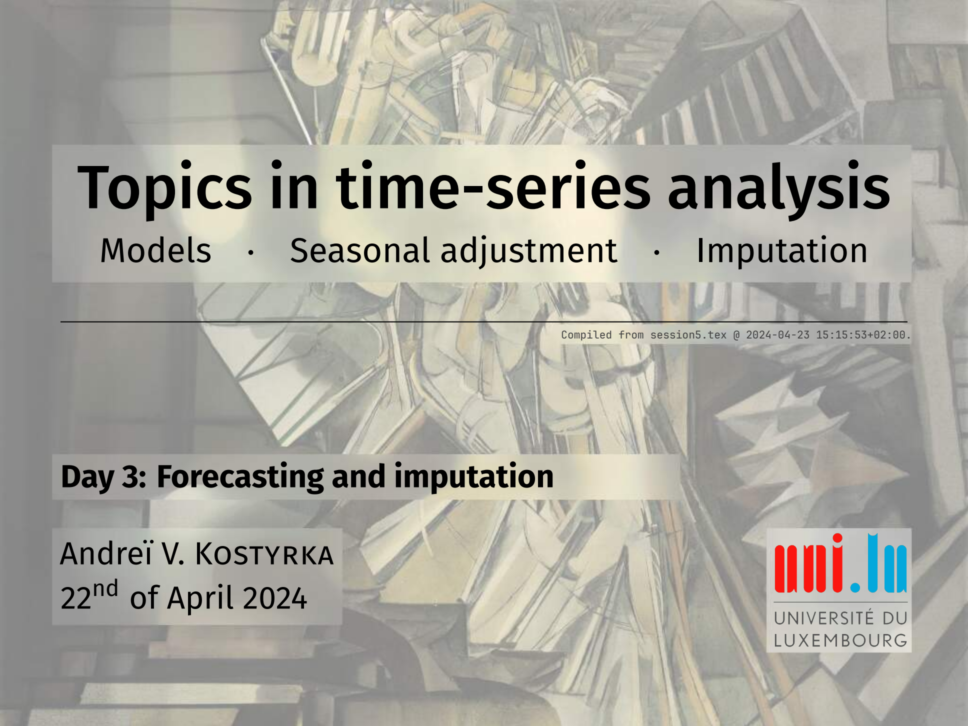


Topics in time-series analysis

Models · Seasonal adjustment · Imputation

`Compiled from session5.tex @ 2024-04-23 15:15:53+02:00.`

Day 3: Forecasting and imputation

Andreï V. Kostyrka
22nd of April 2024

UNIVERSITÉ DU LUXEMBOURG

Presentation structure

1. Forecasting time series

2. Multiple time series

3. Imputation

Forecasting time series

Recall forecasting in ARMA models

$$Y_t = \mu_0 + U_t + \sum_{i=1}^{p} \varphi_i Y_{t-i} + \varphi_p Y_{t-p} + \sum_{j=1}^{q} \theta_q U_{t-q}$$

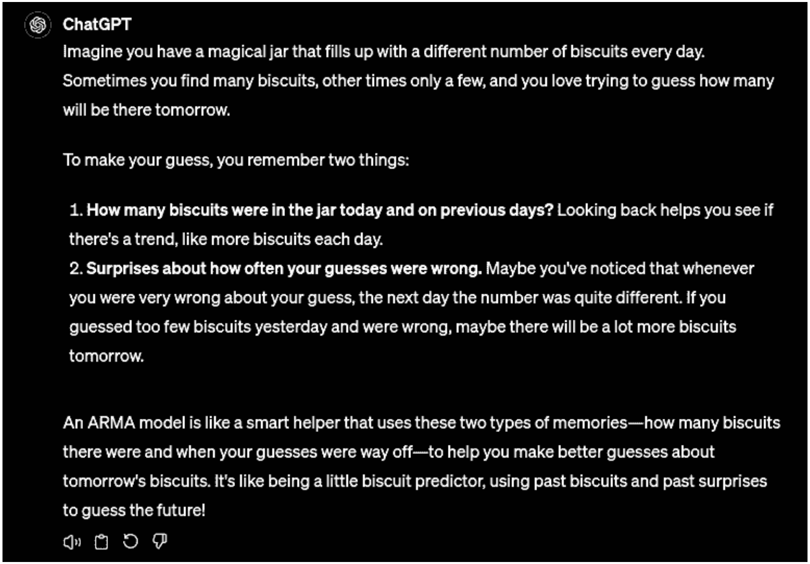
Let Ω_t denote all the information available up to t: $(\{Y_t\}_{t=1}^T$, guesses $\underline{Y} \coloneqq \{Y_0, \dots, Y_{-p+1}\}$, $\underline{U} \coloneqq \{U_0, \dots, U_{-q+1}\}$ and the conditional values of U_t denoted by $\hat{U}_t \coloneqq \hat{U}_t(\{Y_t\}_{t=1}^T, \underline{Y}, \underline{U})$.

Forecast: conditional expectation $\mathbb{E}(Y_{t+h} \mid \Omega_t)$.

$\mathbb{E}(Y_{t+h} \mid \Omega_t)$ is the BLP in linear specifications.

- $\mathbb{E}(U_t \mid Y_{t-1}, \dots, Y_{t-p}) = 0 \Rightarrow$ assume $U_{t+h} = 0$ for $h \geq 1$
- Unlike cross-sectional $\hat{Y}_i = X'\hat{\beta}$, in ARMA models, $\hat{U}_t \coloneqq \hat{U}_t(Y_t, \Omega_{t-1})$ is computed from the formula above!

ChatGPT: explain ARMA to a 7-year-old



ChatGPT

Imagine you have a magical jar that fills up with a different number of biscuits every day. Sometimes you find many biscuits, other times only a few, and you love trying to guess how many will be there tomorrow.

To make your guess, you remember two things:

1. **How many biscuits were in the jar today and on previous days?** Looking back helps you see if there's a trend, like more biscuits each day.
2. **Surprises about how often your guesses were wrong.** Maybe you've noticed that whenever you were very wrong about your guess, the next day the number was quite different. If you guessed too few biscuits yesterday and were wrong, maybe there will be a lot more biscuits tomorrow.

An ARMA model is like a smart helper that uses these two types of memories—how many biscuits there were and when your guesses were way off—to help you make better guesses about tomorrow's biscuits. It's like being a little biscuit predictor, using past biscuits and past surprises to guess the future!

One-step forecasting in ARMA models

$$\hat{Y}_{t+1} \coloneqq \hat{\mu} + \sum_{i=1}^{p} \hat{\varphi}_i Y_{t+1-i} + \sum_{j=1}^{q} \hat{\theta}_j \hat{U}_{t+1-j}$$

- Plug $\{(Y_t, \hat{U}_t)\}_{t=1}^{T}$ into the formula above
- At time t, assume $U_{t+1} = 0$
- At time $t + 1$, compute $\hat{U}_{t+1} \coloneqq Y_{t+1} - \hat{Y}_{t+1}$ for further use in the MA part once Y_{t+1} becomes known

Static and rolling forecasts

- Static forecasts receive no new information at times $t+1, t+2, \dots$ and converge to the long-run mean $\mu/(1-\sum_i \varphi_i)$
  - $\hat{Y}_{t+2} = f(Y_{t+1}, \dots)$ uses $\hat{Y}_{t+1}$ instead of Y_{t+1}
- Rolling forecasts use future values $Y_{t+1}, Y_{t+2}, \dots$ as soon as they are observed

Rolling forecasts are sometimes called **ARIMA filtering:** decompose the observed Y_{t+1} into contributions from its lags, past errors etc. The unexplained part (forecast error) becomes the model error.

- In **R**, `arima()` returns residuals $\{\hat{U}_t\}_{t=1}^{T}$

Multi-step forecasting and filtering

Since SARIMA models are embarrassingly short-term, it only makes sense to use extremely short horizons (1–2 points) for forecasting (+*c* for models with a strong seasonal component).

In practice, researchers often estimate a SARIMA model once and then, use it to update $\hat{Y}_{t+2}, \hat{Y}_{t+3}, \ldots$ as soon as new points become.

- If there is a calendar component, future values of the calendar regressor are also required (easy with JDemetra+ and its R interfaces – see Session 2)

Seasonality and forecasting

Does seasonality help forecasting?

- The air temperature in late July 2025 can be reasonably expected to be 30–35°
- However, climate scientists are interested in deviations from the baseline (average of last 5–10 measurements)

Possible approach: predict T_t with local polynomials (or even linear functions), S_t by extrapolating the MA filter of the seasonal component, and I_t with stationary ARMA models.

Multiple time series

Multivariate normal distribution (MVN)

Recall the univariate normal distribution with density $f_{\mathcal{N}}(t) \coloneqq \frac{1}{\sqrt{2\pi\sigma^2}} \exp \frac{-(t-\mu)^2}{2\sigma^2}$.

Generalise it to the **multi-variate Gaussian** density:

$$f_{\mathcal{N}}(t) \coloneqq (2\pi)^{-\dim t/2} (\det \Sigma)^{-1/2} \exp\left(-\frac{1}{2}(t-\mu)'\Sigma^{-1}(t-\mu)\right)$$

Reproducing property: any linear combination of the Gaussian vector is a Gaussian RV.

- Some textbooks even define the MVN in this manner
- The marginals of a MVN are also normal

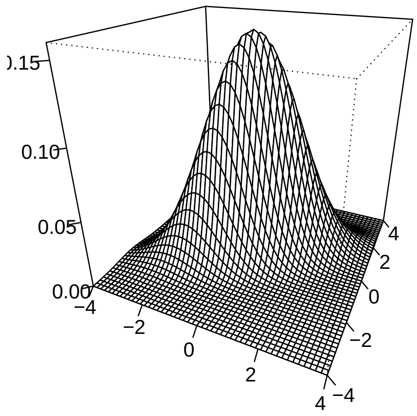
MM estimation of MVN distribution

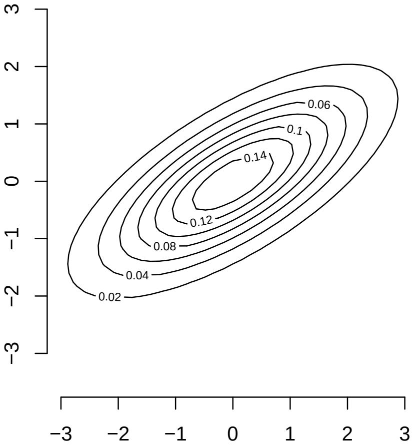
The MVN distribution is computationally extremely convenient – it is the only distribution for which the best (ML) estimator is the set of these sample moments:

- Estimate μ with $T^{-1} \sum_{t=1}^{T} Y_t$
- Estimate Σ with $T^{-1} \sum_{t=1}^{T} (Y_t - \hat{\mu})(Y_t - \hat{\mu})'$

$\operatorname{diag} \Sigma = \operatorname{diag} \operatorname{Var} Y_t$, and the off-diagonal elements are sample covariances $\widehat{\operatorname{Cov}}(Y_t^{(i)}, Y_t^{(j)})$.

Visualisation of the MV normal

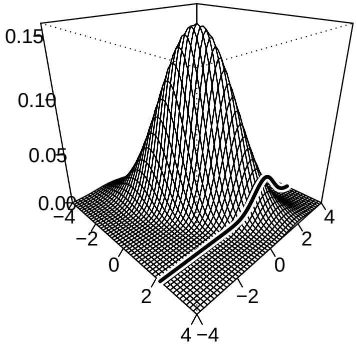


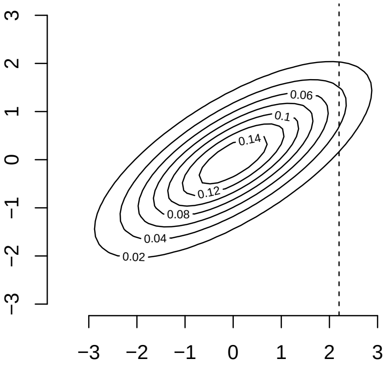


The contour lines of a MVN are elliptical – hence the name ‘spherical disturbances’ for multivariate Gaussian WN!

Marginals of the MV normal

Since under joint normality, BLP($Y \mid X$) = BP($Y \mid X$), knowing the value of one coordinate of a MVN vector reduced uncertainty about other coordinates.





Panel data

- Cross-sections: $\{Y_i\}_{i=1}^{n}$
- Time series: $\{Y_t\}_{t=1}^{T}$
- Panels: $\left\{\{Y_{it}\}_{t=1}^{T}\right\}_{i=1}^{n}$
  - Stacked time series $Y_t \coloneqq \{Y_t^{(1)}, \dots, Y_t^{(n)}\}_{t=1}^{T}$

Typically, each unit has its own level (fixed effect) and its own variances. Units often have cross-sectional correlations: $\mathrm{Cov}(Y_{it}, Y_{jt}) \neq 0$.

In TS analyses, it is not uncommon (but not mandatory) to scale the series to have zero mean unit variance.

Common-factor model idea (skipped)

- Consider a wide panel with many time series $Y_t \coloneqq \{Y_t^{(1)}, \dots, Y_t^{(n)}\}_{t=1}^T$
- Suppose that they are driven by $k \ll n$ dynamic common factors $F_t \coloneqq \{f_t^{(1)}, \dots, f_t^{(k)}\}$

Then, an **exact dynamic factor model** is written as follows:

$$Y_t = \Lambda(L)F_t + U_t, \quad \Psi(L)F_t = V_t,$$

U_t and V_t are serially uncorrelated with $\operatorname{Var} U_t = \Sigma_U$ and $\operatorname{Var} V_t = \vec{1}$. $\Lambda_{i,\cdot}(L)F_t$ is the common-factor contribution to Y_{it}.

Identification: $\mathbb{E}F_tU_t = 0$, $E(U_t V'_{t-k}) = 0 \quad \forall k$ (lags and leads).

State-space models

Constant-parameter linear state-space model with k state variables and n observed factors:

$$\begin{cases} \Psi(L)F_t = V_t & \text{(state)} \\ Y_t = \Lambda(L)F_t + U_t & \text{(observation)} \end{cases}$$

$\operatorname{Var} U_t = \Sigma_U$, $\operatorname{Var} V_t = \Sigma_V$; in many applications, $\Sigma_V = I_k$.

- ARIMA models and various smoothing filters
- Season-trend-irregular model (structural time series)
- Multivariate structural models, dynamic factor models

Kalman filter

- The states / factors F_t are unobserved and cannot be simply recovered from the observed signals
- If the states are known, then, the signals can be predicted by linear combinations of factors
- Estimation idea: pick such Ψ, Λ, and initial state guesses that $\hat{Y}_t(F_t)$ be close to the observed Y_t

Not only Y_{t+1} can be forecast from the current state – the missing entries in the middle can be filled in.

Kalman filter workflow

Recall ARIMA models and generalise them:

1. An initial guess about the state variables (usually zero) is made, $\hat{F}_0$
2. At $t = 1$, the predicted states are computed, $\hat{F}_t$
3. The predicted signals are computed, $\hat{Y}_t(\hat{F}_t)$
4. The prediction error is computed, $\hat{U}_t := Y_t - \hat{Y}_t(\hat{F}_t)$
5. The prediction error is used to correct the state at the present step with a Kalman gain (time-varying matrix), $\tilde{F}_t \coloneqq \hat{F}_t + K_t \hat{U}_t$
6. Repeat from step 2 by predicting the state at $t + 1$ using the corrected present state $\tilde{F}_t$

State updating in the Kalman filter

Estimating the mean of Y_t with no factors and no error auto-correlation: $Y_t = X_t + U_t$, $X_t = c$, $U_t \sim \mathcal{N}(0, 1)$.

1. Assume $X_0 = 0 \;\Rightarrow\; X_1 = 0$ is the predicted state
2. $\hat{U}_1 = Y_1 - 0 = Y_1$
3. Correct the state with a Kalman gain of 1, $\tilde{X}_1 = Y_1$
4. At $t = 2$, the predicted state is $\hat{X}_2 = Y_1$, $\hat{U}_2 = Y_2 - Y_1$, and the Kalman gain is $1/t = 1/2$
5. $\tilde{X}_2 = Y_1 + (Y_2 - Y_1)/2 = (Y_1 + Y_2)/2$

Simplest Kalman filtering: adjust the estimate of the mean by $1/t$ times the difference between the observed Y_t and predicted state (sample average of $(t-1)$ terms of Y_t).

Motivation for principal components

State-space models are often hard to interpret: whatever unobserved guesses minimise the measurement error is the ‘state’.

- The dimensionality of the state space is unknown, the mechanisms are potentially non-linear $\Rightarrow$ virtually everything is unknown in SSMs
- We want to create one informative index variable from the observable ones
- Create a linear combination of variables that has the highest variability (= information content)

Sufficient dimensionality reduction

Imagine a high-dimensional problem: the dependent variable Y_t may or may not depend on other variables X_t. How to select the relevant variables and keep the problem low-dimensional without losing too much information?

Suppose that $\dim X_t = n \gg T$ is high. Consider a fixed matrix $\mathrm{B}_{k\times n}$ with a small row dimension k such that $\mathbb{E}(Y_t \mid X_t) \approx \mathbb{E}(Y \mid \mathrm{B}X_t)$ and $k \ll T$.

More generally, consider finding such B that Y_t be conditionally independent of X_t once the effects of $\mathrm{B}X_t$ have been taken into account (recall the FWL theorem).

PCA minimisation problem

Given any vector of *n* variables $\{Y_t^{(1)}, \dots, Y_t^{(n)}\}$, remix it into *n* linear combinations $\{P_t^{(1)}, \dots, P_t^{(n)}\}$:

- The coordinates of P_t are orthogonal
- $\operatorname{Var} \sum_i Y_t^{(i)} = \operatorname{Var} \sum_i P_t^{(i)} = \sum_i \operatorname{Var} P_t^{(i)}$
- $\operatorname{Var} P^{(1)}$ is the highest possible variance under the aforementioned constraints

This ‘remixing’ through linear combinations is often called ‘rotation’ and denoted via the rotation matrix $\mathrm{R}_{n\times n}$.

Example: regress the child height not onto father’s and mother’s height, but their average height and height difference.

PCA technical representation

- Estimate $\Sigma \coloneqq \operatorname{Cov} Y_t$ (or better, $\operatorname{Cor} Y_t$)
- Calculate eigenvalues and eigenvectors of $\hat{\Sigma}$
  - Eigenvalues $\det(\hat{\Sigma} - \lambda I_n) = 0$
  - Eigenvectors: $(\hat{\Sigma} - \lambda I_n)v = 0$
- Sort the eigenvalues, take the eigenvector corresponding to the largest eigenvalue
- Take a linear combination of Y_t with weights given by this eigenvector

Then, $\operatorname{Var} \sum_i Y_t^{(i)} = \sum_i \operatorname{Var} P_t^{(i)} = \sum_i \lambda_i$.

PCA intuition

PCA is equivalent to fitting a multivariate normal distribution to the data and then, projecting them onto the principal axes of the ellipse.

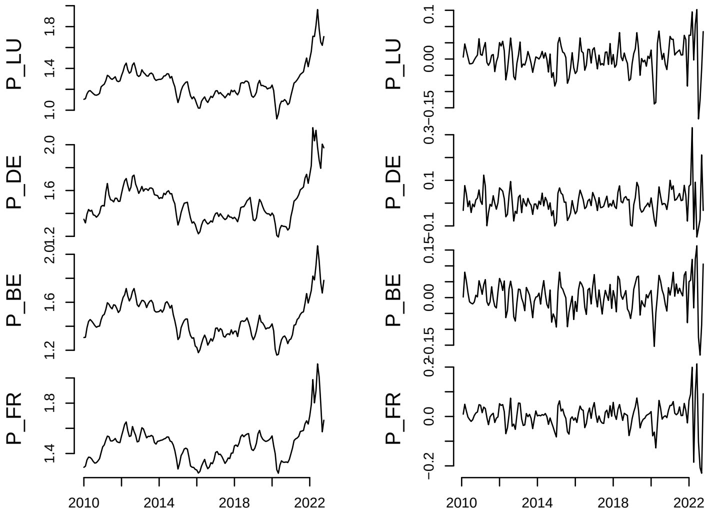
- Dimensionality reduction for a 3D bottle: only the vertical dimension (level of liquid) contains useful information
- Dimensionality reduction for a 3D pizza: two out of three dimensions (top view) contains information about the remaining pizza

If the data do not look like multivariate ellipses, PCA may produce strange results.

Practical aspects of PCA

- Variance manipulations make sense only for finite variances $\Rightarrow$ only stationary variables should be transformed
- No universal rule 'how many components should one take'
  - Reducing 2000 stocks into 30 portfolios is a good idea
  - As long as $\lambda_i > 1$, there is dimensionality reduction
  - Psychological level: 70%? 90%?
  - **Best:** some application-driven criterion (via cross-validation)
- If the correlation structure changes, chunking / localisation may yield an improvement

Example: petrol (SP95) prices



Correlation-based PCA in R

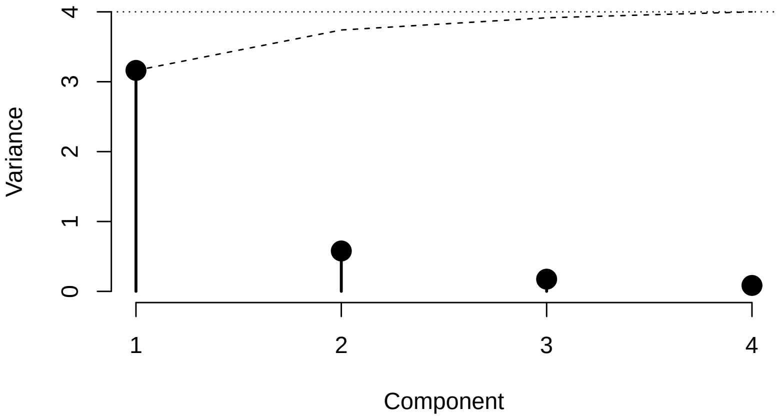
```
dp <- diff(fuel_prices)
p  <- prcomp(dp, scale. = TRUE)
str(p) # A list
```

```
 $ sdev     : num [1:4] 1.78 0.76 0.42 0.29
 $ rotation: num [1:4, 1:4] -0.541 -0.415 ...
 $ center   : Named num [1:4] 0.0039 0.0041 0.0031 0.0024
  ..- attr(*, "names")= chr [1:4] "P_LU" "P_DE" "P_BE" "P_FR"
 $ scale    : Named num [1:4] 0.047 0.057 0.05 0.053
 $ x        : num [1:153, 1:4] 0.19 -2.287 -1.22 0.038 0.484 ...
```

The components are stored in the `$x` list element, and the linear combination weights in `$rotation`.

In this example, since the series are quite similar, $P^{(1)} = 0.54\Delta Y^{\mathrm{LU}} + 0.42\Delta Y^{\mathrm{DE}} + 0.53\Delta Y^{\mathrm{BE}} + 0.50\Delta Y^{\mathrm{FR}}$.

Scree plots



Imputation

Missing data

- Real time series contain missing values
- Guessing the missing values is called **imputation**
- The missingness mechanism can be random or non-random
  - In cross-sectional model, missingness completely at random (MCAR) does not create biases (merely reduces the effective sample size)
  - Missingness at random (conditionally on exogenous variables) leads to inefficiency but not bias
  - Missingness at random (conditionally on all observable regressors, MAR) may lead to biases if not addressed
  - Missingness non at random requires extra fortifying assumptions to enable identification

Imputation basics

- Consider the stock prices of a medium-cap company. Long periods with zero trades ⇒ no price observed ⇒ no 5-minute returns for high-frequency traders
  - Structural solution: extrapolate the previous observed values of price ⇒ zero price changes, zero trade volume
- In many cases there are no natural ways to carry out imputation, and one need models for accurate guessing
  - Guesswork mixes the true DGP with the imputation model
- Nowcasting: predicting past or present values
  - Delays in collection of statistics, ‘ragged’ panel edges
  - GDP: quarterly, labour data: monthly. Can we reconstruct the monthly GDP by observing the co-evolution of the two mixed-frequency series?

In the TS context, missingness may create extra biases.

Imputation using ARMA

- If there are gaps inside a time series, one can estimate the model on the data before the gaps and use predictions for the imputed values
  - Some researchers add lead values of Y_t to the ARMA model to improve imputation accuracy:
    $Y_t = \mu + \varphi_1 Y_{t-1} + \xi_1 Y_{t+1} + U_t + \theta_1 U_{t-1}$
- Another approach: insert some values (e g. median) and estimate SARIMA with additive outliers
  - ARIMA filtered values can be used for imputation
- TRAMO-SEATS-like: remove S_t, extract I_t, impute it, extrapolate T_t and S_t, combine all

Imputation using Kalman filtering

- Write the trend + seasonal + SARIMA model in the state-space representation
- Apply the Kalman filter to predict the next state
- Compute T_t, S_t, I_t from the state values

See Hyndman, Koehler, Ord, Snyder (2008) for details and the `imputeTS` package.

Imputation using PCA

For several time series, iterate:

- Substitute the missing values by something (e. g. the mean or $(Y_{t+1} + Y_{t-1})/2$
- Apply PCA to these multivariate time-series (compute the rotation matrix)
- Using the chosen number of components k and singular-value decomposition (SVD), reconstruct Y_t
- Iterate until convergence

In R, use the `missMDA` package.

Multiple imputations

So far, we have been making point forecasts / predictions based on the model and asymptotic theory for inference.

- Recall: $\operatorname{Var} \hat{Y}_t \le \operatorname{Var} Y_t$ because of the multivariate Pythagorean theorem ($\hat{Y}_t$ is a linear projection of Y_t onto the linear space of Ω_{t-1})
  - Estimation uncertainty: the estimates are not the truth
- Under-estimating the second-order uncertainty: $T^{-1} \sum_t (Y_t - m)^2$ is minimised at the sample average

Recall forecast combination: combining many guesses based on different models improves accuracy. Can we add some randomness to guesses to gauge the impact of these uncertainties on the forecast?

Resampling methods

General idea: sample from the conditional distributions $f(Y^{(1)} \mid Y^{(-1)}; \theta_1)$, ..., $f(Y^{(n)} \mid Y^{(-n)}; \theta_n)$, where $\theta_1, \dots, \theta_n$ are the parameters of the conditional distributions, not necessarily linked to the true joint distribution f_Y.

- Choose the components (possibly all) to impute $Y^{(i)}$, assume some convenient conditional distribution that depends on θ_i
- Approximate the distribution of θ_i, draw $\hat{\theta}_i$ from it, predict the missing values

Multiple imputation for multiple TS

Using sufficient dimensionality reduction, use one of the following algorithms:

- MI with regularised regression
- MI with sequential penalised regression
- MI with recursive partitioning and predictive mean matching
- MI with PCA
  - Flavours: simple; bootstrapped with generalised CV; bayesian MI + regularised PCA

AMELIA II algorithm

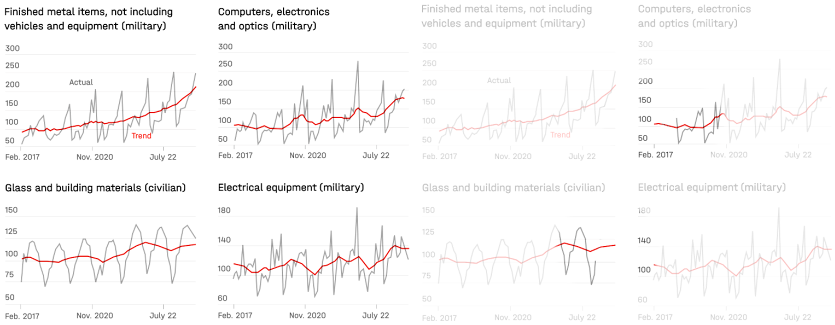
Assuming (1) multivariate normality of the data and
(2) missingness at random (conditioning on observables):

- Use a black box (the EM algorithm + bootstrap) to estimate parameters of the multivariate joint distribution
- Iteratively, draw the guesses from the conditional distributions at random
- With different initial seeds, obtain *m* data sets
- Run the analysis many times, analyse the stability of estimates under various random guesses

Imputation diagnostics

- Plot the imputed time series and their differences!
- **Densities:** compare the distributions of the observed and imputed values
- **Over-imputation:** add several missing values and see how well they are guessed
- **Over-dispersion:** for multiple-imputation algorithms, try ‘wilder’ initial values and analyse the convergence

What you can do now



Final advice

- Plot the object of study before and after applying statistical methods
- Think about all possible dependencies between the variables, and then, make simplifying assumptions
  - From general to specific
- Write modular code, convenience wrappers, macros
- **Never fall in love with your models** and never assume that a model is the ultimate reality decomposition
  - Try many models and many approaches
  - Explore new packages and new methods

Thank you for your attention!