MONASH University

Rob J Hyndman

Functional time series

with applications in demography

5. Forecasting functional time series via PLS

Outline

Functional PLS

- PCA components are designed to explain historical variation. They do not necessarily provide the best predictors.
- Partial least squares extracts uncorrelated latent components cores by maximizing the covariance between predictors and response.
- Response is $s_t(x)$ and predictor is $s_{t-1}(x)$.
- We want to predict $s_t(x)$ using
- This is a functional ARH(1).
- How to choose $b(x,u)$?

Functional PLS

- PCA components are designed to explain historical variation. They do not necessarily provide the best predictors.
- Partial least squares extracts uncorrelated latent components cores by maximizing the covariance between predictors and response.
- Response is $s_t(x)$ and predictor is $s_{t-1}(x)$.
- We want to predict $s_t(x)$ using

$$\hat{s}_{T+1|T}(x) = \hat{\mu}(x) + \int [s_T(x) - \hat{\mu}(x)]\, b(x,u)\, du$$

- This is a functional ARH(1).
- How to choose $b(x,u)$?

Functional PLS

- PCA components are designed to explain historical variation. They do not necessarily provide the best predictors.
- Partial least squares extracts uncorrelated latent components cores by maximizing the covariance between predictors and response.
- Response is $s_t(x)$ and predictor is $s_{t-1}(x)$.
- We want to predict $s_t(x)$ using

$$\hat{s}_{T+1|T}(x) = \hat{\mu}(x) + \int [s_T(x) - \hat{\mu}(x)]\, b(x,u)\, du$$

- This is a functional ARH(1).
- How to choose $b(x,u)$?

Functional PLS

- PCA components are designed to explain historical variation. They do not necessarily provide the best predictors.
- Partial least squares extracts uncorrelated latent components cores by maximizing the covariance between predictors and response.
- Response is $s_t(x)$ and predictor is $s_{t-1}(x)$.
- We want to predict $s_t(x)$ using

$$\hat{s}_{T+1|T}(x) = \hat{\mu}(x) + \int [s_T(x) - \hat{\mu}(x)]\, b(x,u)\, du$$

- This is a functional ARH(1).
- How to choose $b(x,u)$?

Functional PLS

- PCA components are designed to explain historical variation. They do not necessarily provide the best predictors.
- Partial least squares extracts uncorrelated latent components cores by maximizing the covariance between predictors and response.
- Response is $s_t(x)$ and predictor is $s_{t-1}(x)$.
- We want to predict $s_t(x)$ using

$$\hat{s}_{T+1|T}(x) = \hat{\mu}(x) + \int [s_T(x) - \hat{\mu}(x)]\, b(x,u)\, du$$

- This is a functional ARH(1).
- How to choose $b(x,u)$?

Functional PLS

- PCA components are designed to explain historical variation. They do not necessarily provide the best predictors.
- Partial least squares extracts uncorrelated latent components cores by maximizing the covariance between predictors and response.
- Response is $s_t(x)$ and predictor is $s_{t-1}(x)$.
- We want to predict $s_t(x)$ using

$$\hat{s}_{T+1|T}(x) = \hat{\mu}(x) + \int [s_T(x) - \hat{\mu}(x)]\, b(x,u)\, du$$

- This is a functional ARH(1).
- How to choose $b(x,u)$?

Functional PLS

- PCA components are designed to explain historical variation. They do not necessarily provide the best predictors.
- Partial least squares extracts uncorrelated latent components cores by maximizing the covariance between predictors and response.
- Response is $s_t(x)$ and predictor is $s_{t-1}(x)$.
- We want to predict $s_t(x)$ using

$$\hat{s}_{T+1|T}(x) = \hat{\mu}(x) + \int [s_T(x) - \hat{\mu}(x)]\, b(x,u)\, du$$

- This is a functional ARH(1).
- How to choose $b(x,u)$?

Functional weighted PLS

Outer relationship

$$\hat{\boldsymbol{f}}^*(x) = \sum_{k=1}^{\infty} \beta_k \psi_k(x), \quad \text{and} \quad \hat{\boldsymbol{g}}^*(x) = \sum_{k=1}^{\infty} \beta_k \phi_k(x)$$

- $\boldsymbol{W} = \text{diagonal}(w_1, \dots, w_T), \quad w_t = \kappa(1-\kappa)^{T-t}$
- $\boldsymbol{f}^*(x) = \boldsymbol{W}[s_1^*(x), \dots, s_{T-1}^*(x)]'$ and $\boldsymbol{g}^*(x) = \boldsymbol{W}[s_2^*(x), \dots, s_T^*(x)]'$ are weighted decentralized functional predictors and responses
- β_k denotes common latent component scores
- $\psi_k(x)$ and $\phi_k(x)$ are latent components of predictors and responses respectively

Functional weighted PLS

Outer relationship

$$\hat{\boldsymbol{f}}^*(x) = \sum_{k=1}^{\infty} \beta_k \psi_k(x), \quad \text{and} \quad \hat{\boldsymbol{g}}^*(x) = \sum_{k=1}^{\infty} \beta_k \phi_k(x)$$

- $\boldsymbol{W} = \text{diagonal}(w_1, \dots, w_T), \quad w_t = \kappa(1-\kappa)^{T-t}$
- $\boldsymbol{f}^*(x) = \boldsymbol{W}[s_1^*(x), \dots, s_{T-1}^*(x)]'$ and $\boldsymbol{g}^*(x) = \boldsymbol{W}[s_2^*(x), \dots, s_T^*(x)]'$ are weighted decentralized functional predictors and responses
- β_k denotes common latent component scores
- $\psi_k(x)$ and $\phi_k(x)$ are latent components of predictors and responses respectively

Functional weighted PLS

Outer relationship

$$\hat{\boldsymbol{f}}^*(x) = \sum_{k=1}^{\infty} \beta_k \psi_k(x), \quad \text{and} \quad \hat{\boldsymbol{g}}^*(x) = \sum_{k=1}^{\infty} \beta_k \phi_k(x)$$

- $\boldsymbol{W} = \text{diagonal}(w_1, \dots, w_T), \quad w_t = \kappa(1-\kappa)^{T-t}$
- $\boldsymbol{f}^*(x) = \boldsymbol{W}[s_1^*(x), \dots, s_{T-1}^*(x)]'$ and $\boldsymbol{g}^*(x) = \boldsymbol{W}[s_2^*(x), \dots, s_T^*(x)]'$ are weighted decentralized functional predictors and responses
- β_k denotes common latent component scores
- $\psi_k(x)$ and $\phi_k(x)$ are latent components of predictors and responses respectively

Functional weighted PLS

Outer relationship

$$\hat{\boldsymbol{f}}^*(x) = \sum_{k=1}^{\infty} \boldsymbol{\beta}_k \psi_k(x), \quad \text{and} \quad \hat{\boldsymbol{g}}^*(x) = \sum_{k=1}^{\infty} \boldsymbol{\beta}_k \phi_k(x)$$

- $\boldsymbol{W} = \text{diagonal}(w_1, \dots, w_T), \quad w_t = \kappa(1-\kappa)^{T-t}$
- $\boldsymbol{f}^*(x) = \boldsymbol{W}[s_1^*(x), \dots, s_{T-1}^*(x)]'$ and $\boldsymbol{g}^*(x) = \boldsymbol{W}[s_2^*(x), \dots, s_T^*(x)]'$ are weighted decentralized functional predictors and responses
- $\boldsymbol{\beta}_k$ denotes common latent component scores
- $\psi_k(x)$ and $\phi_k(x)$ are latent components of predictors and responses respectively

Functional weighted PLS

$$\beta_k = \int \boldsymbol{f}^*(x) w_k(x) dx = \int \boldsymbol{g}^*(x) m_k(x)\, dx,$$

Compute β_k iteratively:

1. Let $\boldsymbol{f}_0^*(x) = \boldsymbol{f}^*(x)$ and $\boldsymbol{g}_0^*(x) = \boldsymbol{g}^*(x)$
2. Obtain $w_k(x)$ iteratively, starting with $w_k^{(0)}(x) = 1$:
   $w_k^{(i)}(x) = \iint w_k^{(i-1)}(v)[\hat{\boldsymbol{f}}_{k-1}^*(v)]' \hat{\boldsymbol{g}}_{k-1}^*(u)[\hat{\boldsymbol{g}}_{k-1}^*(u)]' \hat{\boldsymbol{f}}_{k-1}^*(x)\, dv\, du$
3. $\beta_k = \int \boldsymbol{f}^*(x) w_k(x) dx.$
4. $\hat{\boldsymbol{f}}_k^*(x) = (\boldsymbol{I} - \beta_k \beta_k') \hat{\boldsymbol{f}}_{k-1}^*(x)$
5. $\hat{\boldsymbol{g}}_k^*(x) = (\boldsymbol{I} - \beta_k \beta_k') \hat{\boldsymbol{g}}_{k-1}^*(x)$

Functional weighted PLS

$$\beta_k = \int \boldsymbol{f}^*(x) w_k(x) dx = \int \boldsymbol{g}^*(x) m_k(x)\, dx,$$

Compute β_k iteratively:

1. Let $\boldsymbol{f}_0^*(x) = \boldsymbol{f}^*(x)$ and $\boldsymbol{g}_0^*(x) = \boldsymbol{g}^*(x)$
2. Obtain $w_k(x)$ iteratively, starting with $w_k^{(0)}(x) = 1$:
$$w_k^{(i)}(x) = \iint w_k^{(i-1)}(v)[\hat{\boldsymbol{f}}_{k-1}^*(v)]' \hat{\boldsymbol{g}}_{k-1}^*(u)[\hat{\boldsymbol{g}}_{k-1}^*(u)]' \hat{\boldsymbol{f}}_{k-1}^*(x)\, dv\, du$$
3. $\beta_k = \int \boldsymbol{f}^*(x) w_k(x) dx.$
4. $\hat{\boldsymbol{f}}_k^*(x) = (\boldsymbol{I} - \beta_k \beta_k') \hat{\boldsymbol{f}}_{k-1}^*(x)$
5. $\hat{\boldsymbol{g}}_k^*(x) = (\boldsymbol{I} - \beta_k \beta_k') \hat{\boldsymbol{g}}_{k-1}^*(x)$

Functional weighted PLS

$$\beta_k = \int \boldsymbol{f}^*(x) w_k(x) dx = \int \boldsymbol{g}^*(x) m_k(x)\, dx,$$

Compute β_k iteratively:

1. Let $\boldsymbol{f}_0^*(x) = \boldsymbol{f}^*(x)$ and $\boldsymbol{g}_0^*(x) = \boldsymbol{g}^*(x)$
2. Obtain $w_k(x)$ iteratively, starting with $w_k^{(0)}(x) = 1$:
$$w_k^{(i)}(x) = \iint w_k^{(i-1)}(v) [\hat{\boldsymbol{f}}_{k-1}^*(v)]' \hat{\boldsymbol{g}}_{k-1}^*(u) [\hat{\boldsymbol{g}}_{k-1}^*(u)]' \hat{\boldsymbol{f}}_{k-1}^*(x)\, dv\, du$$
3. $\beta_k = \int \boldsymbol{f}^*(x) w_k(x) dx.$
4. $\hat{\boldsymbol{f}}_k^*(x) = (\boldsymbol{I} - \beta_k \beta_k') \hat{\boldsymbol{f}}_{k-1}^*(x)$
5. $\hat{\boldsymbol{g}}_k^*(x) = (\boldsymbol{I} - \beta_k \beta_k') \hat{\boldsymbol{g}}_{k-1}^*(x)$

Functional weighted PLS

$$\beta_k = \int \boldsymbol{f}^*(x) w_k(x) dx = \int \boldsymbol{g}^*(x) m_k(x)\, dx,$$

Compute β_k iteratively:

1. Let $\boldsymbol{f}_0^*(x) = \boldsymbol{f}^*(x)$ and $\boldsymbol{g}_0^*(x) = \boldsymbol{g}^*(x)$
2. Obtain $w_k(x)$ iteratively, starting with $w_k^{(0)}(x) = 1$:
   $$w_k^{(i)}(x) = \iint w_k^{(i-1)}(v) [\hat{\boldsymbol{f}}_{k-1}^*(v)]' \hat{\boldsymbol{g}}_{k-1}^*(u) [\hat{\boldsymbol{g}}_{k-1}^*(u)]' \hat{\boldsymbol{f}}_{k-1}^*(x)\, dv\, du$$
3. $\beta_k = \int \boldsymbol{f}^*(x) w_k(x) dx.$
4. $\hat{\boldsymbol{f}}_k^*(x) = (\boldsymbol{I} - \beta_k \beta_k') \hat{\boldsymbol{f}}_{k-1}^*(x)$
5. $\hat{\boldsymbol{g}}_k^*(x) = (\boldsymbol{I} - \beta_k \beta_k') \hat{\boldsymbol{g}}_{k-1}^*(x)$

Functional weighted PLS

$$\beta_k = \int \boldsymbol{f}^*(x) w_k(x) dx = \int \boldsymbol{g}^*(x) m_k(x)\, dx,$$

Compute β_k iteratively:

1. Let $\boldsymbol{f}_0^*(x) = \boldsymbol{f}^*(x)$ and $\boldsymbol{g}_0^*(x) = \boldsymbol{g}^*(x)$
2. Obtain $w_k(x)$ iteratively, starting with $w_k^{(0)}(x) = 1$:
$$w_k^{(i)}(x) = \iint w_k^{(i-1)}(v)[\hat{\boldsymbol{f}}_{k-1}^*(v)]'\hat{\boldsymbol{g}}_{k-1}^*(u)[\hat{\boldsymbol{g}}_{k-1}^*(u)]'\hat{\boldsymbol{f}}_{k-1}^*(x)\, dv\, du$$
3. $\beta_k = \int \boldsymbol{f}^*(x) w_k(x) dx.$
4. $\hat{\boldsymbol{f}}_k^*(x) = (\boldsymbol{I} - \beta_k \beta_k')\hat{\boldsymbol{f}}_{k-1}^*(x)$
5. $\hat{\boldsymbol{g}}_k^*(x) = (\boldsymbol{I} - \beta_k \beta_k')\hat{\boldsymbol{g}}_{k-1}^*(x)$

Functional weighted PLS

Computationally equivalent approach

- Discretize $s_t^*(x)$ on a dense grid of q equally spaced points.
- Denote discretized $s_t^*(x)$ as $T \times q$ matrix $\boldsymbol{G}^*$ and let $\boldsymbol{G} = \boldsymbol{W}\boldsymbol{G}^*$.
- Define $\boldsymbol{G}_{k-1}$ and $\boldsymbol{F}_{k-1}$ analogously
- $w_1(x) =$ largest eigenvector of $\boldsymbol{G}'\boldsymbol{F}$
- $m_k(x) =$ largest eigenvector of $\boldsymbol{G}'_{k-1}\boldsymbol{F}_{k-1}$
- $w_k(x) =$ largest eigenvector of $\boldsymbol{F}'_{k-1}\boldsymbol{G}_{k-1}$

Functional weighted PLS

Computationally equivalent approach

- Discretize $s_t^*(x)$ on a dense grid of q equally spaced points.
- Denote discretized $s_t^*(x)$ as $T \times q$ matrix $\boldsymbol{G}^*$ and let $\boldsymbol{G} = \boldsymbol{W}\boldsymbol{G}^*$.
- Define $\boldsymbol{G}_{k-1}$ and $\boldsymbol{F}_{k-1}$ analogously
- $w_1(x) =$ largest eigenvector of $\boldsymbol{G}'\boldsymbol{F}$
- $m_k(x) =$ largest eigenvector of $\boldsymbol{G}_{k-1}'\boldsymbol{F}_{k-1}$
- $w_k(x) =$ largest eigenvector of $\boldsymbol{F}_{k-1}'\boldsymbol{G}_{k-1}$

Functional weighted PLS

Computationally equivalent approach

- Discretize $s_t^*(x)$ on a dense grid of q equally spaced points.
- Denote discretized $s_t^*(x)$ as $T \times q$ matrix $\boldsymbol{G}^*$ and let $\boldsymbol{G} = \boldsymbol{W}\boldsymbol{G}^*$.
- Define $\boldsymbol{G}_{k-1}$ and $\boldsymbol{F}_{k-1}$ analogously
- $w_1(x) =$ largest eigenvector of $\boldsymbol{G}'\boldsymbol{F}$
- $m_k(x) =$ largest eigenvector of $\boldsymbol{G}'_{k-1}\boldsymbol{F}_{k-1}$
- $w_k(x) =$ largest eigenvector of $\boldsymbol{F}'_{k-1}\boldsymbol{G}_{k-1}$

Functional weighted PLS

Computationally equivalent approach

- Discretize $s_t^*(x)$ on a dense grid of q equally spaced points.
- Denote discretized $s_t^*(x)$ as $T \times q$ matrix $\boldsymbol{G}^*$ and let $\boldsymbol{G} = \boldsymbol{W}\boldsymbol{G}^*$.
- Define $\boldsymbol{G}_{k-1}$ and $\boldsymbol{F}_{k-1}$ analogously
- $w_1(x) =$ largest eigenvector of $\boldsymbol{G}'\boldsymbol{F}$
- $m_k(x) =$ largest eigenvector of $\boldsymbol{G}_{k-1}'\boldsymbol{F}_{k-1}$
- $w_k(x) =$ largest eigenvector of $\boldsymbol{F}_{k-1}'\boldsymbol{G}_{k-1}$

Functional weighted PLS

Computationally equivalent approach

- Discretize $s_t^*(x)$ on a dense grid of q equally spaced points.
- Denote discretized $s_t^*(x)$ as $T \times q$ matrix $\boldsymbol{G}^*$ and let $\boldsymbol{G} = \boldsymbol{W}\boldsymbol{G}^*$.
- Define $\boldsymbol{G}_{k-1}$ and $\boldsymbol{F}_{k-1}$ analogously
- $w_1(x) =$ largest eigenvector of $\boldsymbol{G}'\boldsymbol{F}$
- $m_k(x) =$ largest eigenvector of $\boldsymbol{G}_{k-1}'\boldsymbol{F}_{k-1}$
- $w_k(x) =$ largest eigenvector of $\boldsymbol{F}_{k-1}'\boldsymbol{G}_{k-1}$

Functional weighted PLS

Computationally equivalent approach

- Discretize $s_t^*(x)$ on a dense grid of q equally spaced points.
- Denote discretized $s_t^*(x)$ as $T \times q$ matrix $\boldsymbol{G}^*$ and let $\boldsymbol{G} = \boldsymbol{W}\boldsymbol{G}^*$.
- Define $\boldsymbol{G}_{k-1}$ and $\boldsymbol{F}_{k-1}$ analogously
- $w_1(x) =$ largest eigenvector of $\boldsymbol{G}'\boldsymbol{F}$
- $m_k(x) =$ largest eigenvector of $\boldsymbol{G}_{k-1}'\boldsymbol{F}_{k-1}$
- $w_k(x) =$ largest eigenvector of $\boldsymbol{F}_{k-1}'\boldsymbol{G}_{k-1}$

Functional weighted PLS

Functional autoregression coefficient:

$$b(x,u) = \sum_{k=1}^{\infty} w_k(u)\phi_k(x)$$

By orthogonality of β_k:

$$\phi_k(x) = (\beta_k'\beta_k)^{-1}\beta_k'\hat{\boldsymbol{g}}^*(x).$$

Therefore

$$\hat{b}(x,u) = \sum_{k=1}^{K} w_k(u)(\beta_k'\beta_k)^{-1}\beta_k'\hat{\boldsymbol{g}}^*(x),$$

for some finite K.

Functional weighted PLS

Functional autoregression coefficient:

$$b(x,u) = \sum_{k=1}^{\infty} w_k(u)\phi_k(x)$$

By orthogonality of β_k:

$$\phi_k(x) = (\beta_k'\beta_k)^{-1}\beta_k'\hat{\boldsymbol{g}}^*(x).$$

Therefore

$$\hat{b}(x,u) = \sum_{k=1}^{K} w_k(u)(\beta_k'\beta_k)^{-1}\beta_k'\hat{\boldsymbol{g}}^*(x),$$

for some finite K.

Functional weighted PLS

Functional autoregression coefficient:

$$b(x,u) = \sum_{k=1}^{\infty} w_k(u)\phi_k(x)$$

By orthogonality of $\boldsymbol{\beta}_k$:

$$\phi_k(x) = (\boldsymbol{\beta}_k'\boldsymbol{\beta}_k)^{-1}\boldsymbol{\beta}_k'\hat{\boldsymbol{g}}^*(x).$$

Therefore

$$\hat{b}(x,u) = \sum_{k=1}^{K} w_k(u)(\boldsymbol{\beta}_k'\boldsymbol{\beta}_k)^{-1}\boldsymbol{\beta}_k'\hat{\boldsymbol{g}}^*(x),$$

for some finite K.

Functional weighted PLS

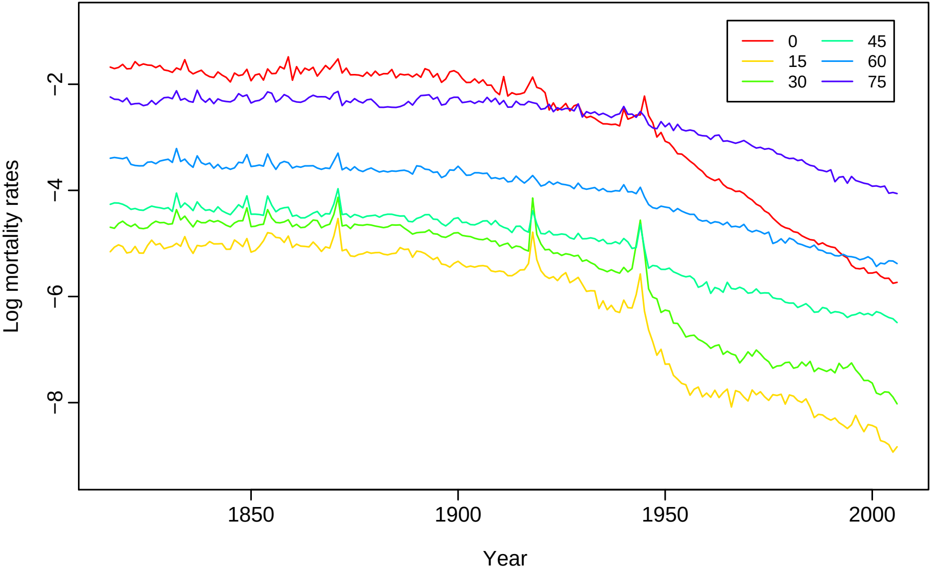
Forecasted curves:

$$\hat{f}_{t+1|t}(x) = \hat{\mu}(x) + \int [\hat{f}_t(u) - \hat{\mu}(u)]\, \hat{b}(x,u)\, du.$$

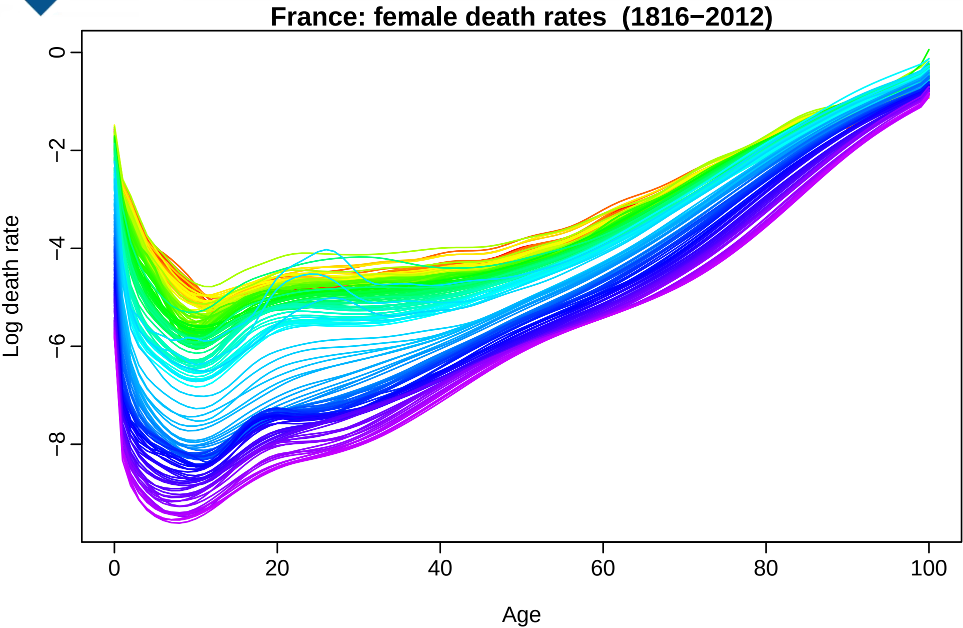
For $\hat{f}_{t+h|t}(x)$ where $h > 1$, apply iteratively.

Outline

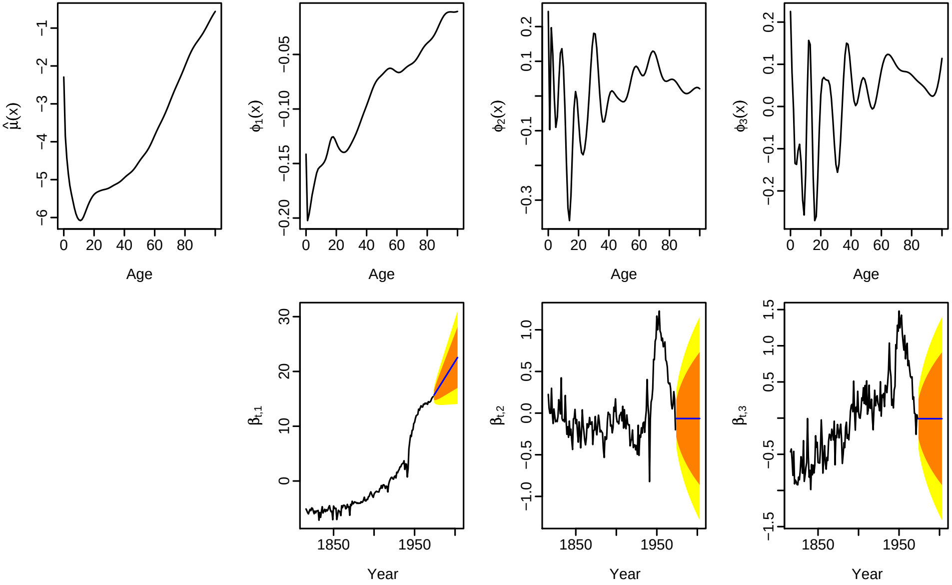
French female mortality rates



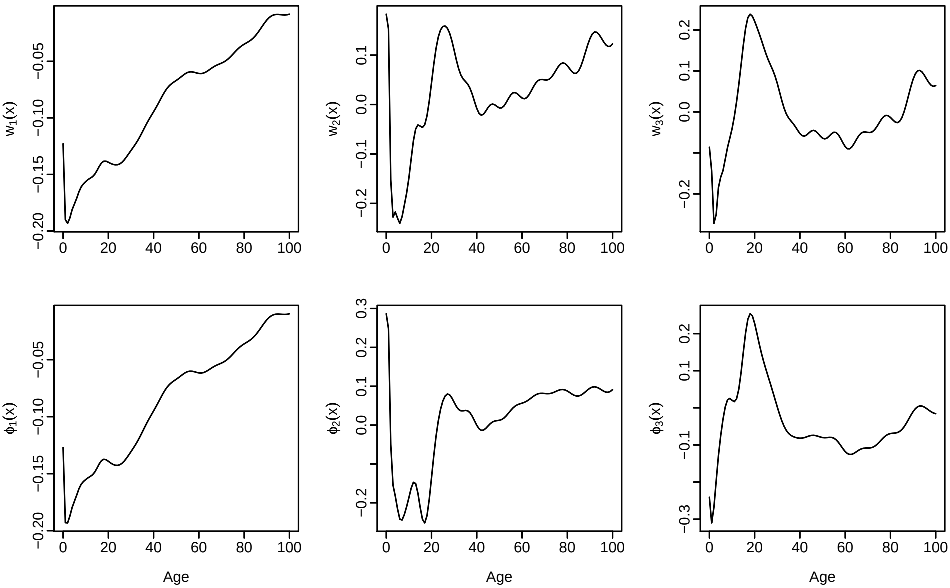
French female mortality rates



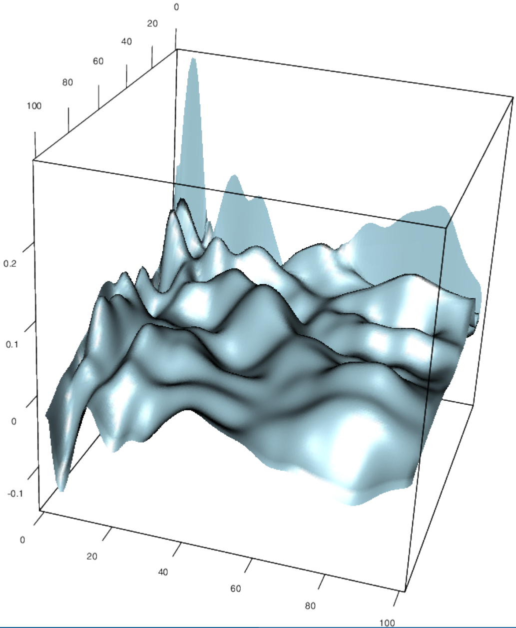
French mortality rate models



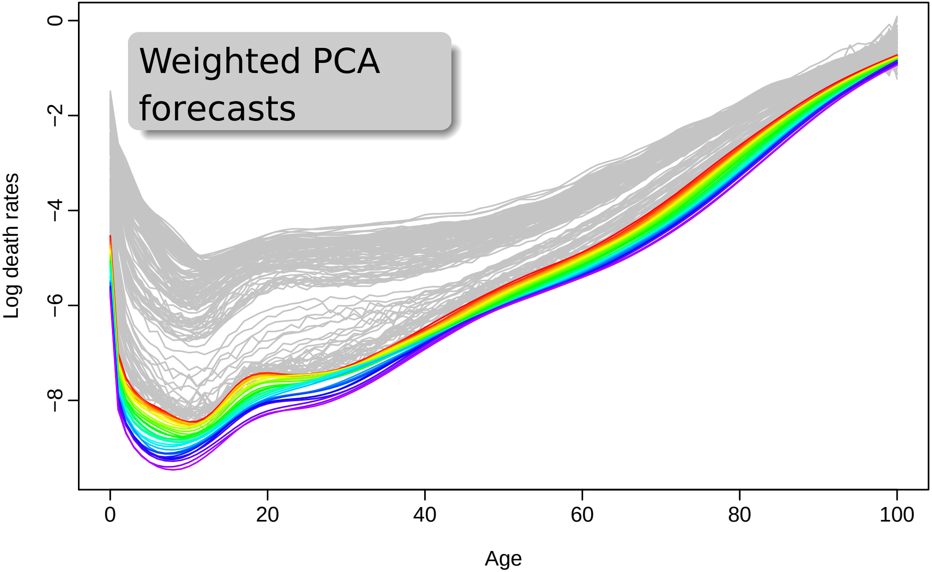
French mortality rate models



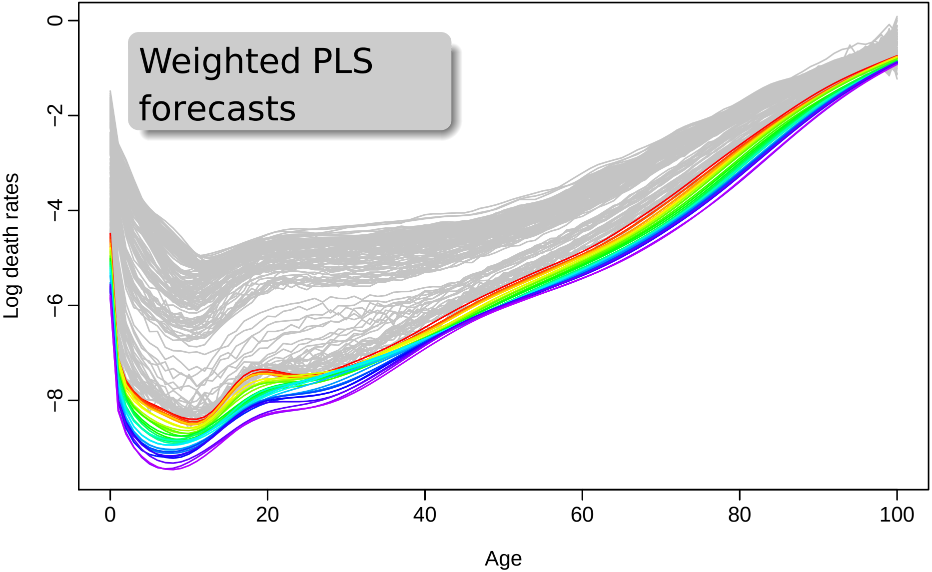
French mortality rate models



French mortality rate forecasts

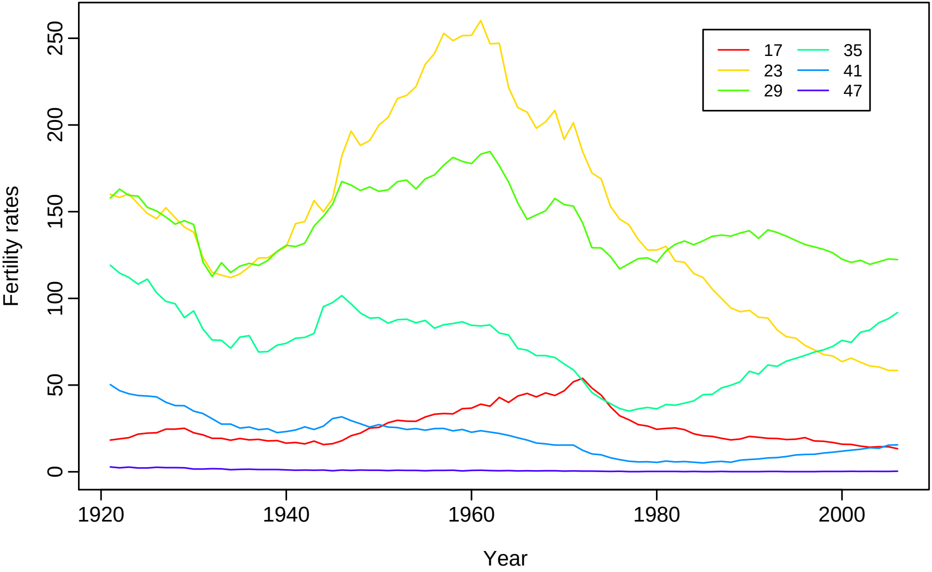


French mortality rate forecasts

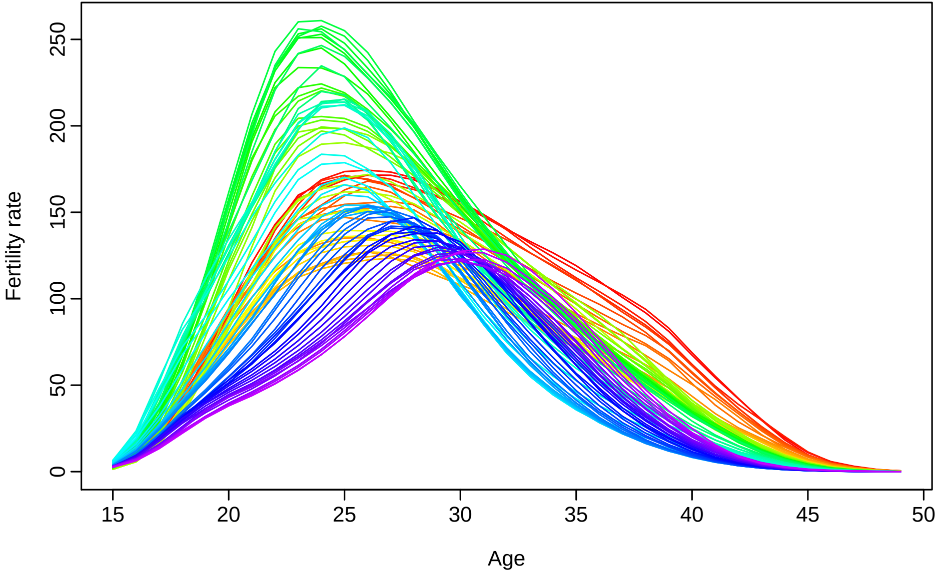


Outline

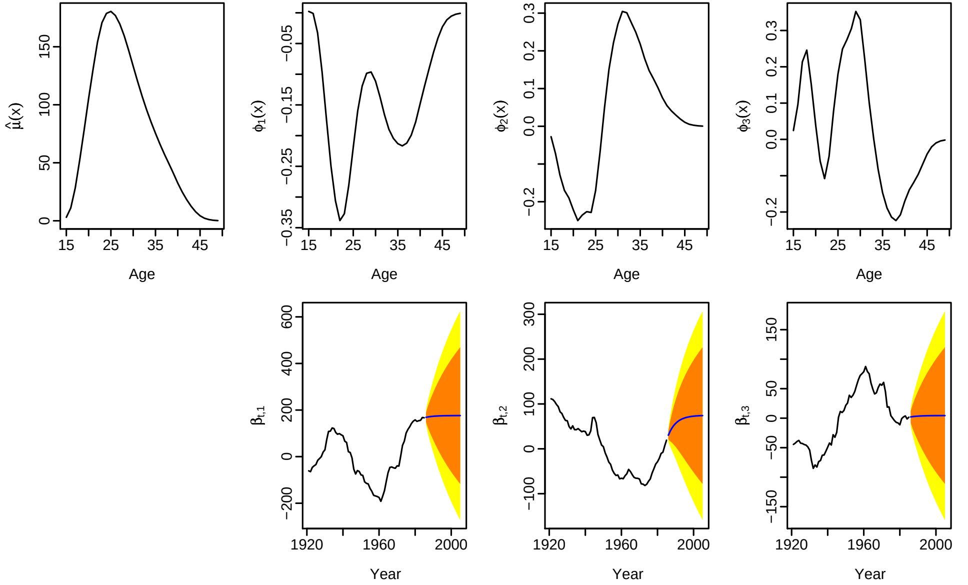
Australian fertility rates



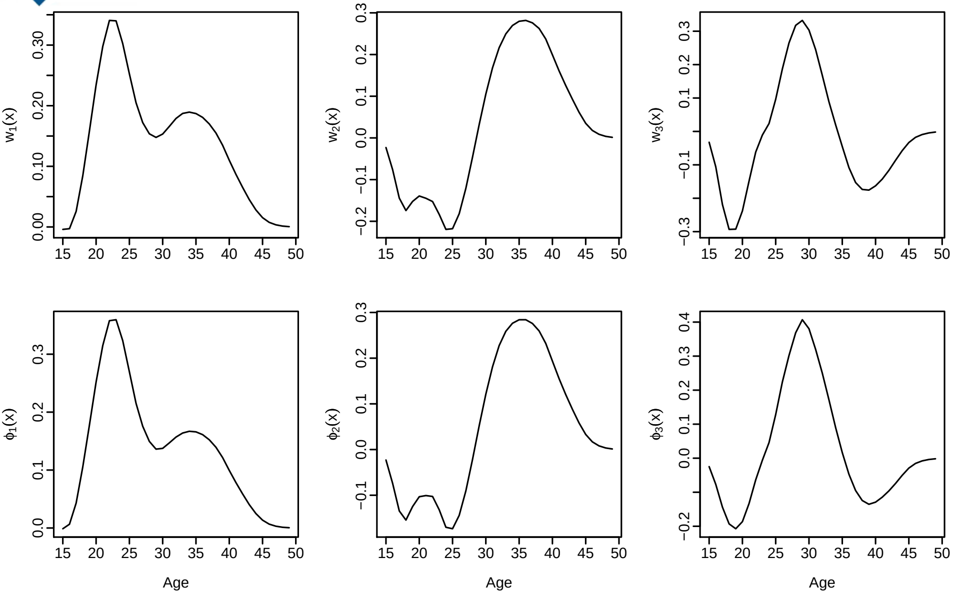
Australian fertility rates



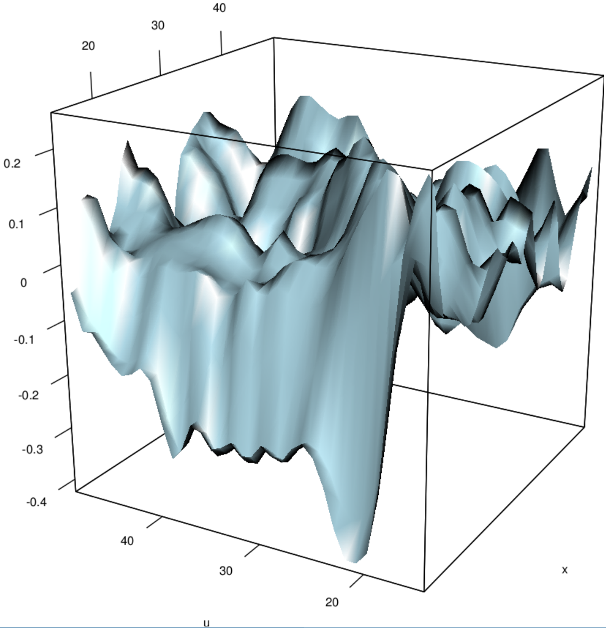
Australian fertility rate models



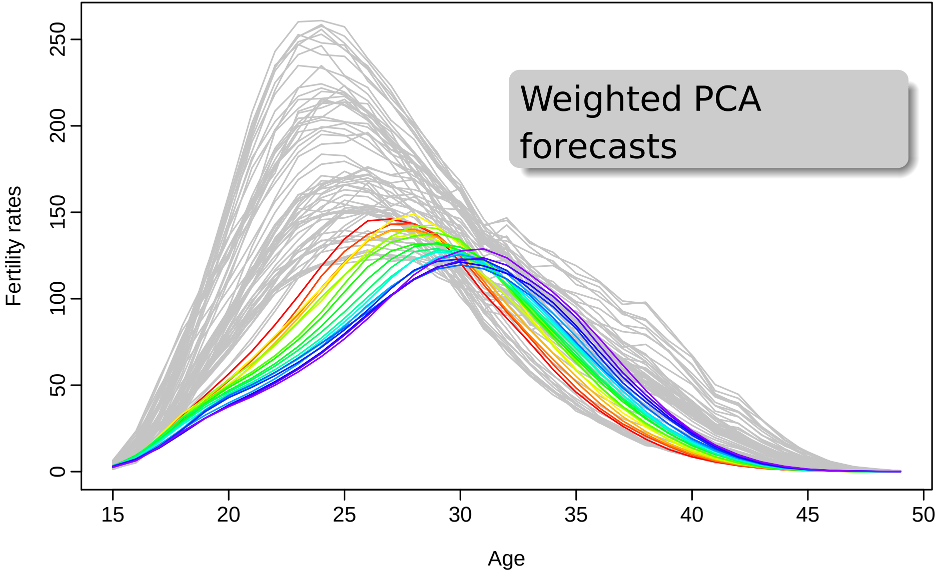
Australian fertility rate models



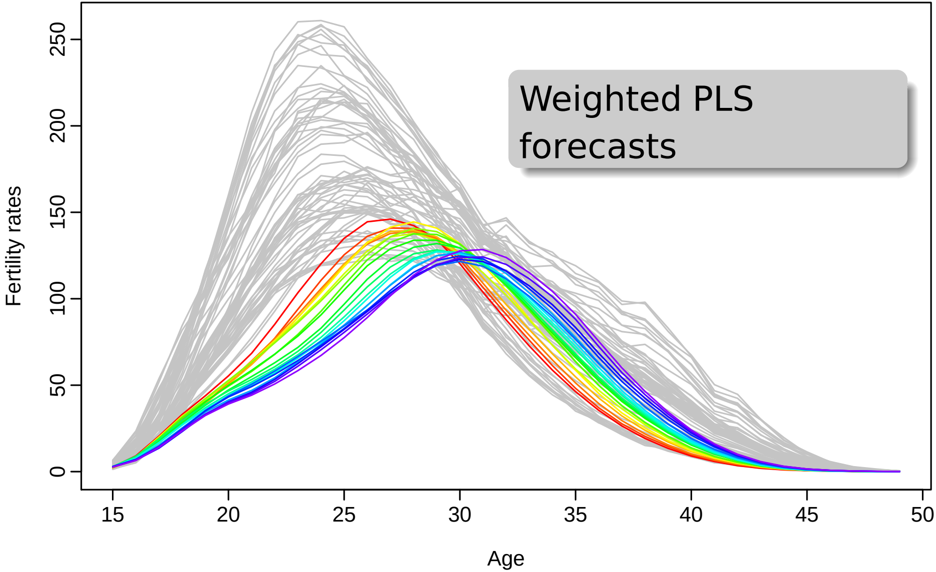
Australian fertility rate models



Australian fertility rate forecasts



Australian fertility rate forecasts



Outline

Forecast accuracy comparisons

$$\text{MSE}_t = \frac{1}{p}\sum_{i=1}^{p}\left[y_t(x_i) - \hat{y}_{t|t-1}(x_i)\right]^2.$$

- Averaged over last m years of observed data.
- For French female mortality, $m = 30$
- For Australian fertility, $m = 20$
- For comparison, compare random walk: $y_{t+1|t}(x) = y_t(x)$.

Forecast accuracy comparisons

$$\text{MSE}_t = \frac{1}{p}\sum_{i=1}^{p}\left[y_t(x_i) - \hat{y}_{t|t-1}(x_i)\right]^2.$$

- Averaged over last m years of observed data.
- For French female mortality, $m = 30$
- For Australian fertility, $m = 20$
- For comparison, compare random walk: $y_{t+1|t}(x) = y_t(x)$.

Forecast accuracy comparisons

$$\text{MSE}_t = \frac{1}{p}\sum_{i=1}^{p}\left[y_t(x_i) - \hat{y}_{t|t-1}(x_i)\right]^2.$$

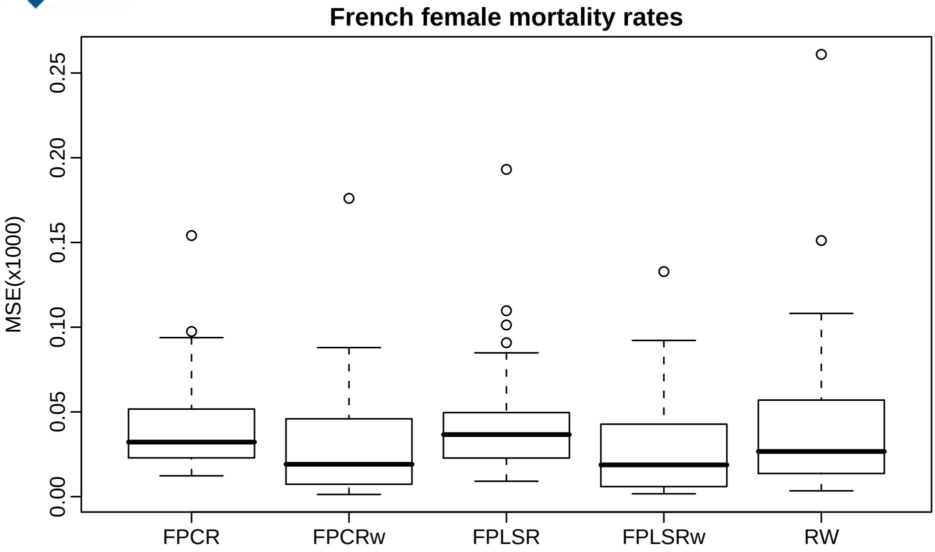
- Averaged over last m years of observed data.
- For French female mortality, $m = 30$
- For Australian fertility, $m = 20$
- For comparison, compare random walk: $y_{t+1|t}(x) = y_t(x)$.

Forecast accuracy comparisons

$$\mathrm{MSE}_t = \frac{1}{p}\sum_{i=1}^{p}\left[y_t(x_i) - \hat{y}_{t|t-1}(x_i)\right]^2 .$$

- Averaged over last m years of observed data.
- For French female mortality, $m = 30$
- For Australian fertility, $m = 20$
- For comparison, compare random walk: $y_{t+1|t}(x) = y_t(x)$.

Forecast accuracy comparisons

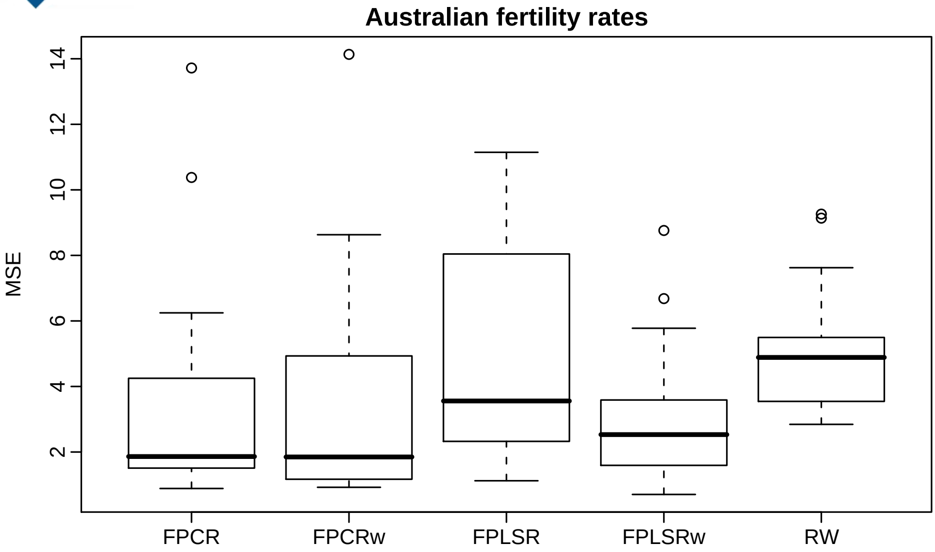


Forecast accuracy comparisons

MSE: French female mortality rates ($x1000$)

K	FPC	FPC$_w$	FPLSR	FPLSR$_w$	RW
1	0.5956	0.0293	0.5994	0.0607	
2	0.0537	0.0310	0.0738	0.0288	
3	0.0316	0.0310	0.0445	0.0288	
4	0.0296	0.0311	0.0428	0.0288	
5	0.0287	0.0311	0.0472	0.0297	
6	0.0425	0.0311	0.0474	0.0291	0.0437

Forecast accuracy comparisons



Forecast accuracy comparisons

MSE: Australian fertility rates

K	FPC	FPC$_w$	FPLSR	FPLSR$_w$	RW
1	99.0611	16.7304	94.0311	53.8186	
2	56.3095	3.3019	54.3410	17.5883	
3	24.9330	3.2580	26.0428	10.2599	
4	15.6845	3.1995	19.7227	4.4818	
5	4.4495	3.2132	5.9299	4.0573	
6	3.4310	3.2123	4.9205	2.9046	4.9800

Computation time

- Weighted FPLSR more efficient than weighted FPC as FPC requires many univariate time series models.
- **Time to fit 100 replications:**

Method	Fertility data	Mortality data
FPC	34.1072	62.2797
FPC_w	33.1424	60.8426
FPLSR	0.4287	2.9184
$FPLSR_w$	0.4537	3.1602
RW	0.0000	0.0002

(Intel Xeon 2.33GHz processor)

Computation time

- Weighted FPLSR more efficient than weighted FPC as FPC requires many univariate time series models.
- **Time to fit 100 replications:**

Method	Fertility data	Mortality data
FPC	34.1072	62.2797
FPC_w	33.1424	60.8426
FPLSR	0.4287	2.9184
$FPLSR_w$	0.4537	3.1602
RW	0.0000	0.0002

(Intel Xeon 2.33GHz processor)

Outline

Sources of uncertainty

Functional PCA

1. smoothing error in estimating $s_t(x)$
2. error in estimating $\mu(x)$
3. error in forecasting the scores $\beta_{t,k}$
4. error in the model residuals $e_t(x)$
5. appropriateness of model

Sources of uncertainty

Functional PCA

1. smoothing error in estimating $s_t(x)$
2. error in estimating $\mu(x)$
3. error in forecasting the scores $\beta_{t,k}$
4. error in the model residuals $e_t(x)$
5. appropriateness of model

Sources of uncertainty

Functional PCA

1. smoothing error in estimating $s_t(x)$
2. error in estimating $\mu(x)$
3. error in forecasting the scores $\beta_{t,k}$
4. error in the model residuals $e_t(x)$
5. appropriateness of model

Sources of uncertainty

Functional PCA

1. smoothing error in estimating $s_t(x)$
2. error in estimating $\mu(x)$
3. error in forecasting the scores $\beta_{t,k}$
4. error in the model residuals $e_t(x)$
5. appropriateness of model

Sources of uncertainty

Functional PCA

1. smoothing error in estimating $s_t(x)$
2. error in estimating $\mu(x)$
3. error in forecasting the scores $\beta_{t,k}$
4. error in the model residuals $e_t(x)$
5. appropriateness of model

Sources of uncertainty

Functional PLS

1. smoothing error in estimating $s_t(x)$
2. error in estimating $\mu(x)$
3. error in estimating $b(x, u)$
4. error in the model residuals $e_t(x)$
5. appropriateness of model

Bootstrap curves for FPC

- Let $\hat{\xi}_{t,k} = \hat{\beta}_{t,k} - \hat{\beta}_{t|t-1,k}$ be 1-step errors of PC scores.
- $\{\xi_k^{(\ell)}\}$ sampled with replacement from $\{\hat{\xi}_{t,k}\}$.
- Simulate future sample paths of $\beta_{T+h|T,k}$ using these bootstrapped residuals: $\{\beta_{T+h|T,k}^{(\ell)}\}$.
- $\{e^{(\ell)}(x)\}$ sampled with replacement from residual functions $\{\hat{e}_1(x), \dots, \hat{e}_T(x)\}$.
- $\{\varepsilon^{(\ell)}\}$ sampled with replacement from $\{\hat{\varepsilon}_{1,i}, \dots, \hat{\varepsilon}_{T,i}\}$.

- Prediction intervals are produced from the bootstrap variants using percentiles.

Bootstrap curves for FPC

- Let $\hat{\xi}_{t,k} = \hat{\beta}_{t,k} - \hat{\beta}_{t|t-1,k}$ be 1-step errors of PC scores.
- $\{\xi_k^{(\ell)}\}$ sampled with replacement from $\{\hat{\xi}_{t,k}\}$.
- Simulate future sample paths of $\beta_{T+h|T,k}$ using these bootstrapped residuals: $\{\beta_{T+h|T,k}^{(\ell)}\}$.
- $\{e^{(\ell)}(x)\}$ sampled with replacement from residual functions $\{\hat{e}_1(x), \dots, \hat{e}_T(x)\}$.
- $\{\varepsilon^{(\ell)}\}$ sampled with replacement from $\{\hat{\varepsilon}_{1,i}, \dots, \hat{\varepsilon}_{T,i}\}$.

- Prediction intervals are produced from the bootstrap variants using percentiles.

Bootstrap curves for FPC

- Let $\hat{\xi}_{t,k} = \hat{\beta}_{t,k} - \hat{\beta}_{t|t-1,k}$ be 1-step errors of PC scores.
- $\{\xi_k^{(\ell)}\}$ sampled with replacement from $\{\hat{\xi}_{t,k}\}$.
- Simulate future sample paths of $\beta_{T+h|T,k}$ using these bootstrapped residuals: $\{\beta_{T+h|T,k}^{(\ell)}\}$.
- $\{e^{(\ell)}(x)\}$ sampled with replacement from residual functions $\{\hat{e}_1(x), \dots, \hat{e}_T(x)\}$.
- $\{\varepsilon^{(\ell)}\}$ sampled with replacement from $\{\hat{\varepsilon}_{1,i}, \dots, \hat{\varepsilon}_{T,i}\}$.
- Prediction intervals are produced from the bootstrap variants using percentiles.

Bootstrap curves for FPC

- Let $\hat{\xi}_{t,k} = \hat{\beta}_{t,k} - \hat{\beta}_{t|t-1,k}$ be 1-step errors of PC scores.
- $\{\xi_k^{(\ell)}\}$ sampled with replacement from $\{\hat{\xi}_{t,k}\}$.
- Simulate future sample paths of $\beta_{T+h|T,k}$ using these bootstrapped residuals: $\{\beta_{T+h|T,k}^{(\ell)}\}$.
- $\{e^{(\ell)}(x)\}$ sampled with replacement from residual functions $\{\hat{e}_1(x), \dots, \hat{e}_T(x)\}$.
- $\{\varepsilon^{(\ell)}\}$ sampled with replacement from $\{\hat{\varepsilon}_{1,i}, \dots, \hat{\varepsilon}_{T,i}\}$.

$$y_{T+h|T}^{(\ell)}(x_i) = \hat{\mu}(x_i) + \sum_{k=1}^{K} \beta_{T+h|T,k}^{(\ell)} \phi_k(x_i) + e^{(\ell)}(x_i) + \hat{\sigma}_{T+h}(x_i)\varepsilon^{(\ell)}$$

- Prediction intervals are produced from the bootstrap variants using percentiles.

Bootstrap curves for FPC

- Let $\hat{\xi}_{t,k} = \hat{\beta}_{t,k} - \hat{\beta}_{t|t-1,k}$ be 1-step errors of PC scores.
- $\{\xi_k^{(\ell)}\}$ sampled with replacement from $\{\hat{\xi}_{t,k}\}$.
- Simulate future sample paths of $\beta_{T+h|T,k}$ using these bootstrapped residuals: $\{\beta_{T+h|T,k}^{(\ell)}\}$.
- $\{e^{(\ell)}(x)\}$ sampled with replacement from residual functions $\{\hat{e}_1(x), \dots, \hat{e}_T(x)\}$.
- $\{\varepsilon^{(\ell)}\}$ sampled with replacement from $\{\hat{\varepsilon}_{1,i}, \dots, \hat{\varepsilon}_{T,i}\}$.

$$y_{T+h|T}^{(\ell)}(x_i) = \hat{\mu}(x_i) + \sum_{k=1}^{K} \beta_{T+h|T,k}^{(\ell)} \phi_k(x_i) + e^{(\ell)}(x_i) + \hat{\sigma}_{T+h}(x_i)\varepsilon^{(\ell)}$$

- Prediction intervals are produced from the bootstrap variants using percentiles.

Bootstrap curves for FPC

- Let $\hat{\xi}_{t,k} = \hat{\beta}_{t,k} - \hat{\beta}_{t|t-1,k}$ be 1-step errors of PC scores.
- $\{\xi_k^{(\ell)}\}$ sampled with replacement from $\{\hat{\xi}_{t,k}\}$.
- Simulate future sample paths of $\beta_{T+h|T,k}$ using these bootstrapped residuals: $\{\beta_{T+h|T,k}^{(\ell)}\}$.
- $\{e^{(\ell)}(x)\}$ sampled with replacement from residual functions $\{\hat{e}_1(x), \dots, \hat{e}_T(x)\}$.
- $\{\varepsilon^{(\ell)}\}$ sampled with replacement from $\{\hat{\varepsilon}_{1,i}, \dots, \hat{\varepsilon}_{T,i}\}$.

$$y_{T+h|T}^{(\ell)}(x_i) = \hat{\mu}(x_i) + \sum_{k=1}^{K} \beta_{T+h|T,k}^{(\ell)} \phi_k(x_i) + e^{(\ell)}(x_i) + \hat{\sigma}_{T+h}(x_i)\varepsilon^{(\ell)}$$

- Prediction intervals are produced from the bootstrap variants using percentiles.

Bootstrap curves for FPC

- Let $\hat{\xi}_{t,k} = \hat{\beta}_{t,k} - \hat{\beta}_{t|t-1,k}$ be 1-step errors of PC scores.
- $\{\xi_k^{(\ell)}\}$ sampled with replacement from $\{\hat{\xi}_{t,k}\}$.
- Simulate future sample paths of $\beta_{T+h|T,k}$ using these bootstrapped residuals: $\{\beta_{T+h|T,k}^{(\ell)}\}$.
- $\{e^{(\ell)}(x)\}$ sampled with replacement from residual functions $\{\hat{e}_1(x), \dots, \hat{e}_T(x)\}$.
- $\{\varepsilon^{(\ell)}\}$ sampled with replacement from $\{\hat{\varepsilon}_{1,i}, \dots, \hat{\varepsilon}_{T,i}\}$.

$$y_{T+h|T}^{(\ell)}(x_i) = \hat{\mu}(x_i) + \sum_{k=1}^{K} \beta_{T+h|T,k}^{(\ell)} \phi_k(x_i) + e^{(\ell)}(x_i) + \hat{\sigma}_{T+h}(x_i)\varepsilon^{(\ell)}$$

- Prediction intervals are produced from the bootstrap variants using percentiles.

Bootstrap curves for FPLS

- Resample residuals and construct bootstrap samples:

$$\boldsymbol{f}^{(\ell)}(x) = \hat{\mu}(x) + \hat{\boldsymbol{f}}^{*}(x) + \boldsymbol{e}^{(\ell)}(x),$$
$$\boldsymbol{g}^{(\ell)}(x) = \hat{\mu}(x) + \hat{\boldsymbol{g}}^{*}(x) + \boldsymbol{e}^{(\ell)}(x).$$

- Construct weighted FPLSR model for each bootstrap sample.
- Compute bootstrapped forecast variants:

$$y^{(\ell)}_{T+1|T}(x_i) = \hat{\mu}(x_i) + \int (f_T(u) - \hat{\mu}(u))\hat{b}^{(\ell)}(x_i, u)du + \hat{\sigma}_{T+1}(x_i)\hat{\varepsilon}^{(\ell)}$$

Bootstrap curves for FPLS

- Resample residuals and construct bootstrap samples:

$$\boldsymbol{f}^{(\ell)}(x) = \hat{\mu}(x) + \hat{\boldsymbol{f}}^{*}(x) + \boldsymbol{e}^{(\ell)}(x),$$
$$\boldsymbol{g}^{(\ell)}(x) = \hat{\mu}(x) + \hat{\boldsymbol{g}}^{*}(x) + \boldsymbol{e}^{(\ell)}(x).$$

- Construct weighted FPLSR model for each bootstrap sample.
- Compute bootstrapped forecast variants:

$$y^{(\ell)}_{T+1|T}(x_i) = \hat{\mu}(x_i) + \int (f_T(u) - \hat{\mu}(u))\hat{b}^{(\ell)}(x_i, u)du + \hat{\sigma}_{T+1}(x_i)\hat{\varepsilon}^{(\ell)}$$

Bootstrap curves for FPLS

- Resample residuals and construct bootstrap samples:
$$\boldsymbol{f}^{(\ell)}(x) = \hat{\mu}(x) + \hat{\boldsymbol{f}}^{*}(x) + \boldsymbol{e}^{(\ell)}(x),$$
$$\boldsymbol{g}^{(\ell)}(x) = \hat{\mu}(x) + \hat{\boldsymbol{g}}^{*}(x) + \boldsymbol{e}^{(\ell)}(x).$$
- Construct weighted FPLSR model for each bootstrap sample.
- Compute bootstrapped forecast variants:

$$y^{(\ell)}_{T+1|T}(x_i) = \hat{\mu}(x_i) + \int (f_T(u) - \hat{\mu}(u))\hat{b}^{(\ell)}(x_i, u)du + \hat{\sigma}_{T+1}(x_i)\hat{\varepsilon}^{(\ell)}$$

- Prediction intervals are produced from the bootstrap variants using percentiles.

Bootstrap curves for FPLS

- Resample residuals and construct bootstrap samples:

$$\boldsymbol{f}^{(\ell)}(x) = \hat{\mu}(x) + \hat{\boldsymbol{f}}^{*}(x) + \boldsymbol{e}^{(\ell)}(x),$$

$$\boldsymbol{g}^{(\ell)}(x) = \hat{\mu}(x) + \hat{\boldsymbol{g}}^{*}(x) + \boldsymbol{e}^{(\ell)}(x).$$

- Construct weighted FPLSR model for each bootstrap sample.
- Compute bootstrapped forecast variants:

$$y^{(\ell)}_{T+1|T}(x_i) = \hat{\mu}(x_i) + \int (f_T(u) - \hat{\mu}(u))\hat{b}^{(\ell)}(x_i, u)du + \hat{\sigma}_{T+1}(x_i)\hat{\varepsilon}^{(\ell)}$$

- Prediction intervals are produced from the bootstrap variants using percentiles.

Bootstrap curves for FPLS

- Resample residuals and construct bootstrap samples:

$$\boldsymbol{f}^{(\ell)}(x) = \hat{\mu}(x) + \hat{\boldsymbol{f}}^{*}(x) + \boldsymbol{e}^{(\ell)}(x),$$

$$\boldsymbol{g}^{(\ell)}(x) = \hat{\mu}(x) + \hat{\boldsymbol{g}}^{*}(x) + \boldsymbol{e}^{(\ell)}(x).$$

- Construct weighted FPLSR model for each bootstrap sample.
- Compute bootstrapped forecast variants:

$$y^{(\ell)}_{T+1|T}(x_i) = \hat{\mu}(x_i) + \int (f_T(u) - \hat{\mu}(u))\hat{b}^{(\ell)}(x_i, u)du + \hat{\sigma}_{T+1}(x_i)\hat{\varepsilon}^{(\ell)}$$

- Prediction intervals are produced from the bootstrap variants using percentiles.

Bootstrap curves for FPLS

- Resample residuals and construct bootstrap samples:

$$\boldsymbol{f}^{(\ell)}(x) = \hat{\mu}(x) + \hat{\boldsymbol{f}}^{*}(x) + \boldsymbol{e}^{(\ell)}(x),$$

$$\boldsymbol{g}^{(\ell)}(x) = \hat{\mu}(x) + \hat{\boldsymbol{g}}^{*}(x) + \boldsymbol{e}^{(\ell)}(x).$$

- Construct weighted FPLSR model for each bootstrap sample.
- Compute bootstrapped forecast variants:

$$y_{T+1|T}^{(\ell)}(x_i) = \hat{\mu}(x_i) + \int (f_T(u) - \hat{\mu}(u))\hat{b}^{(\ell)}(x_i, u)du + \hat{\sigma}_{T+1}(x_i)\hat{\varepsilon}^{(\ell)}$$

- Prediction intervals are produced from the bootstrap variants using percentiles.

Coverage probability

Empirical coverage probability of 95% intervals

$$\frac{1}{mph}\sum_{t=T-m+1}^{T}\sum_{j=1}^{h}\sum_{i=1}^{p}1\left(\hat{y}_{t+j|t}^{(0.025)}(x_i)<y_{t+j}(x_i)<\hat{y}_{t+j|t}^{(0.975)}(x_i)\right)$$

- $\hat{y}_{t+j|t}^{(\alpha)}(x_i)=\alpha$-quantile from bootstrap samples
- $m=$ smallest number of observations used to fit a model

Method	Fertility data	Mortality data
FPC	98.00%	97.19%
FPLS	96.86%	97.23%

Coverage probability

Empirical coverage probability of 95% intervals

$$\frac{1}{mph}\sum_{t=T-m+1}^{T}\sum_{j=1}^{h}\sum_{i=1}^{p} 1\left(\hat{y}_{t+j|t}^{(0.025)}(x_i) < y_{t+j}(x_i) < \hat{y}_{t+j|t}^{(0.975)}(x_i)\right)$$

- $\hat{y}_{t+j|t}^{(\alpha)}(x_i) = \alpha$-quantile from bootstrap samples
- $m =$ smallest number of observations used to fit a model

Method	Fertility data	Mortality data
FPC	98.00%	97.19%
FPLS	96.86%	97.23%

Coverage probability

Empirical coverage probability of 95% intervals

$$\frac{1}{mph}\sum_{t=T-m+1}^{T}\sum_{j=1}^{h}\sum_{i=1}^{p}1\left(\hat{y}_{t+j|t}^{(0.025)}(x_i) < y_{t+j}(x_i) < \hat{y}_{t+j|t}^{(0.975)}(x_i)\right)$$

- $\hat{y}_{t+j|t}^{(\alpha)}(x_i) = \alpha$-quantile from bootstrap samples
- $m =$ smallest number of observations used to fit a model

Method	Fertility data	Mortality data
FPC	98.00%	97.19%
FPLS	96.86%	97.23%

Coverage probability

Empirical coverage probability of 95% intervals

$$\frac{1}{mph}\sum_{t=T-m+1}^{T}\sum_{j=1}^{h}\sum_{i=1}^{p} 1\left(\hat{y}_{t+j|t}^{(0.025)}(x_i) < y_{t+j}(x_i) < \hat{y}_{t+j|t}^{(0.975)}(x_i)\right)$$

- $\hat{y}_{t+j|t}^{(\alpha)}(x_i) = \alpha$-quantile from bootstrap samples
- $m =$ smallest number of observations used to fit a model

Method	Fertility data	Mortality data
FPC	98.00%	97.19%
FPLS	96.86%	97.23%

Adjusted prediction intervals

- Compute $d(x) =$ difference between $(1-\alpha/2)$ and $(\alpha/2)$ quantiles of $\{\hat{f}_{t+1}(x) - \hat{f}_{t+1|t}(x);\ t = m, \dots, T-1\}$
- Let $[\ell_h(x), u_h(x)]$ be $100(1-\alpha)\%$ h-step-ahead prediction intervals obtained from bootstrap methods.
- Ideally, $u_1(x) - \ell_1(x) = d(x)$.

Adjusted prediction interval

$$\Big[0.5\{\ell_h(x)+u_h(x)\} - \{u_h(x)-\ell_h(x)\}p(x),\quad 0.5\{\ell_h(x)+u_h(x)\} + \{u_h(x)-\ell_h(x)\}p(x)\Big]$$

where $p(x) = 0.5d(x)/[u_1(x) - \ell_1(x)]$.

Adjusted prediction intervals

- Compute $d(x)$ = difference between $(1-\alpha/2)$ and $(\alpha/2)$ quantiles of $\{\hat{f}_{t+1}(x) - \hat{f}_{t+1|t}(x);\ t = m, \dots, T-1\}$
- Let $[\ell_h(x), u_h(x)]$ be $100(1-\alpha)\%$ h-step-ahead prediction intervals obtained from bootstrap methods.
- Ideally, $u_1(x) - \ell_1(x) = d(x)$.

Adjusted prediction interval

$$\Big[0.5\{\ell_h(x) + u_h(x)\} - \{u_h(x) - \ell_h(x)\}p(x),\\ 0.5\{\ell_h(x) + u_h(x)\} + \{u_h(x) - \ell_h(x)\}p(x)\Big]$$

where $p(x) = 0.5d(x)/[u_1(x) - \ell_1(x)]$.

Adjusted prediction intervals

- Compute $d(x)$ = difference between $(1-\alpha/2)$ and $(\alpha/2)$ quantiles of $\{\hat{f}_{t+1}(x) - \hat{f}_{t+1|t}(x);\ t = m, \dots, T-1\}$
- Let $[\ell_h(x), u_h(x)]$ be $100(1-\alpha)\%$ h-step-ahead prediction intervals obtained from bootstrap methods.
- Ideally, $u_1(x) - \ell_1(x) = d(x)$.

Adjusted prediction interval

$$\Big[0.5\{\ell_h(x) + u_h(x)\} - \{u_h(x) - \ell_h(x)\}p(x),\quad 0.5\{\ell_h(x) + u_h(x)\} + \{u_h(x) - \ell_h(x)\}p(x)\Big]$$

where $p(x) = 0.5d(x)/[u_1(x) - \ell_1(x)]$.

- For $h = 1$, these have same coverage as in-sample intervals if distributions symmetric.

Adjusted prediction intervals

- Compute $d(x)$ = difference between $(1-\alpha/2)$ and $(\alpha/2)$ quantiles of $\{\hat{f}_{t+1}(x) - \hat{f}_{t+1|t}(x);\ t = m,\dots,T-1\}$
- Let $[\ell_h(x), u_h(x)]$ be $100(1-\alpha)\%$ h-step-ahead prediction intervals obtained from bootstrap methods.
- Ideally, $u_1(x) - \ell_1(x) = d(x)$.

Adjusted prediction interval

$$\Big[0.5\{\ell_h(x)+u_h(x)\} - \{u_h(x)-\ell_h(x)\}p(x),\\ 0.5\{\ell_h(x)+u_h(x)\} + \{u_h(x)-\ell_h(x)\}p(x)\Big]$$

where $p(x) = 0.5d(x)/[u_1(x) - \ell_1(x)]$.

- For $h = 1$, these have same coverage as in-sample intervals if distributions symmetric.

Adjusted prediction intervals

- Compute $d(x)$ = difference between $(1-\alpha/2)$ and $(\alpha/2)$ quantiles of $\{\hat{f}_{t+1}(x) - \hat{f}_{t+1|t}(x);\ t = m, \dots, T-1\}$
- Let $[\ell_h(x), u_h(x)]$ be $100(1-\alpha)\%$ h-step-ahead prediction intervals obtained from bootstrap methods.
- Ideally, $u_1(x) - \ell_1(x) = d(x)$.

Adjusted prediction interval

$$\Big[0.5\{\ell_h(x) + u_h(x)\} - \{u_h(x) - \ell_h(x)\}p(x), \\ 0.5\{\ell_h(x) + u_h(x)\} + \{u_h(x) - \ell_h(x)\}p(x)\Big]$$

where $p(x) = 0.5d(x)/[u_1(x) - \ell_1(x)]$.

- For $h = 1$, these have same coverage as in-sample intervals if distributions symmetric.

Adjusted prediction intervals

- Compute $d(x)$ = difference between $(1-\alpha/2)$ and $(\alpha/2)$ quantiles of $\{\hat{f}_{t+1}(x) - \hat{f}_{t+1|t}(x);\ t = m, \dots, T-1\}$
- Let $[\ell_h(x), u_h(x)]$ be $100(1-\alpha)\%$ h-step-ahead prediction intervals obtained from bootstrap methods.
- Ideally, $u_1(x) - \ell_1(x) = d(x)$.

Adjusted prediction interval

$$\Big[0.5\{\ell_h(x) + u_h(x)\} - \{u_h(x) - \ell_h(x)\}p(x),\quad 0.5\{\ell_h(x) + u_h(x)\} + \{u_h(x) - \ell_h(x)\}p(x)\Big]$$

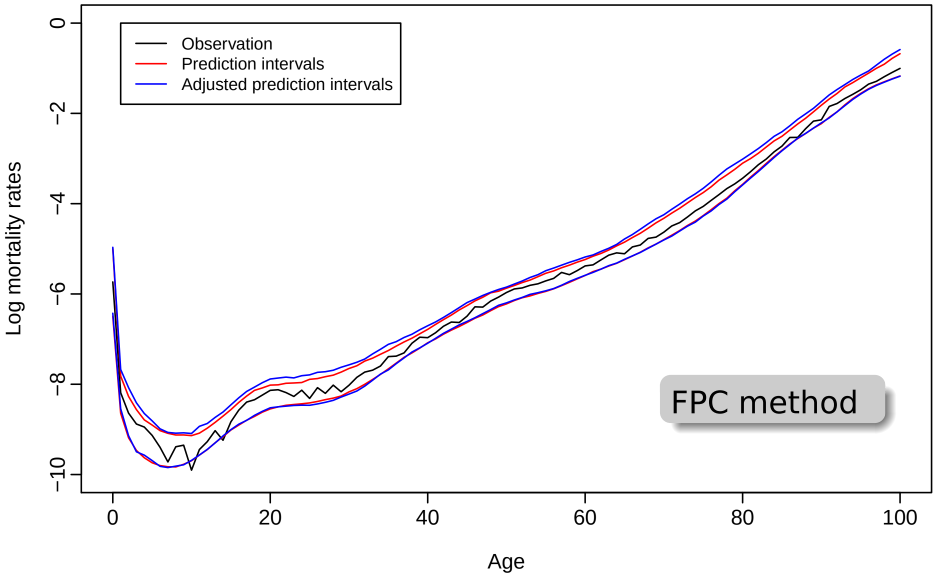
where $p(x) = 0.5d(x)/[u_1(x) - \ell_1(x)]$.

- For $h = 1$, these have same coverage as in-sample intervals if distributions symmetric.

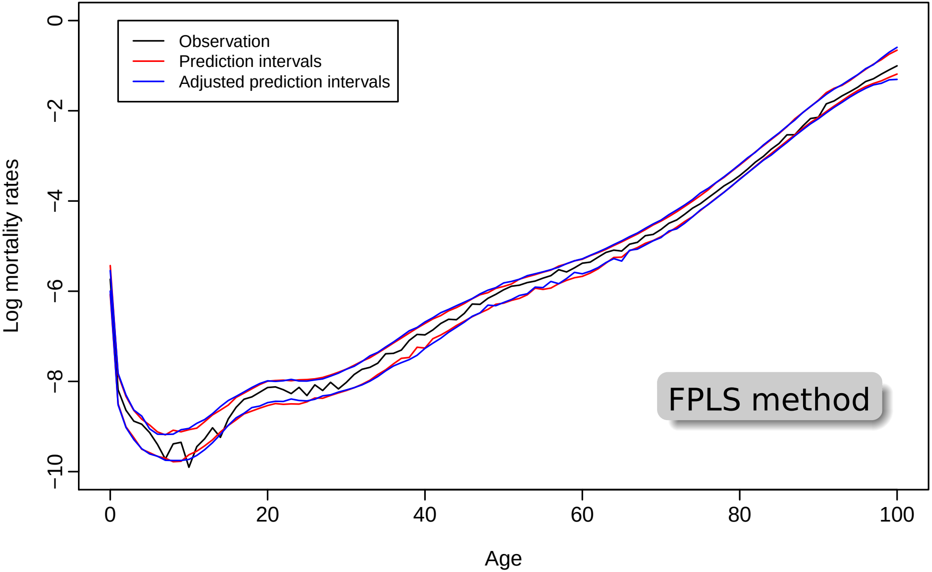
Adjusted prediction intervals

	Fertility data		Mortality data	
	95%	adj 95%	95%	adj 95%
FPC	98.00%	95.86%	97.19%	95.91%
FPLS	96.86%	94.89%	97.23%	94.95%

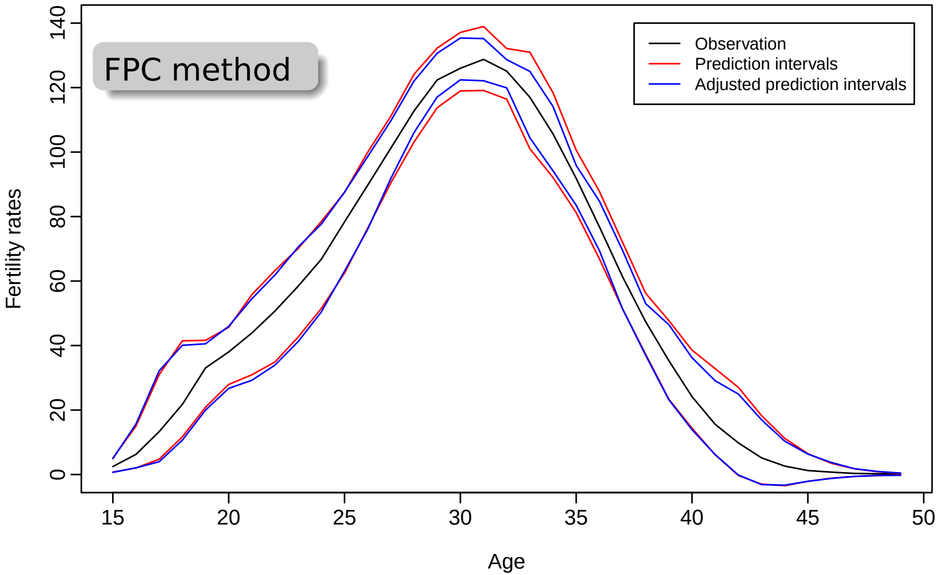
Adjusted prediction intervals



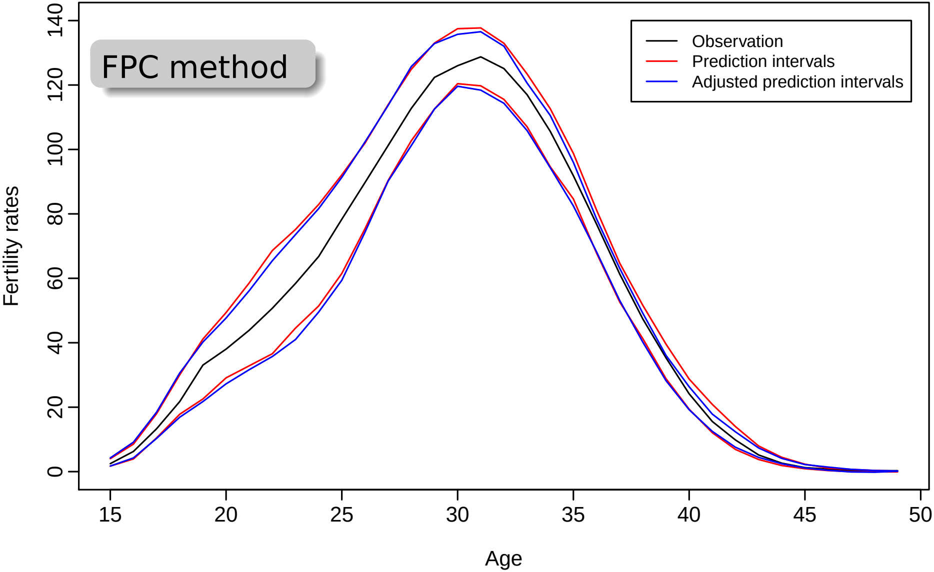
Adjusted prediction intervals



Adjusted prediction intervals



Adjusted prediction intervals



Outline

Comparisons

FPC advantages

1. allows more complex dynamics and higher order components;
2. eases interpretability of dynamic changes by separating out effects of a few orthogonal components;

FPLS advantages

1. latent components more suitable for prediction rather than variance decomposition.

Both methods implemented in `ftsa` package for R.

Comparisons

FPC advantages

1. allows more complex dynamics and higher order components;
2. eases interpretability of dynamic changes by separating out effects of a few orthogonal components;

FPLS advantages

1. latent components more suitable for prediction rather than variance decomposition.

Both methods implemented in ftsa package for R.

Comparisons

FPC advantages

1. allows more complex dynamics and higher order components;
2. eases interpretability of dynamic changes by separating out effects of a few orthogonal components;

FPLS advantages

1. latent components more suitable for prediction rather than variance decomposition.
2. Faster as there is no need to fit univariate time series models

Both methods implemented in `ftsa` package for R.

Comparisons

FPC advantages

1. allows more complex dynamics and higher order components;
2. eases interpretability of dynamic changes by separating out effects of a few orthogonal components;

FPLS advantages

1. latent components more suitable for prediction rather than variance decomposition.
2. Faster as there is no need to fit univariate time series models

Both methods implemented in `ftsa` package for R.

Comparisons

FPC advantages

1. allows more complex dynamics and higher order components;
2. eases interpretability of dynamic changes by separating out effects of a few orthogonal components;

FPLS advantages

1. latent components more suitable for prediction rather than variance decomposition.
2. Faster as there is no need to fit univariate time series models

Both methods implemented in `ftsa` package for R.

Comparisons

FPC advantages

1. allows more complex dynamics and higher order components;
2. eases interpretability of dynamic changes by separating out effects of a few orthogonal components;

FPLS advantages

1. latent components more suitable for prediction rather than variance decomposition.
2. Faster as there is no need to fit univariate time series models

Both methods implemented in `ftsa` package for R.

Comparisons

FPC advantages

1. allows more complex dynamics and higher order components;
2. eases interpretability of dynamic changes by separating out effects of a few orthogonal components;

FPLS advantages

1. latent components more suitable for prediction rather than variance decomposition.
2. Faster as there is no need to fit univariate time series models

Both methods implemented in `ftsa` package for R.

Outline

Selected references

Hyndman, Shang (2009). “Forecasting functional time series (with discussion)”. *Journal of the Korean Statistical Society* **38**(3), 199–221.

Hyndman (2014). *demography: Forecasting mortality, fertility, migration and population data*. `cran.r-project.org/package=demography`

Shang, Hyndman (2013). *ftsa: Functional time series analysis*. `cran.r-project.org/package=ftsa`