intRegGOF un paquete para Bondad de Ajuste mediante Regresión integrada

Jorge Luis Ojeda Cabrera.

Dept. de Métodos Estadísticos,
U. de Zaragoza.

III Jornadas de UsuaRios de R

Nov. 2011

intRegGOF un paquete para Bondad de Ajuste mediante Regresión integrada

Jorge Luis Ojeda Cabrera.

Dept. de Métodos Estadísticos,
U. de Zaragoza.

III Jornadas de UsuaRios de R

Nov. 2011

Objetivo

Presentar el paquete `intRegGOF` para desarrollar test de *Bondad de Ajuste* para modelos de *Regresión*.

Objetivo

Presentar el paquete `intRegGOF` para desarrollar test de *Bondad de Ajuste* para modelos de *Regresión*.

- Test de Bondad de Ajuste y Sesgo por selección.

Objetivo

Presentar el paquete `intRegGOF` para desarrollar test de *Bondad de Ajuste* para modelos de *Regresión*.

- Test de Bondad de Ajuste y Sesgo por selección.
- Uso e implementación del paquete `intRegGOF`:

Objetivo

Presentar el paquete `intRegGOF` para desarrollar test de *Bondad de Ajuste* para modelos de *Regresión*.

- Test de Bondad de Ajuste y Sesgo por selección.
- Uso e implementación del paquete `intRegGOF`:
- Ejemplos de su uso:

Objetivo

Presentar el paquete `intRegGOF` para desarrollar test de *Bondad de Ajuste* para modelos de *Regresión*.

- Test de Bondad de Ajuste y Sesgo por selección.
- Uso e implementación del paquete `intRegGOF`:
- Ejemplos de su uso:
  - Un modelo sencillo.

Objetivo

Presentar el paquete `intRegGOF` para desarrollar test de *Bondad de Ajuste* para modelos de *Regresión*.

- Test de Bondad de Ajuste y Sesgo por selección.
- Uso e implementación del paquete `intRegGOF`:
- Ejemplos de su uso:
  - Un modelo sencillo.
  - Datos Sesgados por longitud.

Objetivo

Presentar el paquete `intRegGOF` para desarrollar test de *Bondad de Ajuste* para modelos de *Regresión*.

- Test de Bondad de Ajuste y Sesgo por selección.
- Uso e implementación del paquete `intRegGOF`:
- Ejemplos de su uso:
  - Un modelo sencillo.
  - Datos Sesgados por longitud.
  - Censura.

Objetivo

Presentar el paquete `intRegGOF` para desarrollar test de *Bondad de Ajuste* para modelos de *Regresión*.

- Test de Bondad de Ajuste y Sesgo por selección.
- Uso e implementación del paquete `intRegGOF`:
- Ejemplos de su uso:
  - Un modelo sencillo.
  - Datos Sesgados por longitud.
  - Censura.
- ¿ ...Mejoras ?

Bondad de Ajuste

Bondad de Ajuste:

Bondad de Ajuste

Bondad de Ajuste: Dada una pobl. (X, Y), y una *familia paramétrica de funciones*

$$\mathcal{M}_0 = \{m(x;\boldsymbol{\beta}) \ : \ \boldsymbol{\beta} = (\beta_1, \ldots, \beta_k) \in \Omega \subset \mathbf{R}^k\}$$

Bondad de Ajuste

Bondad de Ajuste: Dada una pobl. (X, Y), y una *familia paramétrica de funciones*

$$\mathcal{M}_0 = \{m(x;\boldsymbol{\beta}) \ : \ \boldsymbol{\beta} = (\beta_1, \ldots, \beta_k) \in \Omega \subset \mathbf{R}^k\}$$

¿ Podemos asumir $m(x) = \mathbf{E}\,[Y|X = x]$ está en $\mathcal{M}_0$?.

$$H_0 : m \in \mathcal{M}_0 \quad \text{vs.} \quad H_1 : m \notin \mathcal{M}_0.$$

Bondad de Ajuste

Bondad de Ajuste: Dada una pobl. (X, Y), y una *familia paramétrica de funciones*

$$\mathcal{M}_0 = \{m(x;\boldsymbol{\beta}) \ : \ \boldsymbol{\beta} = (\beta_1, \ldots, \beta_k) \in \Omega \subset \mathbf{R}^k\}$$

¿ Podemos asumir $m(x) = \mathbf{E}\,[Y|X = x]$ está en $\mathcal{M}_0$?.

$$H_0 : m \in \mathcal{M}_0 \quad \text{vs.} \quad H_1 : m \notin \mathcal{M}_0.$$

¿ Qué clase de modelos es mejor para m, $\mathcal{M}_0$ o $\mathcal{M}_1$?.

$$H_0 : m \in \mathcal{M}_0 \quad \text{vs.} \quad H_1 : m \in \mathcal{M}_1.$$

[Kozek(1990)], [Härdle and Mammen(1993)], [Hart(1997)], [Stute(1997)],[Delgado and González Manteiga(2001)], [van Keilegom et al.(2007)van Keilegom, Sánchez Sellero, and González-Manteiga], ...

paquete R:gof, ver `http://cran.r-project.org/web/packages/gof/gof.pdf`

Sesgo por Selección

...Pero los datos están sesgados:

Sesgo por Selección

...Pero los datos están sesgados:

Sesgo por Selección: *no disponemos de observaciones* de la pobl. de interés (X, Y) con distrib. F, sino de (X^w, Y^w) con distrib. F^w... ambas relacionadas

Sesgo por Selección

...Pero los datos están sesgados:

Sesgo por Selección: *no disponemos de observaciones* de la pobl. de interés (X, Y) con distrib. F, sino de (X^w, Y^w) con distrib. F^w... ambas relacionadas

$$d\,F^w(x,y) = \frac{w(x,y)}{\mu_w} d\,F(x,y),$$

Sesgo por Selección

...Pero los datos están sesgados:

Sesgo por Selección: *no disponemos de observaciones* de la pobl. de interés (X, Y) con distrib. F, sino de (X^w, Y^w) con distrib. F^w... ambas relacionadas

$$d\,F^w(x,y) = \frac{w(x,y)}{\mu_w} d\,F(x,y),$$

Estimación: *compensación* del sesgo

Sesgo por Selección

...Pero los datos están sesgados:

Sesgo por Selección: *no disponemos de observaciones* de la pobl. de interés (X, Y) con distrib. F, sino de (X^w, Y^w) con distrib. F^w... ambas relacionadas

$$d\,F^w(x,y) = \frac{w(x,y)}{\mu_w} d\,F(x,y),$$

Estimación: *compensación* del sesgo, si $(x_1, y_1), \ldots, (x_n, y_n)$ es m.a.s. de (X^w, Y^w) y $w_i = w(x_i, y_i)$

$$\hat{\beta}_n = \arg\min_{\beta} \sum_{i=1}^{n} \frac{1}{w_i} \left(y_i - m(x_i;\beta)\right)^2$$

[Patil and Rao(1978)],[Quesenberry and Jewell(1986)],[Patil(2002)],

...,[Cohen et al.(2002)Cohen, Kemperman, and Sackrowitz], [Oshlack and Wakefield(2009)],...

Sesgo por Selección

...Pero los datos están sesgados:

Sesgo por Selección: *no disponemos de observaciones* de la pobl. de interés (X, Y) con distrib. F, sino de (X^w, Y^w) con distrib. F^w... ambas relacionadas

$$d\,F^w(x,y) = \frac{w(x,y)}{\mu_w} d\,F(x,y),$$

Estimación: *compensación* del sesgo, si $(x_1, y_1), \ldots, (x_n, y_n)$ es m.a.s. de (X^w, Y^w) y $w_i = w(x_i, y_i)$

$$\hat{\beta}_n = \arg\min_{\beta} \sum_{i=1}^{n} \frac{1}{w_i} \left(y_i - m(x_i;\beta)\right)^2$$

[Patil and Rao(1978)],[Quesenberry and Jewell(1986)],[Patil(2002)],

...,[Cohen et al.(2002)Cohen, Kemperman, and Sackrowitz], [Oshlack and Wakefield(2009)],...

Acumular Residuos

$$\hat{\epsilon}_i = y_i - \hat{y}_i = y_i - m\left(x_i; \hat{\boldsymbol{\beta}}_n\right)$$

Marked Empirical Process: [Stute(1997)], [Delgado and González Manteiga(2001)]

$$R_n(x) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \hat{\epsilon}_i \, \mathbf{1}(x_i \leq x)$$

Acumular Residuos

$$\hat{\epsilon}_i = y_i - \hat{y}_i = y_i - m\left(x_i; \hat{\boldsymbol{\beta}}_n\right)$$

Compensated Marked Empirical Process: [Ojeda et al.(2007)Ojeda, W., and Cristóbal]

$$R_n^w(x) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \frac{1}{w_i} \hat{\epsilon}_i \, \mathbf{1}(x_i \leq x)$$

Acumular Residuos

$$\hat{\epsilon}_i = y_i - \hat{y}_i = y_i - m\left(x_i; \hat{\boldsymbol{\beta}}_n\right)$$

Compensated Marked Empirical Process: [Ojeda et al.(2007)Ojeda, W., and Cristóbal]

$$R_n^w(x) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \frac{1}{w_i} \epsilon_i \, \mathbf{1}(x_i \leq x)$$

$$+ \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \frac{1}{w_i} \left(m(x_i) - m\left(x_i; \hat{\boldsymbol{\beta}}_n\right) \right) \mathbf{1}(x_i \leq x)$$

Acumular Residuos

$$\hat{\epsilon}_i = y_i - \hat{y}_i = y_i - m\left(x_i; \hat{\boldsymbol{\beta}}_n\right)$$

Compensated Marked Empirical Process: [Ojeda et al.(2007)Ojeda, W., and Cristóbal]

$$R_n^w(x) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \frac{1}{w_i} \epsilon_i \, \mathbf{1}(x_i \leq x)$$

$$+ \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \frac{1}{w_i} \left(m(x_i) - m\left(x_i; \hat{\boldsymbol{\beta}}_n\right) \right) \mathbf{1}(x_i \leq x)$$

Si H_0 es cierta: $R_n^w(x) \longrightarrow R_\infty^w(x)$ en $D(\mathbf{R}^d)$

Acumular Residuos

$$\hat{\epsilon}_i = y_i - \hat{y}_i = y_i - m\left(x_i; \hat{\boldsymbol{\beta}}_n\right)$$

Compensated Marked Empirical Process: [Ojeda et al.(2007)Ojeda, W., and Cristóbal]

$$R_n^w(x) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \frac{1}{w_i} \epsilon_i \, \mathbf{1}(x_i \leq x)$$

$$+ \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \frac{1}{w_i} \left(m(x_i) - m\left(x_i; \hat{\boldsymbol{\beta}}_n\right) \right) \mathbf{1}(x_i \leq x)$$

Si H_0 es cierta: $R_n^w(x) \longrightarrow R_\infty^w(x)$ en $D(\mathbf{R}^d)$, $R_\infty^w(x)$ es un proceso gausiano centrado con una func. covar. complicada (se usa Bootstrap).

Acumular Residuos

$$\hat{\epsilon}_i = y_i - \hat{y}_i = y_i - m\left(x_i; \hat{\boldsymbol{\beta}}_n\right)$$

Compensated Marked Empirical Process: [Ojeda et al.(2007)Ojeda, W., and Cristóbal]

$$R_n^w(x) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \frac{1}{w_i} \epsilon_i \, \mathbf{1}(x_i \leq x)$$

$$+ \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \frac{1}{w_i} \left(m(x_i) - m\left(x_i; \hat{\boldsymbol{\beta}}_n\right) \right) \mathbf{1}(x_i \leq x)$$

Si H_0 es cierta: $R_n^w(x) \to R_\infty^w(x)$ en $D(\mathbf{R}^d)$, $R_\infty^w(x)$ es un proceso gausiano centrado con una func. covar. complicada (se usa Bootstrap).
Estadísticos para el Test:

$$K_n^\infty = \sup_{x \in \mathrm{supp}(X)} |R_n^w(x)| \, ; \qquad W_n^2 = \int_{\mathrm{supp}(X)} R_n^w(z)^2 \, dF(z).$$

Bootstrap

$$\begin{aligned} \hat{\epsilon}_i &= \left(y_i - m\left(x; \hat{\boldsymbol{\beta}}_n\right)\right) \\ y_i^* &= m\left(x_i; \hat{\boldsymbol{\beta}}_n\right) + \hat{\epsilon}_i \, \gamma_i; \qquad x_i^* = x_i \end{aligned}$$

Bootstrap

$$\begin{aligned} \hat{\epsilon}_i &= \left(y_i - m\left(x; \hat{\boldsymbol{\beta}}_n\right)\right) \\ y_i^* &= m\left(x_i; \hat{\boldsymbol{\beta}}_n\right) + \hat{\epsilon}_i \, \gamma_i; \qquad x_i^* = x_i \end{aligned}$$

γ_i es la v.a. Wild Bootstrap ($\mathbf{E}[\gamma_i] = 0, \mathbf{Var}[\gamma_i] = 1$).

Bootstrap

$$\begin{aligned} \hat{\epsilon}_i &= \left(y_i - m\left(x; \hat{\boldsymbol{\beta}}_n\right)\right) \\ y_i^* &= m\left(x_i; \hat{\boldsymbol{\beta}}_n\right) + \hat{\epsilon}_i \, \gamma_i; \qquad x_i^* = x_i \end{aligned}$$

γ_i es la v.a. Wild Bootstrap ($\mathbf{E}[\gamma_i] = 0, \mathbf{Var}[\gamma_i] = 1$).
Bootstrap Compensated Marked Empirical Process:

$$R_n^{w^1 *}(x) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \frac{1}{w_i} \left(y_i^* - m_{\tilde{\boldsymbol{\beta}}_n}(x)\right) \mathbf{1}\{x_i^* \leq x\}.$$

siendo $\tilde{\boldsymbol{\beta}}_n^*$ estimado con (x_i^*, y_i^*) $i = 1, \ldots, n$.

Bootstrap

$$\begin{aligned} \hat{\epsilon}_i &= \left(y_i - m\left(x; \hat{\boldsymbol{\beta}}_n\right)\right) \\ y_i^* &= m\left(x_i; \hat{\boldsymbol{\beta}}_n\right) + \hat{\epsilon}_i \, \gamma_i; \qquad x_i^* = x_i \end{aligned}$$

γ_i es la v.a. Wild Bootstrap ($\mathbf{E}[\gamma_i] = 0, \mathbf{Var}[\gamma_i] = 1$).
Bootstrap Compensated Marked Empirical Process:

$$R_n^{w^1 *}(x) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \frac{1}{w_i} \left(y_i^* - m_{\tilde{\boldsymbol{\beta}}_n}(x)\right) \mathbf{1}\{x_i^* \leq x\}.$$

siendo $\tilde{\boldsymbol{\beta}}_n^*$ estimado con (x_i^*, y_i^*) $i = 1, \ldots, n$.
$R_n^{w^1 *}(x) \to R_\infty^w(x)$.

Bootstrap

$$\begin{aligned} \hat{\epsilon}_i &= \left(y_i - m\left(x; \hat{\boldsymbol{\beta}}_n\right)\right) \\ y_i^* &= m\left(x_i; \hat{\boldsymbol{\beta}}_n\right) + \hat{\epsilon}_i \, \gamma_i; \qquad x_i^* = x_i \end{aligned}$$

γ_i es la v.a. Wild Bootstrap ($\mathbf{E}[\gamma_i] = 0, \mathbf{Var}[\gamma_i] = 1$).
Bootstrap Compensated Marked Empirical Process:

$$R_n^{w^1{}^*}(x) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} \frac{1}{w_i} \left(y_i^* - m_{\tilde{\boldsymbol{\beta}}_n}(x)\right) \mathbf{1}\{x_i^* \leq x\}.$$

siendo $\tilde{\boldsymbol{\beta}}_n^*$ estimado con $\left(x_i^*, y_i^*\right)$ $i = 1, \ldots, n$.
$R_n^{w^1{}^*}(x) \to R_\infty^w(x)$.
Se generan $b = 1, \ldots, B$ muestras bootstrap de los estadísticos

$$K_{nb}^* = \sup_{x \in \text{supp}(X)} \left| R_n^{w^1{}^*}(x) \right|, \quad W_{nb}^{2^*} = \int_{\text{supp}(X)} R_n^{w^1{}^*}(z)^2 \, dF(z).$$

y se usan para calcular los p–valores.

Principales Características

- Una clases $\mathcal{M}$ de modelos se representa en `R` mediante su ajuste: un objeto de la clase `lm`, `glm`, `nls`.

Principales Características

- Una clases $\mathcal{M}$ de modelos se representa en `R` mediante su ajuste: un objeto de la clase `lm`, `glm`, `nls`.
- Basta que dichas clases dispongan de métodos `fitted`, `residuals`, además de acceso a los datos y a los pesos (param. `weight`) si es usado.

Principales Características

- Una clases $\mathcal{M}$ de modelos se representa en `R` mediante su ajuste: un objeto de la clase `lm`, `glm`, `nls`.
- Basta que dichas clases dispongan de métodos `fitted`, `residuals`, además de acceso a los datos y a los pesos (param. `weight`) si es usado.
- La salida comprende el valor de los estadísticos K_n y W_n^2, así como sus respectivos p–valores.

Principales Características

- Una clases $\mathcal{M}$ de modelos se representa en `R` mediante su ajuste: un objeto de la clase `lm`, `glm`, `nls`.
- Basta que dichas clases dispongan de métodos `fitted`, `residuals`, además de acceso a los datos y a los pesos (param. `weight`) si es usado.
- La salida comprende el valor de los estadísticos K_n y W_n^2, así como sus respectivos p–valores.
- Dispone de métodos para desarrollar test para un modelo ($H_0 : m \in \mathcal{M}_0$ vs. $H_1 : m \notin \mathcal{M}_0$) o para comparar modelos ($H_0 : m \in \mathcal{M}_0$ vs. $H_1 : m \in \mathcal{M}_1$).

Principales Características

- Una clases $\mathcal{M}$ de modelos se representa en `R` mediante su ajuste: un objeto de la clase `lm`, `glm`, `nls`.
- Basta que dichas clases dispongan de métodos `fitted`, `residuals`, además de acceso a los datos y a los pesos (param. `weight`) si es usado.
- La salida comprende el valor de los estadísticos K_n y W_n^2, así como sus respectivos p–valores.
- Dispone de métodos para desarrollar test para un modelo ($H_0 : m \in \mathcal{M}_0$ vs. $H_1 : m \notin \mathcal{M}_0$) o para comparar modelos ($H_0 : m \in \mathcal{M}_0$ vs. $H_1 : m \in \mathcal{M}_1$).
- Dispone de métodos para desarrollar gráficos.

Interfaz

- Las funciones que desarrollan el interfaz son.

Interfaz

- Las funciones que desarrollan el interfaz son.
  - `intRegGOF(obj, covars = NULL, B = 499, LINMOD = F)`

$$H_0 : m \in \mathcal{M}_0 \quad \text{vs.} \quad H_1 : m \notin \mathcal{M}_0.$$

`obj` es el *mejor ajuste* en $\mathcal{M}_0$.

Interfaz

- Las funciones que desarrollan el interfaz son.
  - `intRegGOF(obj, covars = NULL, B = 499, LINMOD = F)`

    $$H_0 : m \in \mathcal{M}_0 \quad \text{vs.} \quad H_1 : m \notin \mathcal{M}_0.$$

    `obj` es el *mejor ajuste* en $\mathcal{M}_0$.
  - `anovarIntReg(objH0, ..., covars = NULL, B = 499, LINMOD = F, INCREMENTAL = F)`

    $$H_0 : m \in \mathcal{M}_0 \quad \text{vs.} \quad H_1 : m \in \mathcal{M}_k.$$

    `objH0` y `...` son los *mejores ajustes* en las clases $\mathcal{M}_0$, ...,$\mathcal{M}_k$. `INCREMENTAL` determina si las comparaciones de los test son con $\mathcal{M}_0$ o se comparan entre sí.

Interfaz

- Las funciones que desarrollan el interfaz son.
  - `intRegGOF(obj, covars = NULL, B = 499, LINMOD = F)`

    $$H_0 : m \in \mathcal{M}_0 \quad \text{vs.} \quad H_1 : m \notin \mathcal{M}_0.$$

    `obj` es el *mejor ajuste* en $\mathcal{M}_0$.
  - `anovarIntReg(objH0, ..., covars = NULL, B = 499, LINMOD = F, INCREMENTAL = F)`

    $$H_0 : m \in \mathcal{M}_0 \quad \text{vs.} \quad H_1 : m \in \mathcal{M}_k.$$

    `objH0` y `...` son los *mejores ajustes* en las clases $\mathcal{M}_0$, ...,$\mathcal{M}_k$. `INCREMENTAL` determina si las comparaciones de los test son con $\mathcal{M}_0$ o se comparan entre sí.
  - `plot.intRegGOF(obj, covar = 1, ADD = F, ...)` gráfico de R_n^w.

Interfaz

- Las funciones que desarrollan el interfaz son.
  - `intRegGOF(obj, covars = NULL, B = 499, LINMOD = F)`

    $$H_0 : m \in \mathcal{M}_0 \quad \text{vs.} \quad H_1 : m \notin \mathcal{M}_0.$$

    `obj` es el *mejor ajuste* en $\mathcal{M}_0$.
  - `anovarIntReg(objH0, ..., covars = NULL, B = 499, LINMOD = F, INCREMENTAL = F)`

    $$H_0 : m \in \mathcal{M}_0 \quad \text{vs.} \quad H_1 : m \in \mathcal{M}_k.$$

    `objH0` y `...` son los *mejores ajustes* en las clases $\mathcal{M}_0$, ...,$\mathcal{M}_k$. `INCREMENTAL` determina si las comparaciones de los test son con $\mathcal{M}_0$ o se comparan entre sí.
  - `plot.intRegGOF(obj, covar = 1, ADD = F, ...)` gráfico de R_n^w.
- Método `print` y utilidades diversas.

Implementación

- Covariables:

Implementación

- Covariables:
  - Orden lexicográfico para la comparación.

Implementación

- Covariables:
  - Orden lexicográfico para la comparación.
  - Generar para cada x_i la lista de los x_j que son menores que él (`getLessThan()`) y acumularlos (`mvCumSum()`).

Implementación

- Covariables:
  - Orden lexicográfico para la comparación.
  - Generar para cada x_i la lista de los x_j que son menores que él (`getLessThan()`) y acumularlos (`mvCumSum()`).
  - Manejo de las covariables: acceso al `model.frame` del ajuste y distinguir factores, covariables continuas y pesos (`getModelFrame()`, (`getContVar()`, `getModelCovars()` y `getModelWeights()`).

Implementación

- Covariables:
  - Orden lexicográfico para la comparación.
  - Generar para cada x_i la lista de los x_j que son menores que él (`getLessThan()`) y acumularlos (`mvCumSum()`).
  - Manejo de las covariables: acceso al `model.frame` del ajuste y distinguir factores, covariables continuas y pesos (`getModelFrame()`, (`getContVar()`, `getModelCovars()` y `getModelWeights()`).
- Generación de la muestra Bootstrap(`compBootSamp()`):

Implementación

- Covariables:
  - Orden lexicográfico para la comparación.
  - Generar para cada x_i la lista de los x_j que son menores que él (`getLessThan()`) y acumularlos (`mvCumSum()`).
  - Manejo de las covariables: acceso al `model.frame` del ajuste y distinguir factores, covariables continuas y pesos (`getModelFrame()`, (`getContVar()`, `getModelCovars()` y `getModelWeights()`).
- Generación de la muestra Bootstrap(`compBootSamp()`):
  - Con `rWildBoot()`, `fitted()` y `getResiduals()` se genera la remuestra (x_i^*, y_i^*).

Implementación

- Covariables:
  - Orden lexicográfico para la comparación.
  - Generar para cada x_i la lista de los x_j que son menores que él (`getLessThan()`) y acumularlos (`mvCumSum()`).
  - Manejo de las covariables: acceso al `model.frame` del ajuste y distinguir factores, covariables continuas y pesos (`getModelFrame()`, (`getContVar()`, `getModelCovars()` y `getModelWeights()`).
- Generación de la muestra Bootstrap(`compBootSamp()`):
  - Con `rWildBoot()`, `fitted()` y `getResiduals()` se genera la remuestra (x_i^*, y_i^*).
  - En el `obj$call` correspondiente a H_0 se cambian los datos originales por la remuestra bootstrap.

Implementación

- Covariables:
  - Orden lexicográfico para la comparación.
  - Generar para cada x_i la lista de los x_j que son menores que él (`getLessThan()`) y acumularlos (`mvCumSum()`).
  - Manejo de las covariables: acceso al `model.frame` del ajuste y distinguir factores, covariables continuas y pesos (`getModelFrame()`, (`getContVar()`, `getModelCovars()` y `getModelWeights()`).
- Generación de la muestra Bootstrap(`compBootSamp()`):
  - Con `rWildBoot()`, `fitted()` y `getResiduals()` se genera la remuestra (x_i^*, y_i^*).
  - En el `obj$call` correspondiente a H_0 se cambian los datos originales por la remuestra bootstrap.
  - ...y se reevalúa `obj$call`.

Implementación

- Covariables:
  - Orden lexicográfico para la comparación.
  - Generar para cada x_i la lista de los x_j que son menores que él (`getLessThan()`) y acumularlos (`mvCumSum()`).
  - Manejo de las covariables: acceso al `model.frame` del ajuste y distinguir factores, covariables continuas y pesos (`getModelFrame()`, (`getContVar()`, `getModelCovars()` y `getModelWeights()`).
- Generación de la muestra Bootstrap(`compBootSamp()`):
  - Con `rWildBoot()`, `fitted()` y `getResiduals()` se genera la remuestra (x_i^*, y_i^*).
  - En el `obj$call` correspondiente a H_0 se cambian los datos originales por la remuestra bootstrap.
  - ...y se reevalúa `obj$call`.
  - Cuando `obj` es de la clase `lm` se puede abreviar el cálculo utilizando los cálculo matriciales (`LINMOD=F`).

Implementación

- Covariables:
  - Orden lexicográfico para la comparación.
  - Generar para cada x_i la lista de los x_j que son menores que él (`getLessThan()`) y acumularlos (`mvCumSum()`).
  - Manejo de las covariables: acceso al `model.frame` del ajuste y distinguir factores, covariables continuas y pesos (`getModelFrame()`, (`getContVar()`, `getModelCovars()` y `getModelWeights()`).
- Generación de la muestra Bootstrap(`compBootSamp()`):
  - Con `rWildBoot()`, `fitted()` y `getResiduals()` se genera la remuestra (x_i^*, y_i^*).
  - En el `obj$call` correspondiente a H_0 se cambian los datos originales por la remuestra bootstrap.
  - ...y se reevalúa `obj$call`.
  - Cuando `obj` es de la clase `lm` se puede abreviar el cálculo utilizando los cálculo matriciales (`LINMOD=F`).
  - Después de calcular el proceso (`compIntRegProc()`) se calcula K_n, W_n^2 y se compara con dist. empir. K_{nb}^*, $W_{nb}^{2^*}$.

Implementación

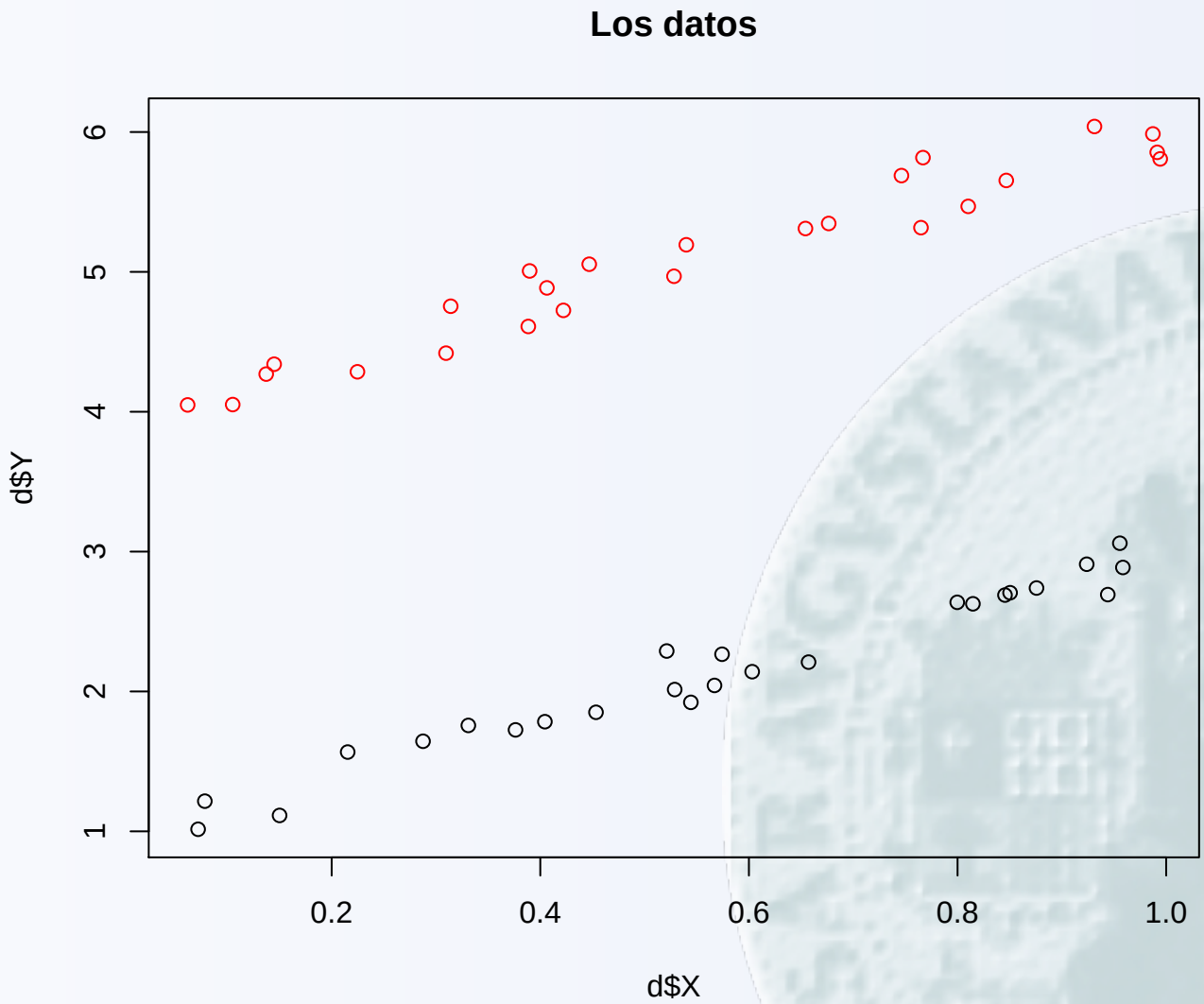
- Covariables:
  - Orden lexicográfico para la comparación.
  - Generar para cada x_i la lista de los x_j que son menores que él (`getLessThan()`) y acumularlos (`mvCumSum()`).
  - Manejo de las covariables: acceso al `model.frame` del ajuste y distinguir factores, covariables continuas y pesos (`getModelFrame()`, (`getContVar()`, `getModelCovars()` y `getModelWeights()`).
- Generación de la muestra Bootstrap(`compBootSamp()`):
  - Con `rWildBoot()`, `fitted()` y `getResiduals()` se genera la remuestra (x_i^*, y_i^*).
  - En el `obj$call` correspondiente a H_0 se cambian los datos originales por la remuestra bootstrap.
  - ...y se reevalúa `obj$call`.
  - Cuando `obj` es de la clase `lm` se puede abreviar el cálculo utilizando los cálculo matriciales (`LINMOD=F`).
  - Después de calcular el proceso (`compIntRegProc()`) se calcula K_n, W_n^2 y se compara con dist. empir. K_{nb}^*, W_{nb}^{2*}.
- Gráficos `plotIntRegProc()`.

Un modelo sencillo I

Selección de un modelo sencillo: *modelo aditivo con un Factor una Covariable*.

```
> n <- 50
> d <- data.frame(X1 = runif(n), F1 = rbinom(n, 1, 0.5))
> d$Y <- 1 + 2 * d$X1 + 3 * d$F1 + rnorm(n, sd = 0.125)
> d$F1 <- as.factor(d$F1)
```

Un modelo sencillo II



Un modelo sencillo III

Algunos modelos para dichos datos:

```
> mA <- lm(Y ~ 1, d)
> mB <- lm(Y ~ X1, d)
> mC <- lm(Y ~ X1 + F1, d)
```

Un modelo sencillo IV

```
> library(intRegGOF, lib.loc = "~/lib")
```

- ¿ Son modelos aceptables para los datos ?

$$H_0 : m \in \mathcal{M}_0 \quad \text{vs.} \quad H_1 : m \notin \mathcal{M}_0.$$

`intRegGOF(mA, B=250)` no tiene sentido, no hay covariables !!

```
> intRegGOF(mA, covars = ~X1, B = 250)

intRegGOF(obj = mA, covars = ~X1, B = 250)
Model Fit Call:
  lm(formula = Y ~ 1, data = d)
Covariates:  X1 .
       value p.value
K   1.933914   0.084
W^2 1.163568   0.072
```

también vale `intRegGOF(mA, covars="X1", B=250)`.

Un modelo sencillo V

```
> intRegGOF(mB, B = 250)

intRegGOF(obj = mB, B = 250)
Model Fit Call:
  lm(formula = Y ~ X1, data = d)
Covariates:  X1 .
         value p.value
K   0.64357986    0.86
W^2 0.07967881    0.80

> intRegGOF(mC, B = 250)

intRegGOF(obj = mC, B = 250)
Model Fit Call:
  lm(formula = Y ~ X1 + F1, data = d)
Covariates:  X1 .
           value p.value
K   0.0566252733   0.764
W^2 0.0008303654   0.524
```

Un modelo sencillo VI

- ¿ Hay alguno mejor que `mA` ?

$$H_0 : m \in \mathcal{M}_0 \quad \text{vs.} \quad H_k : m \in \mathcal{M}_k.$$

```
> anovarIntReg(mA, mB, mC, B = 250)

Integrated Regression Analysis of Variability Table:
  Reference Test Mode
  Model 0: lm(formula = Y ~ 1, data = d)
  Model 1: lm(formula = Y ~ X1, data = d)
  Model 2: lm(formula = Y ~ X1 + F1, data = d)
Covariables: X1, F1

                   K      P(>K)          W P(>W)
Model 0:  1.93391367 0.07200000 1.16356794 0.048
Model 1:  0.64357986 0.02800000 0.07967881 0.020
Model 2:  0.05662527 0.00000000 0.00083037 0.000
```

Un modelo sencillo VII

- ¿ Es mejor que `mA` que `mB` y este que `mC` ?

$$H_0 : m \in \mathcal{M}_k \quad \text{vs.} \quad H_{k+1} : m \in \mathcal{M}_1.$$

```
> anovarIntReg(mA, mB, mC, B = 250, INCREMENTAL = T)

Integrated Regression Analysis of Variability Table:
  Incremental Test Mode
  Model 0: lm(formula = Y ~ 1, data = d)
  Model 1: lm(formula = Y ~ X1, data = d)
  Model 2: lm(formula = Y ~ X1 + F1, data = d)
Covariables: X1, F1

                  K      P(>K)          W P(>W)
Model 0:  1.93391367 0.05200000 1.16356794 0.056
Model 1:  0.64357986 0.02000000 0.07967881 0.020
Model 2:  0.05662527 0.00000000 0.00083037 0.000
```

Un modelo sencillo VIII

- ...Cambiando el orden de los modelos: ¿ Es mejor que `mC` que `mB` y este que `mA` ?

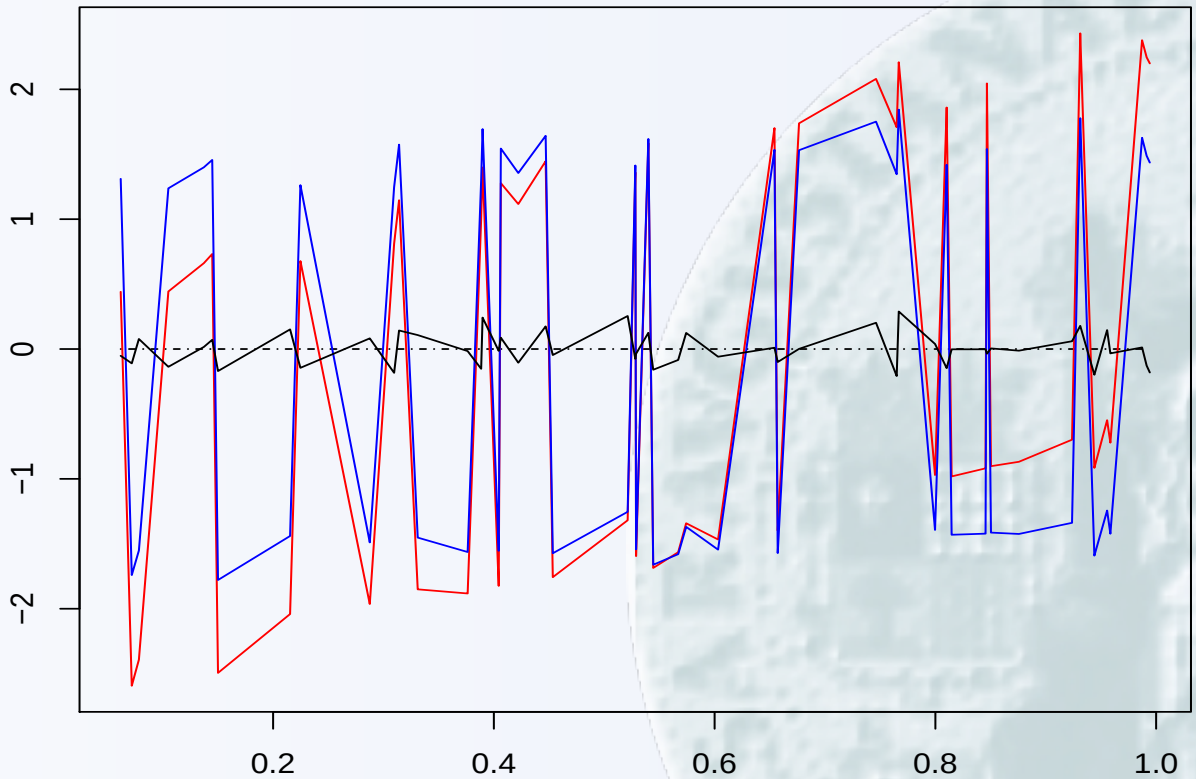
$$H_0 : m \in \mathcal{M}_k \quad \text{vs.} \quad H_{k+1} : m \in \mathcal{M}_1.$$

```
> anovarIntReg(mC, mB, mA, B = 250, INCREMENTAL = T)

Integrated Regression Analysis of Variability Table:
  Incremental Test Mode
  Model 0: lm(formula = Y ~ X1 + F1, data = d)
  Model 1: lm(formula = Y ~ X1, data = d)
  Model 2: lm(formula = Y ~ 1, data = d)
Covariables: X1, F1

                  K      P(>K)          W P(>W)
Model 0:  0.05662527 0.23600000 0.00083037  0.44
Model 1:  0.64357986 0.00000000 0.07967881  0.00
Model 2:  1.93391367 0.00000000 1.16356794  0.00
```

Un modelo sencillo IX

- Gráficamente:

```
plotAsIntRegGOF(mA, covar = ~X1, col="red",type="l")
linesAsIntRegGOF(mB, covar = "X1", col="blue")
linesAsIntRegGOF(mC, covar = ~X1, col="black")
```



Un modelo sencillo X

Datos Sesgados por longitud I

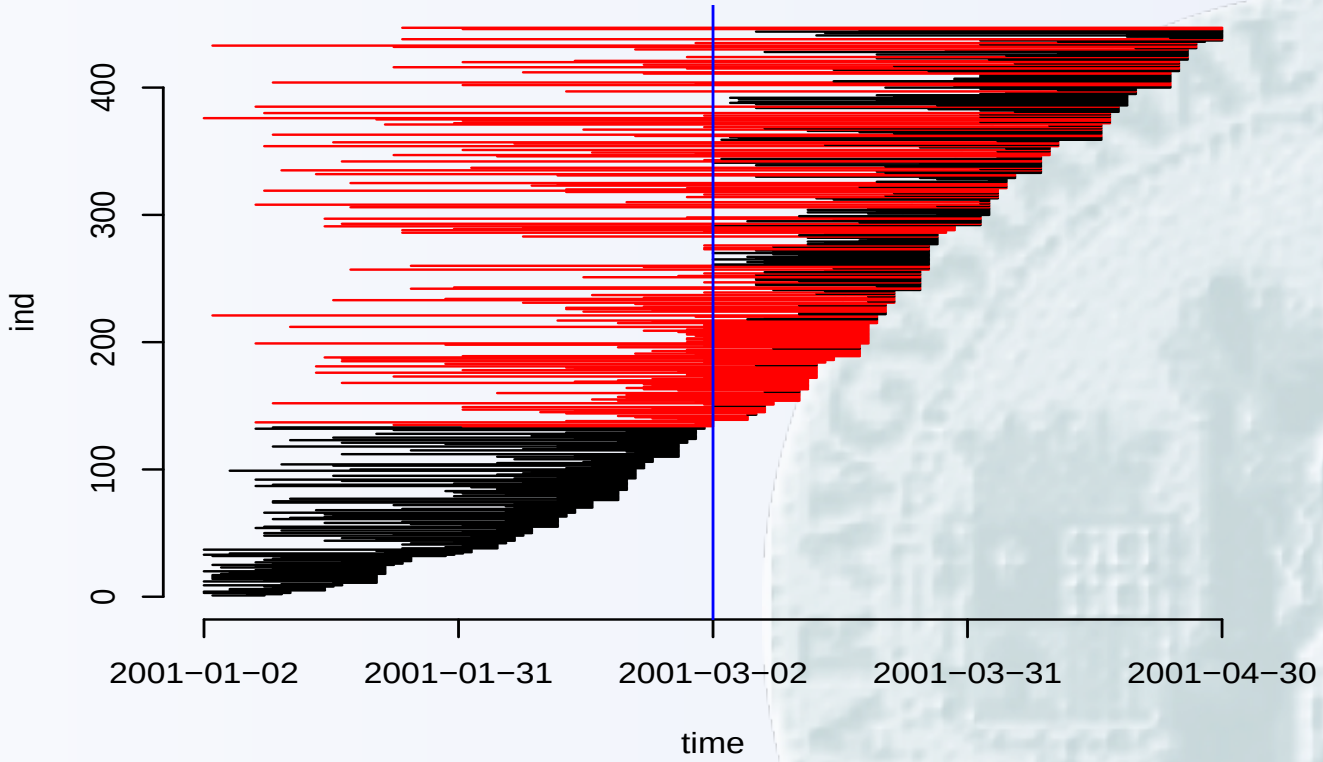
Base de datos del *Servicio Aragonés de Salud*, DGA, de hombres de entre 30 y 85 años de edad a los que les fue prescrita una operación quirúrgica después del *2001-01-01* que fue llevada acabo antes de *2001-04-30* y que comprende las variables:

- `FL`: Fecha en que se prescribe la cirugía.
- `FS`: Fecha en que se realiza la cirugía
- `PR`: Prioridad operación (`preferente`, `normal`).
- `durEsp`: Días de espera hasta la operación. Días entre la prescripción y la realización de la cirugía (`FS-FL`).
- `edad`: Edad del paciente en años.

Considerando esta base de datos como la población, se trata de establecer un modelo que relacione `durEsp` con el resto de las variables cuando la muestra (en `lbsamp`) consiste en aquellos individuos que esperan una operación (n =172 individuos).

Datos Sesgados por longitud II

Duración de la espera, en rojo los individuos de la muestra.
SESGO POR LONGITUD: Duraciones más largas sobrerepresentadas en la muestra.
[van Es et al.(2000)van Es, Klaassen, and Oudshoorn]



Datos Sesgados por longitud III

Comenzamos por un modelo completo:

```
> summary(m0 <- lm(durEsp ~ (edad + PR)^2, lbsamp, weights = 1/lbsamp$durEsp))

Call:
lm(formula = durEsp ~ (edad + PR)^2, data = lbsamp, weights = 1/lbsamp$durEsp)

Residuals:
    Min      1Q  Median      3Q     Max
-17.454  -1.618   1.780   4.225   8.018

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   -4.7662    11.5306  -0.413  0.67987
edad           0.5145     0.1858   2.770  0.00624 **
PRnormal      27.8449    14.3824   1.936  0.05454 .
edad:PRnormal -0.3780     0.2360  -1.602  0.11108
---
Signif. codes:  0
```

...Notar que se usa el parámetro `weigths` para compensar el efecto del *sesgo por longitud*.
...Los test de significación de los parámetros ¡¡ no son válidos !! porque los datos están sesgados por longitud.

Datos Sesgados por longitud IV

Si quitamos uno por uno todos los términos de `m0`:

```
> anovarIntReg(m0, update(m0, ~. - edad:PR), update(m0, ~. - PR),
+     update(m0, ~. - edad), update(m0, ~. - 1))

Integrated Regression Analysis of Variability Table:
  Reference Test Mode
  Model 0: lm(formula = durEsp ~ (edad + PR)^2, data = lbsamp, weights = 1/lbsamp$durEsp)
  Model 1: lm(formula = durEsp ~ edad + PR, data = lbsamp, weights = 1/lbsamp$durEsp)
  Model 2: lm(formula = durEsp ~ edad + edad:PR, data = lbsamp, weights = 1/lbsamp$durEsp)
  Model 3: lm(formula = durEsp ~ PR + edad:PR, data = lbsamp, weights = 1/lbsamp$durEsp)
  Model 4: lm(formula = durEsp ~ edad + PR + edad:PR - 1, data = lbsamp, weights = 1/lbsamp$durEsp)
Covariables: edad, PR

               K   P(>K)       W  P(>W)
Model 0:  0.70090 0.25451 0.11839 0.1904
Model 1:  0.71899 0.23046 0.13220 0.1503
Model 2:  0.74472 0.20641 0.14343 0.1263
Model 3:  0.70090 0.25451 0.11839 0.1904
Model 4:  0.70090 0.25451 0.11839 0.1904
```

Datos Sesgados por longitud V

Partiendo del modelo `update(m0, .-edad:PR)`:

```
> anovarIntReg(m1 <- update(m0, ~. - edad:PR), update(m1, ~. -
+     PR), update(m1, ~. - edad), update(m1, ~. - 1))

Integrated Regression Analysis of Variability Table:
  Reference Test Mode
  Model 0: lm(formula = durEsp ~ edad + PR, data = lbsamp, weights = 1/lbsamp$durEsp)
  Model 1: lm(formula = durEsp ~ edad, data = lbsamp, weights = 1/lbsamp$durEsp)
  Model 2: lm(formula = durEsp ~ PR, data = lbsamp, weights = 1/lbsamp$durEsp)
  Model 3: lm(formula = durEsp ~ edad + PR - 1, data = lbsamp, weights = 1/lbsamp$durEsp)
Covariables: edad, PR

                K   P(>K)       W  P(>W)
Model 0:  0.71899 0.23848 0.13220 0.1423
Model 1:  0.80394 0.14629 0.16634 0.0962
Model 2:  1.41608 0.00000 0.51229 0.0000
Model 3:  0.71899 0.23848 0.13220 0.1423
```

Datos Sesgados por longitud VI

Como parece que quitar edad, lleva a rechazar H_0:m1, mientras que no hay problema si quitamos PR, vamos a considerar el modelo sin interacciones y a quitar cada uno de los términos (intercept incluido):

```
> anovarIntReg(m1, update(m1, ~. - PR), update(m1, ~. - PR - 1))

Integrated Regression Analysis of Variability Table:
  Reference Test Mode
  Model 0: lm(formula = durEsp ~ edad + PR, data = lbsamp, weights = 1/lbsamp$durEsp)
  Model 1: lm(formula = durEsp ~ edad, data = lbsamp, weights = 1/lbsamp$durEsp)
  Model 2: lm(formula = durEsp ~ edad - 1, data = lbsamp, weights = 1/lbsamp$durEsp)
Covariables: edad, PR

                K    P(>K)        W  P(>W)
Model 0:  0.718986 0.252505 0.132205 0.1523
Model 1:  0.803941 0.152305 0.166342 0.0842
Model 2:  1.135077 0.002004 0.356187 0.0000
```

Datos Sesgados por longitud VII

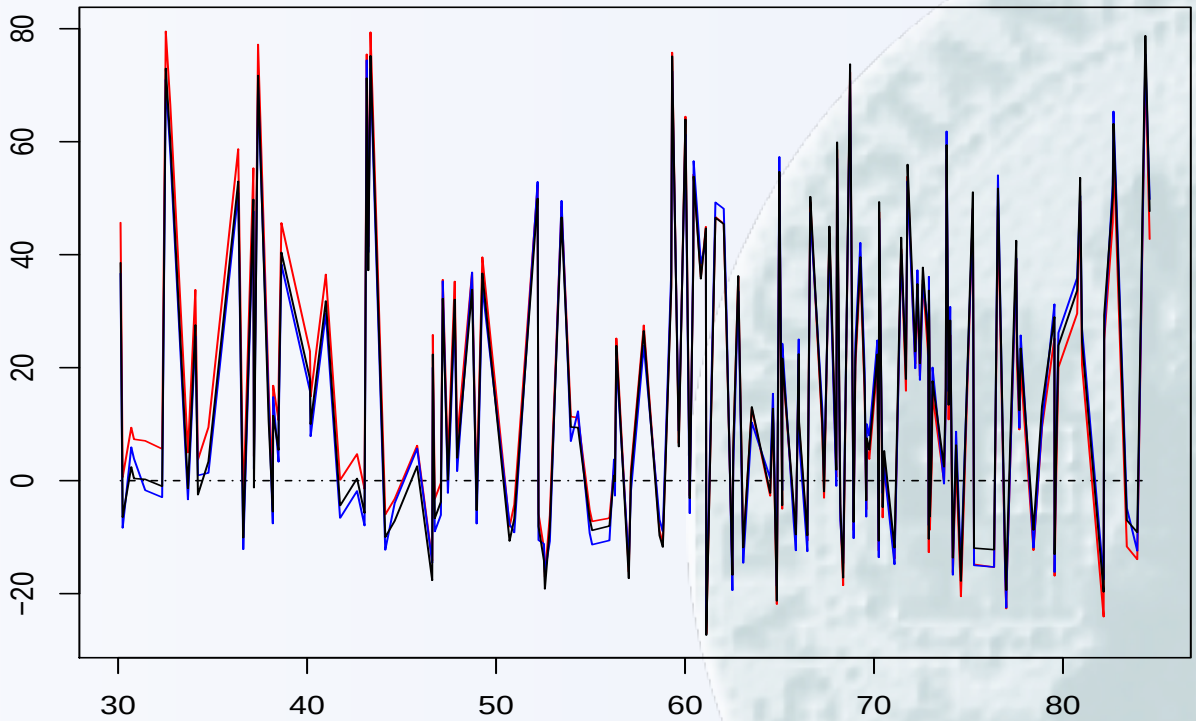
...con lo que los datos responden al modelo durEsp~edad+1:

```
> summary(update(m1, ~. - PR))

Call:
lm(formula = durEsp ~ edad, data = lbsamp, weights = 1/lbsamp$durEsp)

Residuals:
    Min      1Q  Median      3Q     Max
-19.327  -1.257   1.671   4.266   7.519

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  13.7575     6.9403   1.982   0.0491 *
edad          0.2548     0.1147   2.222   0.0276 *
---
Signif. codes:  0
```

Datos Sesgados por longitud VIII

...Para visualizar el proceso R_n^w:

```
plotAsIntRegGOF(update(m1,~.-PR-1), covar = ~edad, col="red",type="l")
linesAsIntRegGOF(m1, covar = "edad", col="blue")
linesAsIntRegGOF(update(m1,~.-PR), covar = ~edad, col="black")
```



Datos Sesgados por longitud IX

...Y si "nos olvidamos" del sesgo por longitud

```
> summary(m1b <- lm(durEsp ~ edad, lbsamp))

Call:
lm(formula = durEsp ~ edad, data = lbsamp)

Residuals:
    Min      1Q  Median      3Q     Max
-43.122 -20.505  -5.159  18.271  61.226

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  24.9907     8.0977   3.086  0.00237 **
edad          0.3293     0.1297   2.540  0.01199 *
---
Signif. codes:  0
```

- Notar la diferencia en la estimación de los parámetros entre `m1` y `m1b`.
- En ambos casos : Notar la diferencia de los *p*–valores con los calculados por `intRegGOF` (prácticamente nulos al quitar tanto `PR` como el intercept(1)).

Censura I

Datos de medidas sobre estrellas

- `Type`: Tipo de estrella con o sin planeta (conocido)* (`planet`, `no planet`).
- `Teff`: Temperatura estrella (grados Kelvin).
- `IndBe`: Indicadoe de censura (1=Obs. no censurada, 0=Obs. censurada).
- `logNBe`: log de la abundancia de Berilio, scalada con la del sol ($\log N(Be) = 0$).

Las medidas se obtienen a partir del espectro de la estrella. Debido a errores en los equipos de medición y otras causas, se supone que los datos están *censurados*. Se trata de desarrollar un modelo que relaciones `logNBe` con `Type` y `Teff`. `logNBe` se considera un indicador de la existencia de metal en la estrella, que a su vez es indicador de la posible existencia de planetas.
OBSERVACIÓN: El sesgo en los datos no es sólo censura, posiblemente se deba además y entre otras causas a la distancia a la estrella...
Los datos, que están en `d` comprenden 68 estrellas, 39 con planeta y 29 sin planeta.
Una explicación más detallada se pueden obtener en
`http://astrostatistics.psu.edu/datasets/censor.html`

Censura II

Estimación Kaplan–Meier de la función de distribución de `logNBe`, en realidad de la "masa de probabilidad" de cada observación:

```
> kmw <- kmEst(d$logNBe, d$IndBe)
```

ó con el paquete `survival`

```
> library(survival)
> fit <- survfit(Surv(d$logNBe, d$IndBe) ~ 1, type = "kaplan-meier")
> res <- as.data.frame(cbind(fit$time, fit$surv))
> colnames(res) <- c("z", "kmSurv")
> res <- res[!rev(duplicated(rev(res$kmSurv))), ]
> res$kmWeig <- -diff(c(1, res$kmSurv))
```

Censura III

Como las únicas observaciones disponible sobre el `logNBe` son aquellas para las que `IndBe` vale 1, y la masa de probabilidad debe corresponderse con su valor, seleccionamos según `kmw`.

```
> beSamp <- d[as.numeric(rownames(kmw)), ]
```

y con ellos comenzamos por un modelo completo:

```
> summary(m0 <- lm(logNBe ~ (Teff + Type)^2, data = beSamp, weights = kmw$kmWeig))

Call:
lm(formula = logNBe ~ (Teff + Type)^2, data = beSamp, weights = kmw$kmWeig)

Residuals:
     Min       1Q   Median       3Q      Max
-0.08465 -0.01199  0.00193  0.01515  0.04106

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)     -9.749e-01  5.738e-01  -1.699  0.09768 .
Teff             3.425e-04  1.009e-04   3.394  0.00166 **
Typeplanet      -3.188e-01  9.853e-01  -0.324  0.74807
Teff:Typeplanet  6.488e-05  1.703e-04   0.381  0.70547
---
Signif. codes:  0
```

Censura IV

Examinamos qué ocurre al eliminar términos:

```
> anovarIntReg(m0, update(m0, ~. - Teff:Type), update(m0, ~. -
+     Type), update(m0, ~. - Teff), update(m0, ~. - 1))

Integrated Regression Analysis of Variability Table:
  Reference Test Mode
  Model 0: lm(formula = logNBe ~ (Teff + Type)^2, data = beSamp, weights = kmw$kmWeig)
  Model 1: lm(formula = logNBe ~ Teff + Type, data = beSamp, weights = kmw$kmWeig)
  Model 2: lm(formula = logNBe ~ Teff + Teff:Type, data = beSamp, weights = kmw$kmWeig)
  Model 3: lm(formula = logNBe ~ Type + Teff:Type, data = beSamp, weights = kmw$kmWeig)
  Model 4: lm(formula = logNBe ~ Teff + Type + Teff:Type - 1, data = beSamp, weights = kmw$kmWeig)
Covariables: Teff, Type

                  K      P(>K)          W  P(>W)
Model 0:  3.0781e-03 4.8096e-02 2.0658e-06 0.0441
Model 1:  3.0098e-03 5.6112e-02 1.9718e-06 0.0581
Model 2:  3.0208e-03 5.2104e-02 1.9870e-06 0.0561
Model 3:  3.0781e-03 4.8096e-02 2.0658e-06 0.0441
Model 4:  3.0781e-03 4.8096e-02 2.0658e-06 0.0441
```

...parece que se puede eliminar Type

Censura V

...y efectivamente:

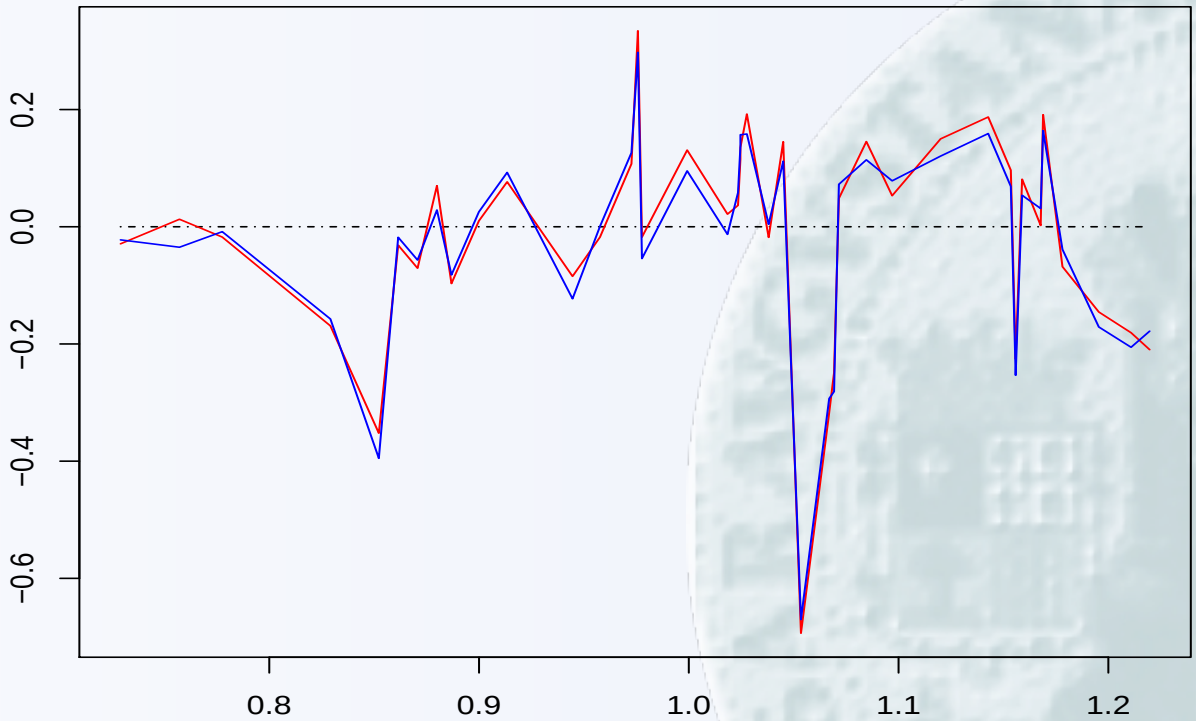
```
> anovarIntReg(m1 <- update(m0, ~. - Teff:Type), mOk <- update(m1,
+     ~. - Type), update(m1, ~. - Teff - Type), update(m1, ~. -
+     Type - 1))

Integrated Regression Analysis of Variability Table:
  Reference Test Mode
  Model 0: lm(formula = logNBe ~ Teff + Type, data = beSamp, weights = kmw$kmWeig)
  Model 1: lm(formula = logNBe ~ Teff, data = beSamp, weights = kmw$kmWeig)
  Model 2: lm(formula = logNBe ~ 1, data = beSamp, weights = kmw$kmWeig)
  Model 3: lm(formula = logNBe ~ Teff - 1, data = beSamp, weights = kmw$kmWeig)
Covariables: Teff, Type

                  K      P(>K)          W  P(>W)
Model 0:  3.0098e-03 3.8076e-02 1.9718e-06 0.0381
Model 1:  2.9721e-03 4.6092e-02 1.9087e-06 0.0461
Model 2:  8.9994e-03 0.0000e+00 3.2088e-05 0.0000
Model 3:  5.8867e-03 0.0000e+00 1.1284e-05 0.0000
```

el modelo que parece adecuado es `logNBe~Teff+1`.

Censura VI

...gráficamente, y utilizando `fitted...` como covariable:

```
plotAsIntRegGOF(mOk, covar = fitted(mOk),col="red",type="l")
linesAsIntRegGOF(m1, covar = fitted(mOk), col="blue")
```



...Por hacer

...Por hacer

- Testeo:

...Por hacer

- Testeo:
  - Testeo intensivo, comparando los ajustes `lm`, `glm`, `nls`.

...Por hacer

- Testeo:
  - Testeo intensivo, comparando los ajustes `lm`, `glm`, `nls`.
  - Se pueden incluir ¿¡ `nlme` !?, ¿¡ `survfit` y similares !?, ¿¡ `locpol` !?, ¿¡ `gam` !?...

...Por hacer

- Testeo:
  - Testeo intensivo, comparando los ajustes `lm`, `glm`, `nls`.
  - Se pueden incluir ¿¡ `nlme` !?, ¿¡ `survfit` y similares !?, ¿¡ `locpol` !?, ¿¡ `gam` !?...
  - Comparación con otros métodos (*sin sesgo*).

...Por hacer

- Testeo:
  - Testeo intensivo, comparando los ajustes `lm`, `glm`, `nls`.
  - Se pueden incluir ¿¡ `nlme` !?, ¿¡ `survfit` y similares !?, ¿¡ `locpol` !?, ¿¡ `gam` !?...
  - Comparación con otros métodos (*sin sesgo*).
  - Conj. de datos de referencia...

...Por hacer

- Testeo:
  - Testeo intensivo, comparando los ajustes `lm`, `glm`, `nls`.
  - Se pueden incluir ¿¡ `nlme` !?, ¿¡ `survfit` y similares !?, ¿¡ `locpol` !?, ¿¡ `gam` !?...
  - Comparación con otros métodos (*sin sesgo*).
  - Conj. de datos de referencia...
- Mejoras en el código:

...Por hacer

- Testeo:
  - Testeo intensivo, comparando los ajustes `lm`, `glm`, `nls`.
  - Se pueden incluir ¿¡ `nlme` !?, ¿¡ `survfit` y similares !?, ¿¡ `locpol` !?, ¿¡ `gam` !?...
  - Comparación con otros métodos (*sin sesgo*).
  - Conj. de datos de referencia...
- Mejoras en el código:
  - Códico más eficiente: `getLessThan()`,...

...Por hacer

- Testeo:
  - Testeo intensivo, comparando los ajustes `lm`, `glm`, `nls`.
  - Se pueden incluir ¿¡ `nlme` !?, ¿¡ `survfit` y similares !?, ¿¡ `locpol` !?, ¿¡ `gam` !?...
  - Comparación con otros métodos (*sin sesgo*).
  - Conj. de datos de referencia...
- Mejoras en el código:
  - Códico más eficiente: `getLessThan()`,...
  - Permitir *cualquier estadístico* sobre el proceso R_n^w.

...Por hacer

- Testeo:
  - Testeo intensivo, comparando los ajustes `lm`, `glm`, `nls`.
  - Se pueden incluir ¿¡ `nlme` !?, ¿¡ `survfit` y similares !?, ¿¡ `locpol` !?, ¿¡ `gam` !?...
  - Comparación con otros métodos (*sin sesgo*).
  - Conj. de datos de referencia...
- Mejoras en el código:
  - Códico más eficiente: `getLessThan()`,...
  - Permitir *cualquier estadístico* sobre el proceso R_n^w.
  - Añadir métodos de selección automática de variables (`add1()`, `drop1()`,`step()`,...).

...Por hacer

- Testeo:
  - Testeo intensivo, comparando los ajustes `lm`, `glm`, `nls`.
  - Se pueden incluir ¿¡ `nlme` !?, ¿¡ `survfit` y similares !?, ¿¡ `locpol` !?, ¿¡ `gam` !?...
  - Comparación con otros métodos (*sin sesgo*).
  - Conj. de datos de referencia...
- Mejoras en el código:
  - Códico más eficiente: `getLessThan()`,...
  - Permitir *cualquier estadístico* sobre el proceso R_n^w.
  - Añadir métodos de selección automática de variables (`add1()`, `drop1()`,`step()`,...).
  - ...clases S4

...Por hacer

- Testeo:
  - Testeo intensivo, comparando los ajustes `lm`, `glm`, `nls`.
  - Se pueden incluir ¿¡ `nlme` !?, ¿¡ `survfit` y similares !?, ¿¡ `locpol` !?, ¿¡ `gam` !?...
  - Comparación con otros métodos (*sin sesgo*).
  - Conj. de datos de referencia...
- Mejoras en el código:
  - Códico más eficiente: `getLessThan()`,...
  - Permitir *cualquier estadístico* sobre el proceso R_n^w.
  - Añadir métodos de selección automática de variables (`add1()`, `drop1()`,`step()`,...).
  - ...clases S4
- Gráficos:

...Por hacer

- Testeo:
  - Testeo intensivo, comparando los ajustes `lm`, `glm`, `nls`.
  - Se pueden incluir ¿¡ `nlme` !?, ¿¡ `survfit` y similares !?, ¿¡ `locpol` !?, ¿¡ `gam` !?...
  - Comparación con otros métodos (*sin sesgo*).
  - Conj. de datos de referencia...
- Mejoras en el código:
  - Códico más eficiente: `getLessThan()`,...
  - Permitir *cualquier estadístico* sobre el proceso R_n^w.
  - Añadir métodos de selección automática de variables (`add1()`, `drop1()`,`step()`,...).
  - ...clases S4
- Gráficos:
  - Gráficos condicionales según una variable discreta.

...Por hacer

- Testeo:
  - Testeo intensivo, comparando los ajustes `lm`, `glm`, `nls`.
  - Se pueden incluir ¿¡ `nlme` !?, ¿¡ `survfit` y similares !?, ¿¡ `locpol` !?, ¿¡ `gam` !?...
  - Comparación con otros métodos (*sin sesgo*).
  - Conj. de datos de referencia...
- Mejoras en el código:
  - Códico más eficiente: `getLessThan()`,...
  - Permitir *cualquier estadístico* sobre el proceso R_n^w.
  - Añadir métodos de selección automática de variables (`add1()`, `drop1()`,`step()`,...).
  - ...clases S4
- Gráficos:
  - Gráficos condicionales según una variable discreta.
  - ... ¿ Comportamiento multidimensional ?

...Por hacer

- Testeo:
  - Testeo intensivo, comparando los ajustes `lm`, `glm`, `nls`.
  - Se pueden incluir ¿¡ `nlme` !?, ¿¡ `survfit` y similares !?, ¿¡ `locpol` !?, ¿¡ `gam` !?...
  - Comparación con otros métodos (*sin sesgo*).
  - Conj. de datos de referencia...
- Mejoras en el código:
  - Códico más eficiente: `getLessThan()`,...
  - Permitir *cualquier estadístico* sobre el proceso R_n^w.
  - Añadir métodos de selección automática de variables (`add1()`, `drop1()`,`step()`,...).
  - ...clases S4
- Gráficos:
  - Gráficos condicionales según una variable discreta.
  - ... ¿ Comportamiento multidimensional ?
- ¿ ...Para variables discretas ?

...Por hacer

- Testeo:
  - Testeo intensivo, comparando los ajustes `lm`, `glm`, `nls`.
  - Se pueden incluir ¿¡ `nlme` !?, ¿¡ `survfit` y similares !?, ¿¡ `locpol` !?, ¿¡ `gam` !?...
  - Comparación con otros métodos (*sin sesgo*).
  - Conj. de datos de referencia...
- Mejoras en el código:
  - Códico más eficiente: `getLessThan()`,...
  - Permitir *cualquier estadístico* sobre el proceso R_n^w.
  - Añadir métodos de selección automática de variables (`add1()`, `drop1()`,`step()`,...).
  - ...clases S4
- Gráficos:
  - Gráficos condicionales según una variable discreta.
  - ... ¿ Comportamiento multidimensional ?
- ¿ ...Para variables discretas ?
- ...

Muchas Gracias

Bibliografía I

Cohen, A., Kemperman, J. H. B., and Sackrowitz, H. (2002).

On the bias in estimating genetic length and other quantities in simplex constrained models.

Ann. Statist., **30**(1), pp. 202–219.
URL `http://dx.doi.org/10.1214/aos/1015362190`

Delgado, M. A. and González Manteiga, W. (2001).

Significance testing in nonparametric regression based on the bootstrap.

Ann. Statist., **29**(5), pp. 1469–1507.

van Es, B., Klaassen, C. A. J., and Oudshoorn, K. (2000).

Survival analysis under cross-sectional sampling: length bias and multiplicative censoring.

Journal of Statistical Planning and Inference, **91**(2), pp. 295–312.
URL `http://www.sciencedirect.com/science/article/B6V0M-41Y8KJM-8/1/8664946cf1cec647202f83a01dd61e13`

Härdle, W. and Mammen, E. (1993).

Comparing nonparametric versus parametric regression fits.

Ann. Statist., **21**(4), pp. 1926–1947.

Hart, J. D. (1997).

Nonparametric smoothing and lack–of–fit tests.

Springer Series in Statistics. Springer-Verlag, New York.

Bibliografía II

van Keilegom, I., Sánchez Sellero, C., and González-Manteiga, W. (2007).

Goodness–of–fit test in parametric regression based on the estimation of the error distribution.

TEST.

Kozek, A. S. (1990).

A nonparametric test of fit of a linear model.

Comm. Statist. Theory Methods, **19**(1), pp. 169–179.

Ojeda, J., W., G.-M., and Cristóbal, J. (2007).

A bootstrap based model checking for selection-biased data.

Technical report, Reports in Statistics and Operations Research.
URL
http://eio.usc.es/eipc1/BASE/BASEMASTER/FORMULARIOS-PHP-DPTO/REPORTS/447report07_05.pdf

Oshlack, A. and Wakefield, M. (2009).

Transcript length bias in rna-seq data confounds systems biology.

Biology Direct, **4**(1), p. 14.
URL http://www.biology-direct.com/content/4/1/14

Patil, G. (2002).

Weigthed distributions.

Encyclopedia of Environmetrics, **4**, pp. 2369–2377.

Bibliografía III

Patil, G. P. and Rao, C. R. (1978).

Weighted distributions and size-biased sampling with applications to wildlife populations and human families.

Biometrics, **34**(2), pp. 179–189.

Quesenberry, Jr., C. P. and Jewell, N. P. (1986).

Regression analysis based on stratified samples.

Biometrika, **73**(3), pp. 605–614.

Stute, W. (1997).

Nonparametric model checks for regression.

Ann. Statist., **25**(2), pp. 613–641.