 11.0

La elección del kernel es siguiendo a esta tabla

- 1 = Uniforme
- 2 = Triangular
- 3 = Epanechnikov
- 4 = Cuártico (biponderado)
- 5 = Triponderado
- 6 = Gaussiano
- 7 = Coseno

kernel1.ado



**kernel1 length, bw(11.7)
k(6) np(50)**

**kernel1 length, bw(30.7)
k(4) np(50)**

Amplitud crítica de banda

- En la evaluación no paramétrica de la multimodalidad por el método de bootstrap suavizado propuesto por Silverman (1981) es la determinación precisa del último valor de amplitud de banda compatible con la hipótesis para un cierto número de modas (la amplitud crítica de banda)
- Si la banda crítica no se estima con precisión, el resultado de la prueba será cuestionable

Amplitud crítica de banda

- ⦿ Por lo general un procedimiento simple de búsqueda binario puede ser usado en la práctica para encontrar la amplitud crítica de banda (Silverman, 1986).
- ⦿ Pero en nuestra experiencia (con nuestros algoritmos) se ha mostrado que en ocasiones es necesario probar el número de modas de una gran colección de EDK's con una variación gradual de la amplitud de banda.
- ⦿ Esta tarea tal vez sea monótona y consume tiempo incluso con las teclas de edición de ayuda de Stata (Regreso de Página) que permite repetir el comando y cambiar solo las partes requeridas.

Amplitud crítica de banda

- ⦿ Esta fue la principal motivación para escribir el programa critiband.ado y su actualización a critiband1.ado. Este programa repite el cálculo del EDK con una serie específica de valores de ancho de banda, contando el número de modas y reportando el resultado.
- ⦿ Como critiband1.ado tiene una relación esencial con el programa warpdenm1.ado, casi todas las acciones para las opciones del EDK (warpdenm1.ado) requieren la misma entrada.
- ⦿ Esta nota es importante por que en la búsqueda del ancho de banda crítico hemos encontrado que un número de 30 o 40 histogramas desplazados es necesario para tener resultados confiables.

Las actualizaciones para el programa critiband se presentan a continuación

- 5.- local options "BWHHigh(real 0) BWLow(real 0) STsize(real 0) Mval(integer 0) noGraph T1title(string) Symbol(string) Connect(string) *"
 - ◉ 14.- exit}
 - ◉ 55.- scalar nuobs= _result(1)
 - ◉ 56.- scalar maxval= _result(6)
 - ◉ 57. scalar minval= _result(5)
 - ◉ 100.- scalar binum= _result(1)
 - ◉ 124.- if "`graph'" ~= "nograph" {
 - ◉ 128.- if "`symbol'"==" " { local symbol "." }
 - ◉ 134.- graph `lfh' `midval', `options' t1("`t1title'") s(`symbol') c(`connect')
- ◉ 5.- local options "BWHHigh(real 0) BWLow(real 0) STsize(real 0) Mval(integer 0) noGraph T1title(string) **M**Symbol(string) Connect(string) *"
 - ◉ 14.- exit
 - ◉ }
 - ◉ 55.- scalar nuobs= **r(N)**
 - ◉ 56.- scalar maxval= **r(max)**
 - ◉ 57. scalar minval= **r(min)**
 - ◉ 100.- scalar binum= **r(N)**
 - ◉ 124.- if "`graph'" != "nograph" {
 - ◉ 128.- if "`**msymbol**'"==" " {
 - ◉ local **msymbol** " . "
 - ◉ }
 - ◉ 134.- **scatter** `lfh' `midval', `options' t1("`t1title'") **ms**(`**msymbol**') c(`connect')

critiband.ado versión 4.0

critiband1.ado versión 11.0

Amplitud crítica de banda

```
. critiband1 bodlen, bwh(23.5) bwl(23.1) st(.01) m(40)
Estimation number = 1           Bandwidth = 23.5           Number of modes = 1
Estimation number = 2           Bandwidth = 23.49         Number of modes = 1
Estimation number = 3           Bandwidth = 23.48         Number of modes = 1
Estimation number = 4           Bandwidth = 23.47         Number of modes = 1
Estimation number = 5           Bandwidth = 23.46         Number of modes = 2
Estimation number = 6           Bandwidth = 23.45         Number of modes = 1
Estimation number = 7           Bandwidth = 23.44         Number of modes = 1
Estimation number = 8           Bandwidth = 23.43         Number of modes = 1
Estimation number = 9           Bandwidth = 23.42         Number of modes = 2
Estimation number = 10          Bandwidth = 23.41         Number of modes = 1
Estimation number = 11          Bandwidth = 23.4           Number of modes = 1
Estimation number = 12          Bandwidth = 23.39         Number of modes = 1
Estimation number = 13          Bandwidth = 23.38         Number of modes = 1
Estimation number = 14          Bandwidth = 23.37         Number of modes = 1
Estimation number = 15          Bandwidth = 23.36         Number of modes = 1
Estimation number = 16          Bandwidth = 23.35         Number of modes = 2
Estimation number = 17          Bandwidth = 23.34         Number of modes = 2
Estimation number = 18          Bandwidth = 23.33         Number of modes = 2
```

Amplitud crítica de banda

```
. critiband1 bodlen, bwh(4) bwl(3.7) st(.01) m(40)
```

| | | |
|------------------------|------------------|---------------------|
| Estimation number = 1 | Bandwidth = 4 | Number of modes = 4 |
| Estimation number = 2 | Bandwidth = 3.99 | Number of modes = 4 |
| Estimation number = 3 | Bandwidth = 3.98 | Number of modes = 4 |
| Estimation number = 4 | Bandwidth = 3.97 | Number of modes = 4 |
| Estimation number = 5 | Bandwidth = 3.96 | Number of modes = 5 |
| Estimation number = 6 | Bandwidth = 3.95 | Number of modes = 4 |
| Estimation number = 7 | Bandwidth = 3.94 | Number of modes = 4 |
| Estimation number = 8 | Bandwidth = 3.93 | Number of modes = 4 |
| Estimation number = 9 | Bandwidth = 3.92 | Number of modes = 5 |
| Estimation number = 10 | Bandwidth = 3.91 | Number of modes = 4 |
| Estimation number = 11 | Bandwidth = 3.9 | Number of modes = 4 |
| Estimation number = 12 | Bandwidth = 3.89 | Number of modes = 5 |
| Estimation number = 13 | Bandwidth = 3.88 | Number of modes = 4 |
| Estimation number = 14 | Bandwidth = 3.87 | Number of modes = 5 |
| Estimation number = 15 | Bandwidth = 3.86 | Number of modes = 5 |
| Estimation number = 16 | Bandwidth = 3.85 | Number of modes = 5 |
| Estimation number = 17 | Bandwidth = 3.84 | Number of modes = 5 |
| Estimation number = 18 | Bandwidth = 3.83 | Number of modes = 5 |
| Estimation number = 19 | Bandwidth = 3.82 | Number of modes = 5 |
| Estimation number = 20 | Bandwidth = 3.81 | Number of modes = 5 |
| Estimation number = 21 | Bandwidth = 3.8 | Number of modes = 5 |
| Estimation number = 22 | Bandwidth = 3.79 | Number of modes = 4 |
| Estimation number = 23 | Bandwidth = 3.78 | Number of modes = 4 |
| Estimation number = 24 | Bandwidth = 3.77 | Number of modes = 5 |
| Estimation number = 25 | Bandwidth = 3.76 | Number of modes = 5 |
| Estimation number = 26 | Bandwidth = 3.75 | Number of modes = 5 |

Amplitud crítica de banda

```
. critiband1 bodlen, bwh(3.1) bwl(2.9) st(.01) m(40)
Estimation number = 1           Bandwidth = 3.1           Number of modes = 6
Estimation number = 2           Bandwidth = 3.09          Number of modes = 6
Estimation number = 3           Bandwidth = 3.08          Number of modes = 7
Estimation number = 4           Bandwidth = 3.07          Number of modes = 6
Estimation number = 5           Bandwidth = 3.06          Number of modes = 6
Estimation number = 6           Bandwidth = 3.05          Number of modes = 7
Estimation number = 7           Bandwidth = 3.04          Number of modes = 6
Estimation number = 8           Bandwidth = 3.03          Number of modes = 6
Estimation number = 9           Bandwidth = 3.02          Number of modes = 6
Estimation number = 10          Bandwidth = 3.01          Number of modes = 7
Estimation number = 11          Bandwidth = 3              Number of modes = 7
Estimation number = 12          Bandwidth = 2.99          Number of modes = 7
Estimation number = 13          Bandwidth = 2.98          Number of modes = 7
Estimation number = 14          Bandwidth = 2.97          Number of modes = 7
Estimation number = 15          Bandwidth = 2.96          Number of modes = 7
Estimation number = 16          Bandwidth = 2.95          Number of modes = 7
Estimation number = 17          Bandwidth = 2.94          Number of modes = 7
Estimation number = 18          Bandwidth = 2.93          Number of modes = 7
Estimation number = 19          Bandwidth = 2.92          Number of modes = 7
Estimation number = 20          Bandwidth = 2.91          Number of modes = 7
Estimation number = 21          Bandwidth = 2.9           Number of modes = 7
```

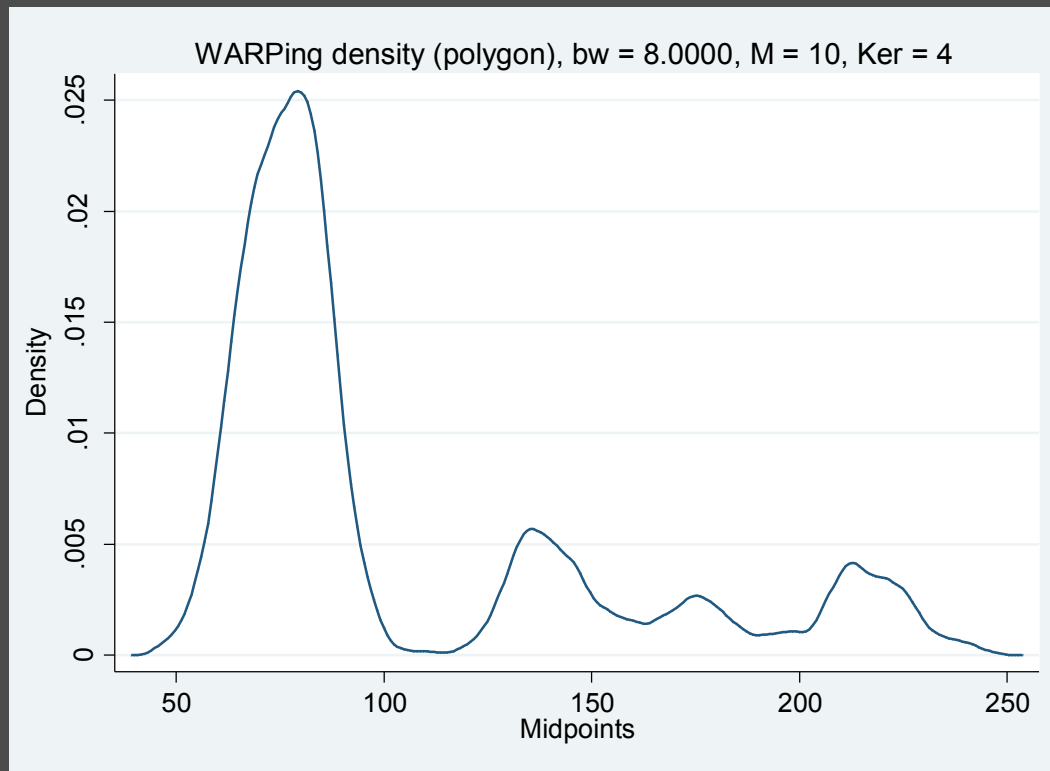
Las actualizaciones para el programa warpdenm son las siguientes

- 5.- local options "Bwidth(real 0) Mval(integer 0) Kercode(integer 0) SStep NUMOdes MOdes NPoints Gen(string) noGraph T1title(string) Symbol(string) Connect(string) *"
 - 10.- if "`gen"~="" {
 - 36.- if "`modes"~="" & "`numodes"=="" {
 - 41.- scalar nuobs=_result(1)
 - 42.- scalar maxval=_result(6)
 - 43.- scalar minval=_result(5)
 - 116.- scalar binum=_result(1)
 - 141.- if "`graph" ~= "nograph" {
 - 144.- if "`step"~="" {
 - 153.- if "`symbol"=="" { local symbol "." }
 - 163.- graph `lfh' `lowcut', `options' t1("`t1title") s(`symbol') c(`connect')
- 5.- local options "Bwidth(real 0) Mval(integer 0) Kercode(integer 0) SStep NUMOdes MOdes NPoints Gen(string) noGraph T1title(string) **M**Symbol(string) Connect(string) *"
 - 10.- if "`gen"**!**="" {
 - 36.- if "`modes"**!**="" & "`numodes"=="" {
 - 41.- scalar nuobs= **r(N)**
 - 42.- scalar maxval= **r(max)**
 - 43.- scalar minval= **r(min)**
 - 116.- scalar binum= **r(N)**
 - 141.- if "`graph" **!**= "nograph" {
 - 144.- if "`step"**!**="" {
 - 153. if "`**msymbol**"=="" {
 - local **msymbol** "."
 - }
 - 163.- **scatter** `lfh' `lowcut', `options' t1("`t1title") **ms**(`**msymbol**') c(`connect')

Warpdenm.ado versión 5.0

Warpdenm1.ado versión 11.0

Estimación de densidad por HDP-PPPR warpdenm1 bodlen, b(8) m(10) k(4)



Actualizaciones para el programa silvtest.ado

- 5.- local options "CRitbw(real 0) Mval(integer 0) NURIni(integer 1) NURFin(integer 0) CNModes(integer 0) noGraph T1title(string) Symbol(string) Connect(string) *"
- 57.-scalar nuobs= _result(1)
- 58.-scalar maxval= _result(6)
- 59.-scalar minval= _result(5)
- 103.-scalar binum= _result(1)
- 132.-if "`symbol'"==" " { local symbol " " }
- 138.-graph `lfh' `midval', `options' t1("`t1title'") s(`symbol') c(`connect')

Silvtest.ado version 4.0

- 5.-local options "CRitbw(real 0) Mval(integer 0) NURIni(integer 1) NURFin(integer 0) CNModes(integer 0) noGraph T1title(string) MSymbol(string) Connect(string) *"
- 57.-scalar nuobs= r(N)
- 58.-scalar maxval= r(max)
- 59.-scalar minval= r(min)
- 103.-scalar binum= r(N)
- 132.-if "`msymbol'"==" " {
local symbol " "
}
138.-sactter `lfh' `midval', `options' t1("`t1title'") ms(`msymbol') c(`connect')

Silvtest1.ado versión 11.0

Prueba de multimodalidad de Silverman (con bootsam)

```
. use catfilen, clear
. set mem 32m
. keep bodlen
. sum catfilen
. set seed 220409
. boot bootsamb, ar(bodlen 23.36 49.5904) i(500)
```

warning: data in memory will be lost.

Press enter to continue, Ctrl-Break to abort.

(output omitted)

Contains data

```
obs:          320,500          bootsamb bootstrap
vars:          4
size:         6,410,000 (80.9% of memory free)
```

| variable name | storage type | display format | value label | variable label |
|---------------|--------------|----------------|-------------|----------------|
| _rep | long | %12.0g | | replication |
| bodlen | float | %9.0g | | |
| ysm | float | %9.0g | | |
| _obs | long | %12.0g | | observations |

Sorted by:

Note: dataset has changed since last saved

Prueba de multimodalidad de Silverman

```
⊙ . silvtest ysm _rep, cr(23.36) m(40) nurf(500) cnm(1) nog
⊙ bs sample      1          Number of modes = 1
⊙ bs sample      2          Number of modes = 1
⊙ bs sample      3          Number of modes = 1
⊙ bs sample      4          Number of modes = 1
⊙ bs sample      5          Number of modes = 1
⊙ .
⊙ .
⊙ .
⊙ bs sample      497        Number of modes = 1
⊙ bs sample      498        Number of modes = 1
⊙ bs sample      499        Number of modes = 1
⊙ bs sample      500        Number of modes = 1

⊙ Critical number of modes =      1

⊙ P value =          0 / 500 = 0.0000
```

Cuadro de la prueba de Silverman

Amplitudes críticas de banda y niveles estimados de significancia para datos de longitud estándar de *Cathorops melanopus* ($n = 641$)

| Número de modas | Amplitudes críticas de banda | valor de P |
|-----------------|------------------------------|--------------|
| 1 | 23.36 | 0.0000 |
| 2 | 19.43 | 0.0000 |
| 3 | 9.64 | 0.1560 |
| 4 | 3.78 | 0.7660 |
| 5 | 3.23 | 0.8140 |
| 6 | 3.02 | 0.6780 |

Nota: los valores de P se obtiene a partir de $B = 500$ repeticiones bootstrap de tamaño 641. en ancho de banda sugerido por la prueba es $(9.63+3.78)/2= 6.705$

El número de modas que mejor representa a nuestros datos

- . use catfile, clear
- . di (9.63+3.78)/2
- 6.705
- . warpdenm bodlen, b(6.7) m(10) k(6) numo mo

○ Number of modes = 4

○

○ Modes in WARPing density estimation, bw = 6.7, M = 10, Ker = 6

○

○ Mode (1) = 77.7200

○ Mode (2) = 136.6800

○ Mode (3) = 174.2000

○ Mode (4) = 214.4000

○

warpdenm1 bodlen, b(6.7) m(10)
k(6) numo mo

