Financial Time Series Analysis

Yuri F. Saporito
yuri.saporito@fgv.br

Raiffeisen Bank
March, 2024

Time Series Analysis

Time Series Analysis deals with sequential data, ordered by time. It could be used for the following situations, for instance:

- Compute the risk of a financial portfolio. For this, one needs to understand the future distribution of the multivariate time series with stochastic volatility (hence, a statistical inference problem).
- Predict the interest rate curve. For this, one could use dimensionality reduction, classical prediction using classical time series models (e.g. VAR) and Machine Learning techniques (e.g. LSTM).
- Fama–French factors and their capacity to explain the cross-sectional of returns. In this situation, one needs Time Series Regression techniques.

Overview

References

For the classical part of Time Series modeling and analysis, these two references are excellent:

- Kevin Sheppard. Financial Econometrics Notes, 2021. https://www.kevinsheppard.com/files/teaching/mfe/notes/financial-econometrics-2020-2021.pdf
- Rob J Hyndman and George Athanasopoulos. Forecasting: Principles and Practice 2021. https://otexts.com/fpp3/
- Klaus Neusser. Time series econometrics. Springer, 2016. https://link.springer.com/book/10.1007/978-3-319-32862-1

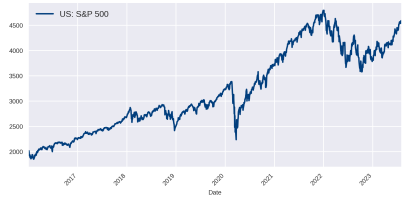
For the Machine Learning part, there are several online references.

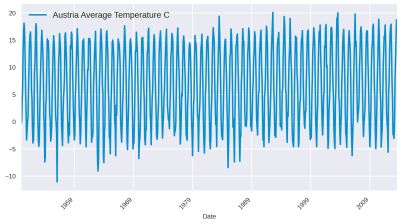
opentimeseries.com

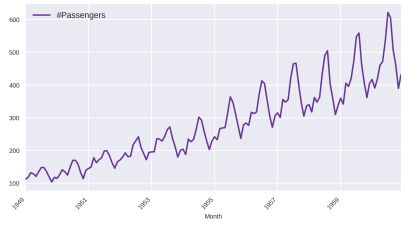
Great online resource `opentimeseries.com`

Time Series Decomposition

Examples of Time Series







Time Series Decomposition

Based on the examples of time series of the previous slides, we may observe the following behaviors:

- Trend: long-term monotonic pattern (increase or decrease), the *direction* of the time series;
- Seasonal: pattern of fixed and known frequency, denoted by S;
- Cyclic: rises and falls of non-fixed frequency; related to *business cycles*.
- Residual: whatever is not explained by the behaviors above, it is included in the residual, denoted by R.

It is common to combine Trend and Cyclic patterns into one component, Trend-Cycle, denoted by T.

Time Series Decomposition

Mathematically, we consider two possible decomposition of the time series X into the patterns we saw in the previous slide:

- Additive: $X_t = T_t + S_t + R_t$;
- Multiplicative: $X_t = T_t \cdot S_t \cdot R_t$.

Additional comments:

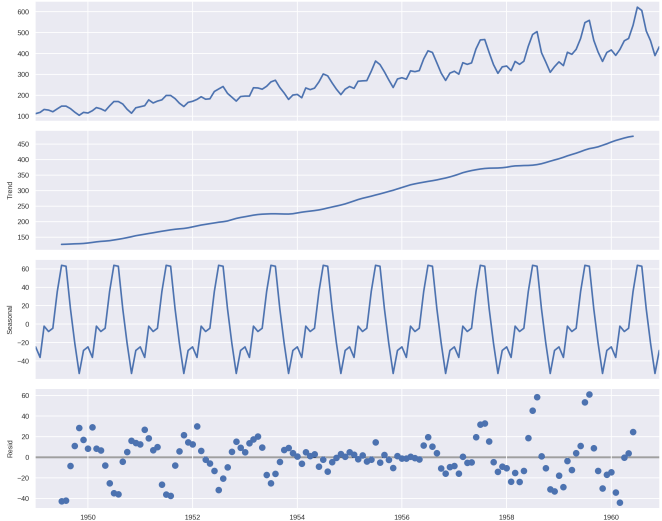
- The multiplicative decomposition is equivalent to the additive decomposition of the log of the time series:

$$\log X_t = \log T_t + \log S_t + \log R_t$$

- Once the decomposition is computed we might consider a seasonally adjusted series: $X_t - S_t$ or X_t / S_t.

Airline Passengers

Additive decomposition computed using `statsmodel`'s function `seasonal_decompose`.



Time Series Decomposition

How is the decomposition calculated? There are several algorithms to perform this task. Let us first describe the classical method:

- Trend-Cycle is estimated using a (central) moving average of order p:

$$\widehat{T}_t = \frac{1}{2p+1} \sum_{k=-p}^{p} X_{t+k}.$$

  The average makes it the series smoother; the largest the p, the smoother becomes $\widehat{T}_t$.

- Compute the detrended time series: $Y_t = X_t - \widehat{T}_t$.

- The estimator $\widehat{S}_t$ of seasonal component is the average of the detrended series Y_t for the chosen season (monthly, yearly, weekly, etc).

- The residual is simply $\widehat{R}_t = X_t - \widehat{T}_t - \widehat{S}_t$

Time Series Decomposition

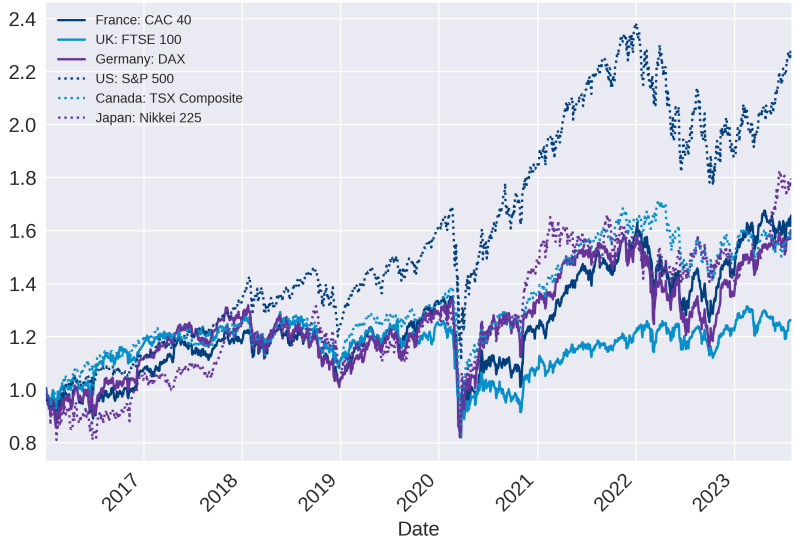
There are several other methods to compute time series decomposition that we list now:

- X11: it is based on the classical decomposition, but deals with several drawbacks of that classical method. For more details, see link.
- SEATS: (Seasonal Extraction in ARIMA Time Series) uses the ARIMA model that we will study in this course. It works only with quarterly or monthly data. Details here: link.
- STL: (Seasonal and Trend decomposition using Loess) uses locally estimated scatterplot smoothing (loess) to estimate the trend-cycle term. It involves an iterative procedure, more information here: link.

Jupyter Notebook `Time Series - TS Decomposition`.

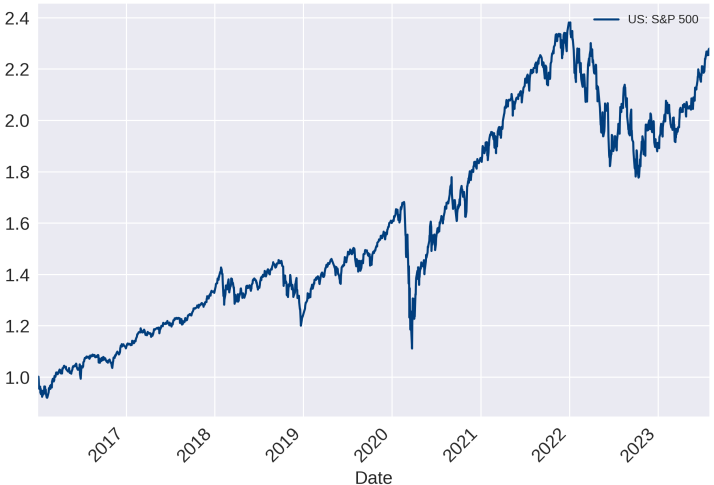
Time Series Modeling

Stock Prices



Stock Prices

Let us consider the S&P 500 index:



Stock Price Data

- We have used `yfinance` (Yahoo Finance) to download the stock price data.
- The following columns are available: Open, Low, High, Adj. Close, Close and Volume.
- Open: is the price at which a stock started trading when the trading day started. It might be equal to the previous close, but it could be affected by after-hours trading.
- Low and High: the lowest and highest price the stock has traded during the trading day.
- Close: is the price for the last transaction before the market closes, sometimes adjusted for splits.
- Volume: the number of shares traded in a stock during the trading day.

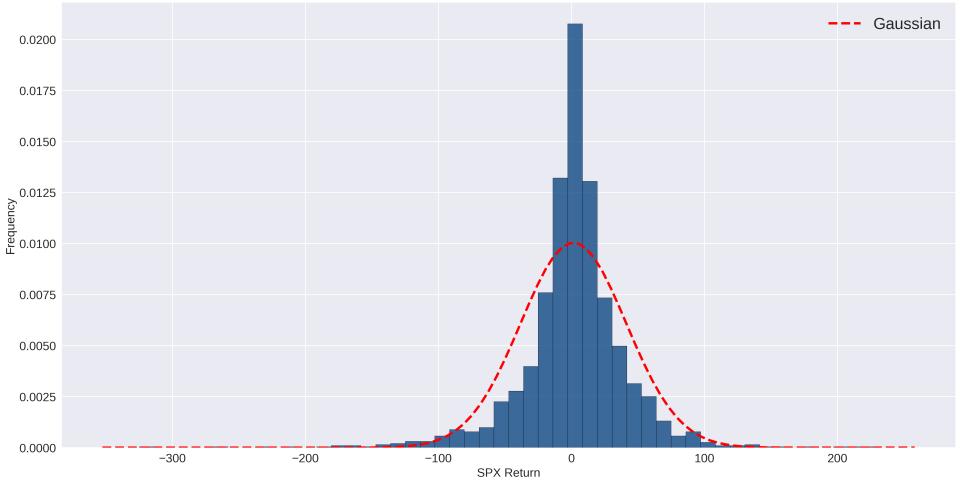
Adjusted Closing Price

There are three events that adjust the close price:

- Dividends: if the company pays $5 per share and is trading at $100, the Adjusted Closing Price will be $95.
- Splits: the company may split the its shares into smaller pieces if they stock price becomes too expensive. The market capitalization and the value of each stock holder remain the same. For instance, if the company splits its stock in a 2:1 ratio and the stock price is $100, the new Adjusted Closing Price will be $50.
- New offerings: it is usually done to raise more capital for the company. Since there are more individual shares, each share represents a lower portion of the total value and then the offering reduces the value of the existing stock. It may be offered preferably to existing shareholders.

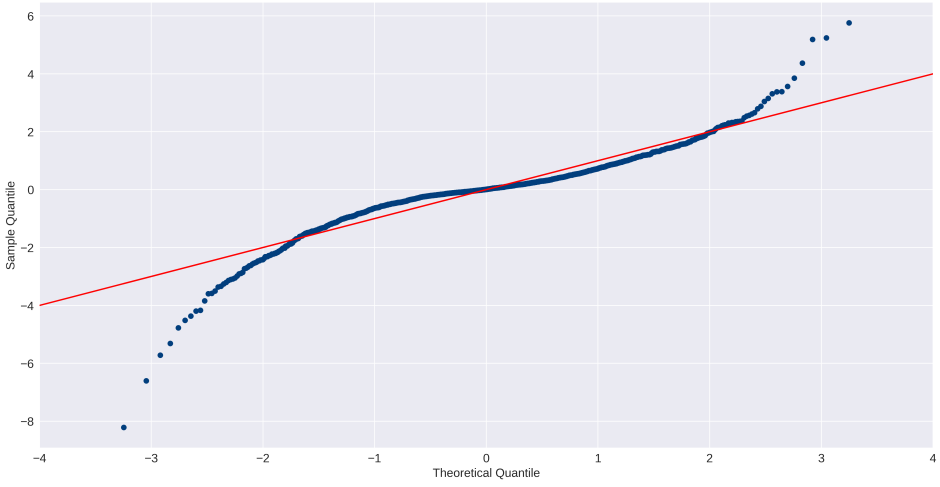
Stock Prices

We want to understand the distribution of returns. Let us first plot the histogram:



Stock Prices

Now the qq-plot:



Time Series Modeling

- We need models that creates those stylized facts (and others that we will see later).
- In Statistics, we learn how to create distributions with given characteristics: uni-variate, multivariate, fat-tailed, positive, in a given interval, etc, and how to estimate their parameters.
- Several of these distributions start with very simple ones, like Uniform and Gaussian.
- And to create more complex distributions we could modify some moments of the desired distribution as the mean, variance, covariance, etc.
- Therefore, we need to create some simple "distributions" for time series and and a way to make them more complex.

Time Series Analysis

- Compute the risk of a financial portfolio. For this, one needs to understand the future distribution of the multivariate time series with stochastic volatility (hence, a statistical inference problem).
- Predict the interest rate curve. For this, one could use dimensionality reduction, classical prediction using classical time series models (e.g. VAR) and Machine Learning techniques (e.g. LSTM).
- Fama–French factors and their capacity to explain the cross-sectional of returns. In this situation, one needs Time Series Regression techniques.

Time Series Modeling

- Particularly, for time series we are interested in the joint distribution of observations in different times.
- Let $\mathcal{T}$ be a discrete time index, as $\{0, 1, \ldots\}$ or $\{0, 1, \ldots, N\}$, and $X = (X_t)_{t \in \mathcal{T}}$ be a collection of random variable indexed by t (a time series).
- In principle, we would need to consider the distribution of the random vector $(X_{t_1}, \ldots, X_{t_n})$, for any $t_1, \ldots, t_n \in \mathcal{T}$; these are called the finite-dimensional distributions of X.
- In practice, for a wide class of models, we need much less than this: only moments of X_t and (X_t, X_s).

Stationary Processes

Stationary Process

- Before moving to these models, an important definition: we say that X is (strongly) stationary if

$$(X_{t_1+\tau}, \ldots, X_{t_n+\tau}) \sim (X_{t_1}, \ldots, X_{t_n})$$

  for any τ.

- That is, the distribution of time series X does not depend on the instant we start observing it (here denoted by τ).

- In particular, no trend could be observed in this time series.

- However, this is a very strong modeling assumption and weaker stationarity is considered.

Stationary Process

Before defining this weaker notion, some auxiliary definitions:

- Mean function: $m_X(t) = \mathbb{E}[X_t]$.
- Autocovariance function: $k_X(t, s) = \text{Cov}(X_t, X_s)$.

(Weakly) Stationary

We say that X is (weakly) stationary if $m_X(t)$ is constant in t and $k_X(t, s)$ depends only on $|s - t|$.

Stationary Process

If X is stationary, then

- $m_X(t) = \mu$;
- $\sigma_X^2(t) = \text{Var}(X_t) = \text{Cov}(X_t, X_t) = \sigma^2$;
- Autocorrelation function: $\rho_X(\tau) = \dfrac{\text{Cov}(X_t, X_{t+\tau})}{\sigma^2}$, for $\tau \geq 0$. Notice $\rho_X(0) = 1$.

White Noise

We say that $\varepsilon = (\varepsilon_t)_{t \in \mathcal{T}}$ is a white noise if ε is stationary with zero mean and zero autocovariance for $t \neq s$.

Estimation of the Autocorrelation function

- We can then estimate, for a given observed path of X, $\{X_1, \ldots, X_T\}$, the autocorrelation function by considering the following sample estimators for the covariance and variance of X:

$$\hat{k}(\tau) = \frac{1}{T} \sum_{k=1}^{T-\tau} (X_k - \bar{X}_T)(X_{t+\tau} - \bar{X}_T),$$

$$\hat{\rho}(\tau) = \frac{\hat{k}(\tau)}{\hat{k}(0)}, \quad \bar{X}_T = \frac{1}{T} \sum_{k=1}^{T} X_k.$$

- Notice that we are normalizing $\hat{k}$ by T and not $T - \tau$ (the quantity of terms in the sum). This create a bias in the estimator, however, as matrices varying $\tau = 0, 1, 2, \ldots, T$, the autocovariance and autocorrelation with this normalization are always positive definite.

Confidence Interval for $\hat{\rho}$

- Under stationarity assumption (and other mild technicalities), for any τ, the vector $(\hat{\rho}(1), \ldots, \hat{\rho}(\tau))$ is consistent estimator for $(\rho(1), \ldots, \rho(\tau))$, asymptotically Gaussian with covariance given by $\frac{W}{T}$, for a given covariance matrix W.

- Under the iid white noise case, one can show that the covariance matrix is given by $W = I$. Hence, $\sqrt{T}\hat{\rho}(\tau)$ converges in distribution to $N(0, 1)$ and hence, for instance, the asymptotic 95% interval for $\rho(\tau)$ is given by $\left(\frac{-1.96}{\sqrt{T}}, \frac{1.96}{\sqrt{T}}\right)$.

- One could also prove the same asymptotic confidence interval for PACF under the AR(p) model for $\tau > p$.

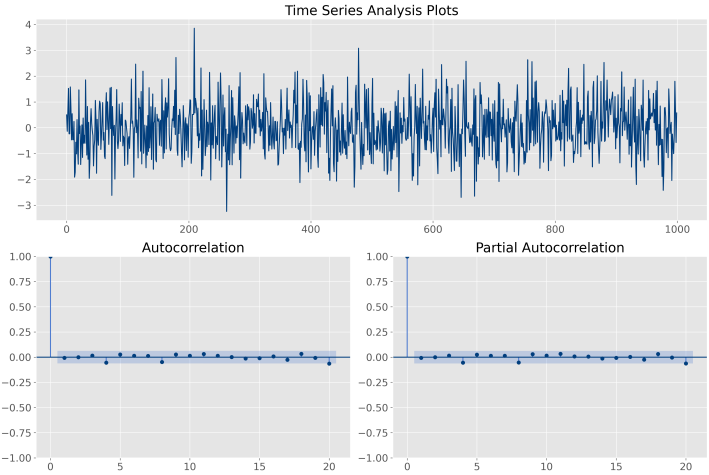
- Using these results, we could also test $\rho(1) = \rho(2) = \cdots = \rho(\tau) = 0$ using the Box-Pierce Q statistic:

$$Q = T \sum_{k=1}^{\tau} \hat{\rho}^2(k) \sim \chi^2_\tau,$$

(one could also use the Ljung-Box Q statistic).

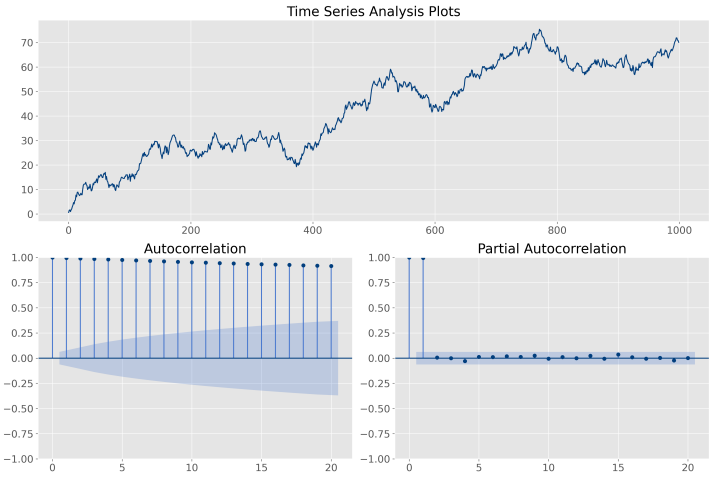
White Noise

One example of white noise is an iid sequence of mean 0 and variance σ^2 random variables, e.g. $\varepsilon_t \overset{iid}{\sim} N(0, \sigma^2)$:



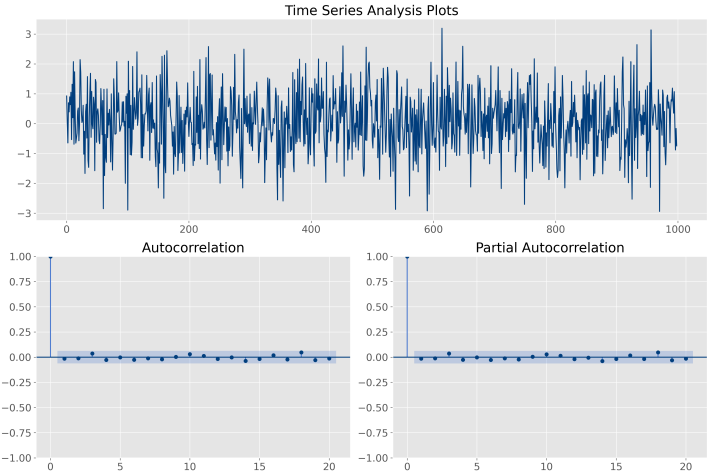
Random Walk

Once we have created the white noise process, we could consider the following simple and interesting modification, the so-called Random Walk: $X_t = X_{t-1} + \varepsilon_t$:



Random Walk

Clearly, if we consider the (temporal) difference of the process X, we recuperate the white noise: $\Delta X_t := X_t - X_{t-1} = \varepsilon_t$:



Remarks

A couple of remarks are in order:

- The direct dependence on the previous time $t-1$ is a financial sensible assumption: the price of the stock tomorrow could be seen as the price today plus some "non-correlated" noise.
- So, differentiating financial time series (i.e. considering the increments) could be a good idea to create stationary processes, as we will see. Although we will not explore it here, one could consider fractional differentiation, which usually creates stationary process and does not remove path dependence.
- We haven't seen precisely what the Partial Autocorrelation plot means, but it clearly gives a better understanding of time dependence as we saw in the Random Walk plot.

More on the Random Walk

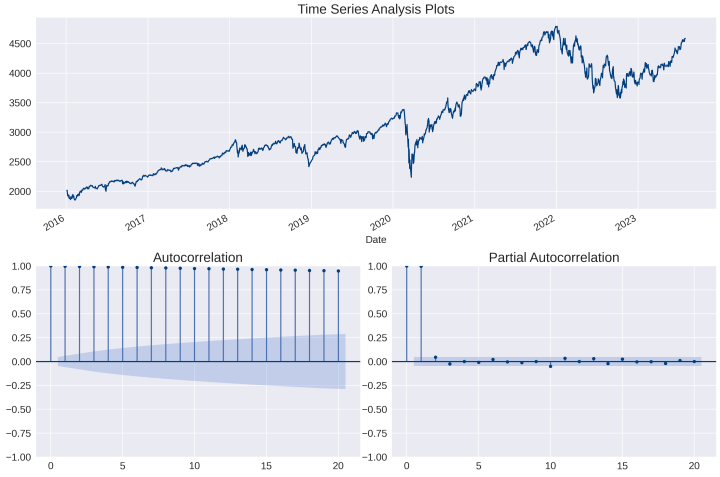
The random walk is not stationary:

$$X_t = X_{t-1} + \varepsilon_t = X_{t-2} + \varepsilon_{t-1} + \varepsilon_t = \cdots = X_0 + \sum_{k=1}^{t} \varepsilon_k,$$

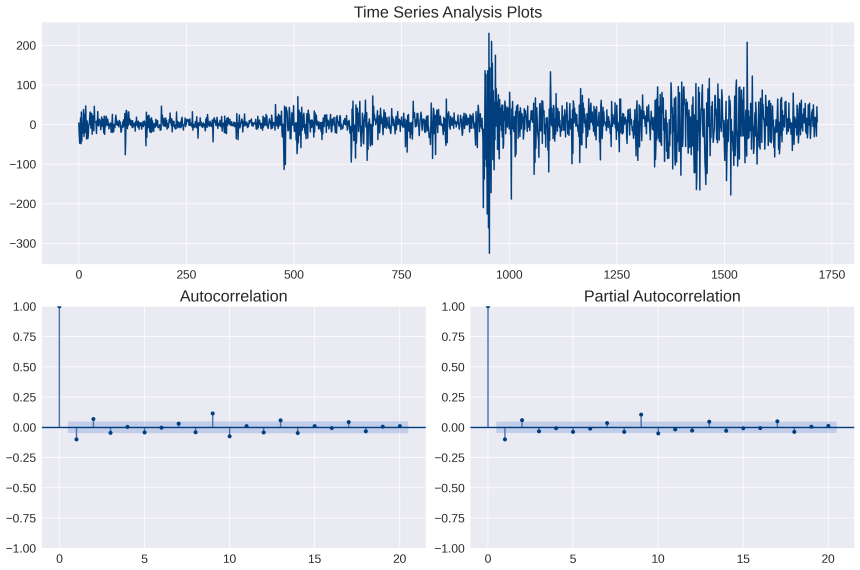
- $m_X(t) = X_0 + \sum_{k=1}^{t} \mathbb{E}[\varepsilon_k] = 0$;
- $\sigma_X^2(t) = \sum_{k=1}^{t} \mathrm{Var}[\varepsilon_k] = \sigma^2\, t$;
- Therefore, the variance grows linearly with time.

SPX

Let us now analyze some real stock price data:



SPX - Difference



More Remarks

- Some lags seems to have some (although small) effect in the current value of the increment.
- The volatility of the increments is clearly not constant and appear to be stochastic.
- The distribution of the increments also appear to have fat tails (fatter than Gaussian).
- Since the index value and stock prices are positive, we could (should) have considered the log-price and its difference: $Y_t = \log(X_t)$ and $\Delta Y_t = \log(X_t/X_{t-1})$.
- We will now address the first item, i.e. consider models in which X_t might depend on previous observed values of X.
- Do Questions 1 and 2 on the 

Jupyter Notebook `Time Series - ARMA, GARCH and VaR`.

Autoregressive (AR), Moving-Average (MA) and ARMA Models

Autoregressive Models

- Following the discussion from the previous slides, we consider the autoregressive model:

$$X_t = a_0 + a_1 X_{t-1} + \cdots + a_p X_{t-p} + \varepsilon_t,$$

  where $p \geq 1$ is the order of the model and $\varepsilon = (\varepsilon_t)_{t \geq 0}$ is a white noise.

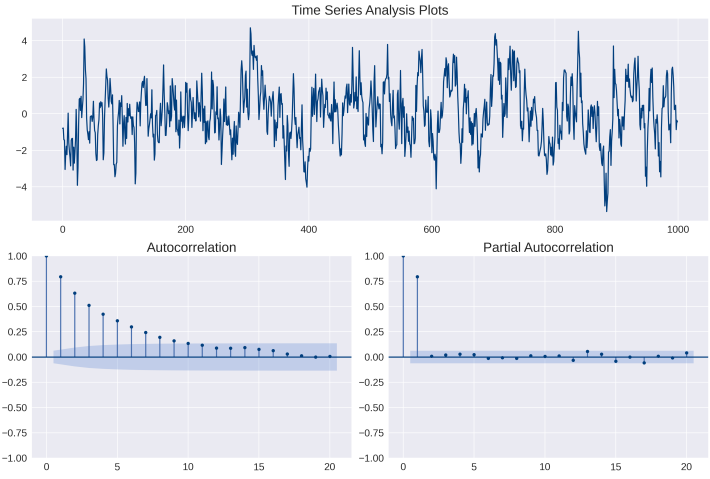
- We denote these models by AR(p).

- If you known what is a Brownian motion: the AR(1) could be seen as a time discretization of the mean-reverting stochastic process known as Ornstein-Uhlenbeck: $dX_t = \kappa(m - X_t)dt + \sigma dW_t$.

- Let us simulate an AR(1) and consider the autocorrelation analysis we have done so far.

AR(1)

$X_t = 0.8\,X_{t-1} + \varepsilon_t$, with $\varepsilon_t \overset{iid}{\sim} N(0,1)$:



Partial ACF

- We are now able to properly define the PACF (Partial Autocorrelation Function).
- The issue when when computing the ACF of an AR(p) model (or any other model for that matter) is that if X_t depends on X_{t-1}, then X_{t-1} will depend on X_{t-2}, which implies then that X_t also depends on X_{t-2} and so on.
- We want to remove this (linear) dependence, i.e. we want to compute the correlation of X_t and X_{t+k} removing the (linear) dependence of X_{t+1} up to X_{t+k-1}.
- Hence, consider the prediction of X_{t+k} using $X_{t+k-1}, \dots, X_t$:

$$X_{t+k} = \alpha_1 X_{t+k-1} + \alpha_2 X_{t+k-2} + \cdots + \alpha_k X_t + \eta_{t+k},$$

  where η_{t+k} is the prediction error, uncorrelated to the covariates $X_{t+k-1}, \dots, X_t$.
- We are then computing the contribution of X_t for the prediction of X_{t+k}, controlling for $X_{t+k-1}, \dots, X_{t+1}$, i.e. α_k is a measure of feature importance of X_t.

Partial ACF

- α_k is the PACF or order k;
- If X is a AR(p) process, to predict X_{t+k}, for $t + k - 1 > p$, we need only the last p time observations, so $\alpha_k = 0$;
- One can prove that, for a mean-zero stationary process X, $\alpha_1 = \text{Corr}(X_2, X_1)$ and

$$\alpha_k = \text{Corr}(X_{k+1} - \hat{X}_{k+1}, X_1 - \hat{X}_1),$$

  where $\hat{X}_{k+1}$ and $\hat{X}_1$ is the linear prediction of X_{k+1} and X_1, respectively, using $X_2, \ldots, X_k$;
- Box-Jenkins methodology: for an AR(p) process, the ACF declines exponentially (monotonically or oscillating) to zero and the PACF is 0 for lags $k > p$.

AR(1) is Stationary

- Let us now analyze the stationarity of an AR(1) model: $X_t = aX_{t-1} + \varepsilon_t$:

$$X_t = a(aX_{t-2} + \varepsilon_{t-1}) + W_t = a^2 X_{t-2} + a\varepsilon_{t-1} + \varepsilon_t = \cdots$$
$$= a^{k+1} X_{t-k-1} + \varepsilon_t + a\varepsilon_{t-1} + \cdots + a^k \varepsilon_{t-k}$$

- Assume that we could repeat this process forever and that $|a| < 1$ (this is a MA(∞)):

$$X_t = \sum_{k=0}^{+\infty} a^k \varepsilon_{t-k}$$

- Hence, $m_X(t) = 0$ and

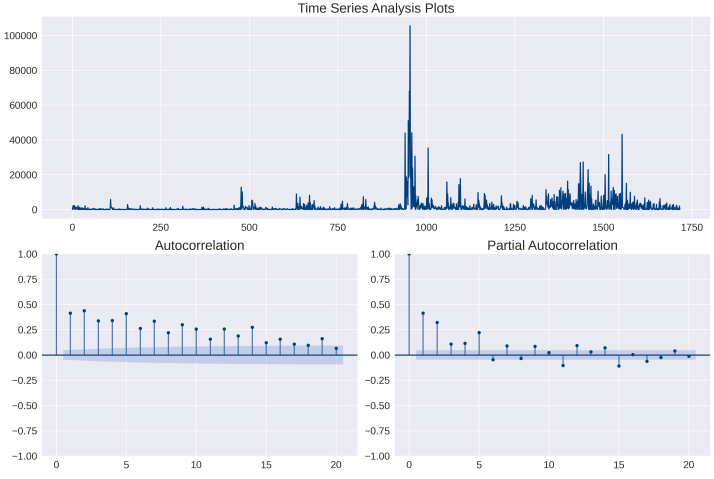
$$k_X(t, t+\tau) = \sum_{k=0}^{+\infty} \sum_{j=0}^{+\infty} a^{k+j} \mathbb{E}[\varepsilon_{t-k} \varepsilon_{t+\tau-j}]$$
$$\overset{j=k+\tau}{=} \sum_{k=\tau}^{+\infty} a^{2k-\tau} \mathbb{E}[\varepsilon_{t-k}^2] = \sigma^2 a^\tau \sum_{k=0}^{+\infty} a^{2k} = \sigma^2 \frac{a^\tau}{1-a^2}$$

When is AR(1) is stationary?

- Therefore, if $|a| < 1$, then the AR(1) is stationary;
- The case $|a| = 1$ ($\Leftrightarrow$ unit root of $r^2 - a = 0$) is the random walk and we have seen that it is not stationary;
- This dichotomy is also observed for general AR(p) models, but for more general unit root: $r^p - a_1\, r^{p-1} - \cdots a_{p-1}\, r - a_p = 0$;
- The most well-known statistical test for unit root of AR(p) (or test for stationarity) is the augmented Dickey-Fuller (ADF) test.

SPX - Difference Squared

As we have seen for the ΔSPX_t, the volatility seems stochastic, so consider $(\Delta SPX_t)^2$:



Moving Average Models

- Analyzing the ACF and PACF for the squared difference series, we see that white noise or AR models are the appropriate choices.
- We need then a new model so we could explain the stochastic volatility of these financial time series.
- We hence consider the MA(q) model:

$$X_t = b_0 + \varepsilon_t + b_1\varepsilon_{t-1} + \cdots + b_q\varepsilon_{t-q},$$

where $(\varepsilon_t)_{t\in\mathcal{T}}$ is a white noise process.

- Box-Jenkins methodology: one could show that for an MA(q) process, the ACF is 0 for lags $k > q$ and the PACF declines exponentially (monotonically or oscillating) to zero.

Moving Average Models

- Let us further analyze the MA(1) model: $X_t = b_0 + \varepsilon_t + b\varepsilon_{t-1}$.
- This could be seen as a moving average of ε_t and ε_{t-1}.
- Notice that $\mu_X(t) = b_0$ and

$$\begin{aligned} k_X(t+\tau, t) &= \mathrm{Cov}(X_{t+\tau}, X_t) = \mathrm{Cov}(\varepsilon_{t+\tau} + b\varepsilon_{t+\tau-1}, \varepsilon_t + b\varepsilon_{t-1}) \\ &= \mathbb{E}[\varepsilon_{t+\tau}\varepsilon_t] + b\mathbb{E}[\varepsilon_{t+s}\varepsilon_{t-1}] + b\mathbb{E}[\varepsilon_{t+\tau-1}\varepsilon_t] + b^2\mathbb{E}[\varepsilon_{t+\tau-1}\varepsilon_{t-1}], \end{aligned}$$

  and hence, the ACF is

$$\rho_X(t+\tau, t) = \begin{cases} 1, \text{ if } s = 0, \\ \dfrac{b}{1+b^2}, \text{ if } s = \pm 1, \\ 0, \text{ otherwise.} \end{cases}$$

- Therefore, the MA model is always stationary, independent of the parameters b's.

Moving Average Models

- For AR models, the values of the PACF could be used as estimators for the AR coefficients $(a_1, \ldots, a_p)$. The same is not true for MA models.
- We have seen that in the case of the MA(1) model, $\rho_X(0) = 1$, $\rho_X(\pm 1) = \frac{b}{1+b^2}$, and $\rho_X(\tau) = 0$, if $|\tau| > 1$.
- We then pose the question if it is possible to find the MA(1) model if we know the autocorrelation function $\rho_X(0) = 1$, $\rho_X(\pm 1) = c$, and $\rho_X(\tau) = 0$, if $|\tau| > 1$. This implies that

$$\frac{b}{1+b^2} = c \Leftrightarrow c\, b^2 - b + c = 0 \Leftrightarrow b_\pm = \frac{1 \pm \sqrt{1-4c^2}}{2c}.$$

- Hence, if $|c| < 1/2$, there are two possible MA(1) models that give the same autocorrelation function; if $c = \pm 1/2$, there exists only one MA(1); finally, if $|c| > 1/2$, there is no MA(1) model.

MA(1)

Two MA(1) processes: $b = 2$ (dark blue) and $b = 0.5$ (light blue), both generate $\rho(1) = 0.4$:

ARMA Models

- Putting AR and MA models together gives us the well-known ARMA(p, q) models:

$$X_t = a_0 + a_1 X_{t-1} + \cdots + a_p X_{t-p} + \varepsilon_t + b_1 \varepsilon_{t-1} + \cdots + b_q \varepsilon_{t-q},$$

  with $a_1 + \cdots + a_p \neq 1$, where $(\varepsilon_t)_{t \in \mathcal{T}}$ is a white noise process.

- An important case is the ARMA$(1, 1)$:

$$X_t = a_0 + a_1 X_{t-1} + \varepsilon_t + b_1 \varepsilon_{t-1}.$$

- Do Question 3 on the Jupyter Notebook `Time Series - ARMA, GARCH and VaR`.

ARMA Estimation and Model Selection

Estimation of ARMA Models

- Estimation of AR(p) models could be performed in quite straightforward ways as the Yule-Walker estimator (that considers sample estimation of autocovariance function and their relation to the model parameters).
- Another approach would be to see the AR(p) as a regression of X_t against the covariates $X_{t-1}, \ldots, X_{t-p}$. The model parameters then could be estimated using standard OLS (but one should remove the first p observations).
- This estimation procedure might be troublesome because the covariates could be quite correlated, implying that the OLS estimator is no longer unbiased.
- However, one could prove that the estimator is consistent and asymptotically Gaussian.
- These approaches breakdown for MA(q) processes since one cannot identify the parameters using the autocovariance functions anymore and that the past values of the residual are not observable.

Estimation of ARMA Models

- To estimate the full ARMA(p, q) model, the most used method is the maximum likelihood estimator assuming iid Gaussian white noise $\varepsilon = (\varepsilon_t)_{t \in \mathcal{T}}$.
- The maximization of the likelihood must be computed numerically by employing optimization algorithms.
- Even if the data seems not Gaussian, using the Gaussian assumption for the estimation is still sensible since this could still be interpreted as a measure of fit of the model.
- We will now see two approaches to choose p and q to perform the estimation procedure that we just discussed.

Model Selection of ARMA Models

The Box-Jenkins methodology is possibly the most used approach to model selection among ARMA models. It starts with the visual inspection of the ACF and PACF as in the following table:

Process	ACF	PACF
White Noise	zeros	zeros
AR(p)	decays exponentially	non-zero until lag p and zeros after
MA(q)	non-zero until laq q, zeros after	decays exponentially
ARMA(p, q)	decays exponentially	decays exponentially

After selecting candidate models (p and q), estimation of the model is performed and residual analysis of $\varepsilon = (\varepsilon_t)_{t \in \mathcal{T}}$ is performed. If some structure is detected, a new model is selected and the analysis repeated. Parsimony is desired (Occam's Razor) and we should always start with the simplest model (less parameters) after visual inspection.

Model Selection of ARMA Models

- Another approach to model selection is to minimize a information criterion for different values of p and q.
- Assume we have T observations of the time series X and we have estimated an ARMA(p, q) using the approaches we have presented above.
- This estimation gives us an estimator for the variance of the residuals $\varepsilon = (\varepsilon_t)_{t\in\mathcal{T}}$, denoted here by $\hat{\sigma}^2_{p,q}$.
- We then consider the following criteria:

Akaike information criterion - $\text{AIC}(p, q) = \log \hat{\sigma}^2_{p,q} + (p + q)\frac{2}{T}$,

Bayesian information criterion - $\text{BIC}(p, q) = \log \hat{\sigma}^2_{p,q} + (p + q)\frac{\log T}{T}$,

Hannan-Quinn information criterion - $\text{HQC}(p, q) = \log \hat{\sigma}^2_{p,q} + (p + q)\frac{2 \log\log T}{T}$

Model Selection of ARMA Models

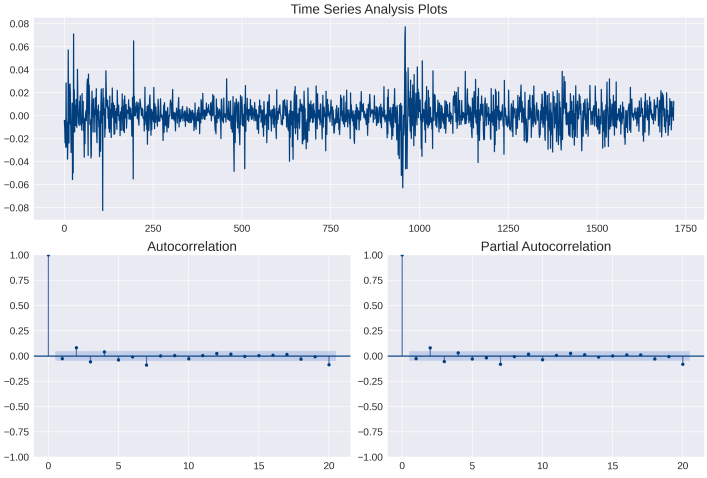
- All these information criteria are of the form

$$\underbrace{\log \hat{\sigma}^2_{p,q}}_{\text{goodness of fit}} + \underbrace{(p+q)\frac{C(T)}{T}}_{\text{penalty term}}$$

- It is a trade-off of goodness of fit and complexity of the model, and $C(T)$ must be a non-decreasing function of T.
- One could easily show that AIC $<$ HCQ $<$ BIC, for T mildly large, AIC gives the largest models (highest order $p+q$).
- AIC is the most used criterion, although it has a tendency to choose models that overfit.
- These information criteria approaches are "alternatives" for cross-validation, when we do not have (or do not want to separate a) validation set to select a model.

SPX - Log Difference

Let us analyze the $X_t = \Delta \log(\text{SPX}_t)$



Estimation of ARMA(1, 1)

Dep. Variable:	y	**No. Observations:**	1717
Model:	ARIMA(1, 0, 1)	**Log Likelihood**	5137.397
Date:	Sun, 24 Sep 2023	**AIC**	-10266.795
Time:	20:11:22	**BIC**	-10245.001
Sample:	0	**HQIC**	-10258.731
	- 1717		
Covariance Type:	opg		

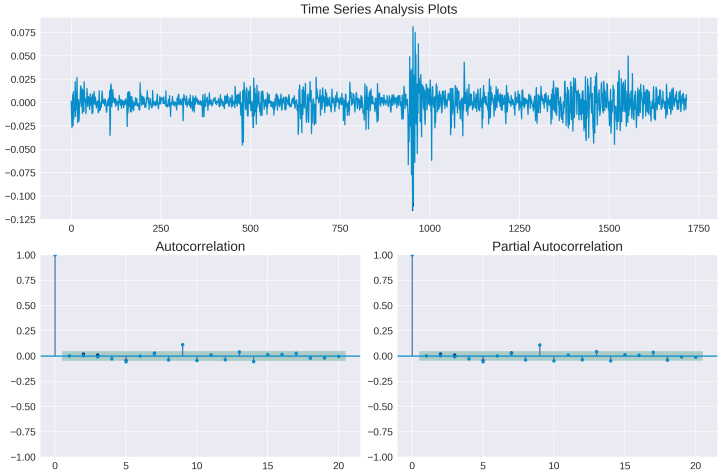
	coef	**std err**	**z**	**P> |z|**	**[0.025**	**0.975]**
const	0.0005	0.000	1.680	0.093	-7.93e-05	0.001
ar.L1	-0.6757	0.025	-27.322	0.000	-0.724	-0.627
ma.L1	0.5245	0.029	18.347	0.000	0.468	0.580
sigma2	0.0001	2.05e-06	71.742	0.000	0.000	0.000

Estimation of ARMA(2, 2)

Dep. Variable:	y	**No. Observations:**	1717
Model:	ARIMA(2, 0, 2)	**Log Likelihood**	5137.727
Date:	Sun, 24 Sep 2023	**AIC**	-10263.453
Time:	20:12:18	**BIC**	-10230.763
Sample:	0	**HQIC**	-10251.357
	- 1717		
Covariance Type:	opg		

	coef	**std err**	**z**	**P> |z|**	**[0.025**	**0.975]**
const	0.0005	0.000	1.418	0.156	-0.000	0.001
ar.L1	0.0908	0.227	0.400	0.689	-0.354	0.536
ar.L2	0.4080	0.159	2.572	0.010	0.097	0.719
ma.L1	-0.2426	0.232	-1.048	0.295	-0.696	0.211
ma.L2	-0.2732	0.130	-2.097	0.036	-0.529	-0.018
sigma2	0.0001	2.07e-06	71.234	0.000	0.000	0.000

ARMA(1, 1) vs. ARMA(2, 2)

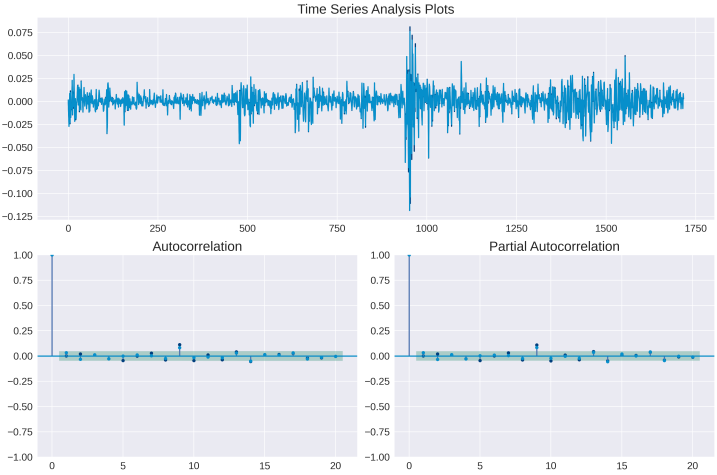


Estimation of ARMA(5, 5)

Dep. Variable:	y	**No. Observations:**	1717
Model:	ARIMA(5, 0, 5)	**Log Likelihood**	5145.014
Date:	Sun, 24 Sep 2023	**AIC**	-10266.028
Time:	20:13:03	**BIC**	-10200.648
Sample:	0	**HQIC**	-10241.836
	- 1717		
Covariance Type:	opg		

	coef	std err	z	P> |z|	[0.025	0.975]
const	0.0005	0.000	1.527	0.127	-0.000	0.001
ar.L1	-0.7153	0.415	-1.723	0.085	-1.529	0.098
ar.L2	0.0031	0.509	0.006	0.995	-0.994	1.001
ar.L3	-0.2347	0.384	-0.611	0.541	-0.987	0.518
ar.L4	-0.3032	0.429	-0.706	0.480	-1.145	0.539
ar.L5	0.0377	0.253	0.149	0.882	-0.459	0.534
ma.L1	0.5363	0.419	1.281	0.200	-0.284	1.357
ma.L2	0.0255	0.443	0.058	0.954	-0.844	0.895
ma.L3	0.2679	0.351	0.763	0.446	-0.420	0.956
ma.L4	0.2604	0.414	0.630	0.529	-0.550	1.071
ma.L5	-0.0961	0.216	-0.445	0.656	-0.519	0.327
sigma2	0.0001	2.26e-06	64.549	0.000	0.000	0.000

ARMA(1, 1) vs. ARMA(5, 5)



Model Selection

- $\text{AIC}(1,1) = -10266.795$
- $\text{AIC}(2,2) = -10263.453$
- $\text{AIC}(5,5) = -10266.028$
- Among those models, the minimum AIC is attained for the ARMA$(1,1)$
- Other choices of p and q could be considered. In fact, using the method `arma_order_select_ic` of `statsmodels` shows that ARMA$(2,0)$ is even better than ARMA$(1,1)$ considering the AIC metric ($\text{AIC}(2,0) = -10267.615$).

Estimation of ARMA(2, 0)

Dep. Variable:	y	**No. Observations:**	1717
Model:	ARIMA(2, 0, 0)	**Log Likelihood**	5137.808
Date:	Sun, 24 Sep 2023	**AIC**	-10267.615
Time:	20:14:50	**BIC**	-10245.822
Sample:	0	**HQIC**	-10259.551
	- 1717		
Covariance Type:	opg		

	coef	std err	z	P> |z|	[0.025	0.975]
const	0.0005	0.000	1.543	0.123	-0.000	0.001
ar.L1	-0.1528	0.010	-15.081	0.000	-0.173	-0.133
ar.L2	0.1045	0.010	10.562	0.000	0.085	0.124
sigma2	0.0001	2.01e-06	73.176	0.000	0.000	0.000

ARMA(1, 1) vs. ARMA(2, 0)

Do Question 4 on the Jupyter Notebook `Time Series - ARMA, GARCH and VaR`.

Variations on ARMA models

ARIMA

- Usually, in financial and economics time series, we observe a non-stationarity behavior by noticing that the mean function is not constant.
- We have often differentiate the time series to remove this behavior and then create a stationary process that could be modeled as ARMA.
- This is called ARI(ntegrated)MA model. In the case where we have to differentiate the series d times to arrive in a stationary process, we write ARIMA(p, d, q).
- Another approach is to detrend the time series by considering a specific formula for the mean function.
- Commonly, one considers linear by parts or polynomials:

$$X_t = \delta + \delta_1\, t + \cdots + \delta_k\, t^k + \text{ARMA}.$$

SARIMA

- In economics and climate time series, it is commonly observed a seasonality effect, which usually creates non-stationarity.
- One well-known way to remove seasonality of time series is to take moving-averages, creating much smoother time series and evening out the seasonal movements.
- For instance, one could consider the (causal) moving average of the process X using n past observations:

$$Y_t = \frac{1}{n} \sum_{i=1}^{n} X_{t-i+1}$$

- Another way to model seasonality is to consider the S(easonal)ARIMA models that says that X_t is impacted by the usual ARMA(p, q) model, but additionally impacted by another ARMA(p_s, q_s) with a delay of s time steps (the seasonality, e.g. set $s = 12$ for year seasonality in monthly data), i.e. the additional lags are offset by the frequency of the seasonality.

SARIMAX

- Although we will not explore in this course, an important generalization of the SARIMA is the SARIMA(with e)X(ogenous factors) model.
- It could be understood as (time series) regression with SARIMA errors:

$$Y_t = \beta X_t + \varepsilon_t,$$

  where $\varepsilon = (\varepsilon_t)_{t \in \mathcal{T}}$ follows a SARIMA model (or simpler).
- If after regression Y over X, the error ε is not stationary and differentiating is necessary, then we should differentiate the regression before fitting the model.
- "Estimation of a model with non-stationary errors is not consistent and can lead to 'spurious regression'."[1]

[1] Rob J Hyndman, from `https://robjhyndman.com/hyndsight/arimax/`

Stochastic Volatility - GARCH Models

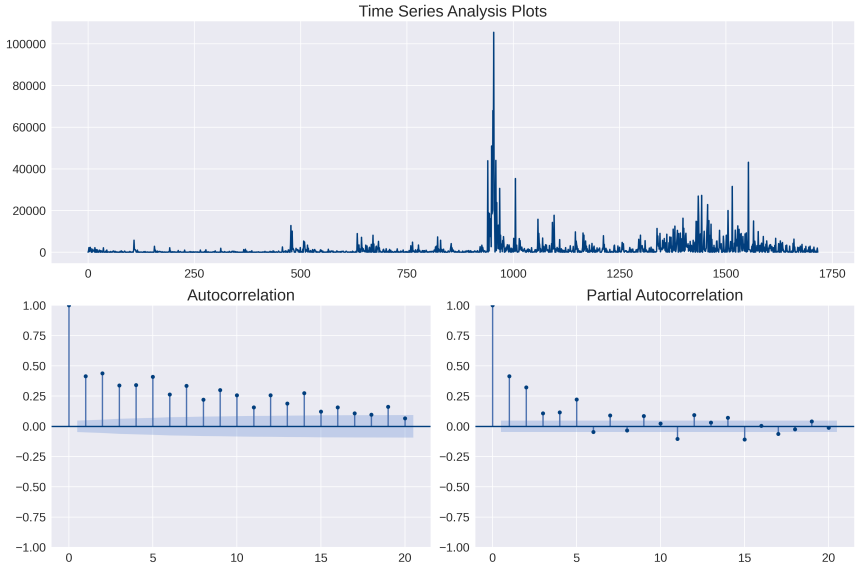
Volatility

- Volatility modeling is very important for a plethora of reasons, an important one being the protagonism of the derivatives market and the fact that the main driver of the price of derivatives is the volatility of the underlying asset.
- However, we will not consider financial derivatives in this course. Another very important financial application that volatility plays a major role is risk management.
- In particular, we will consider the Value-at-Risk (VaR) of a financial portfolio. This risk measure became essential in banks and investment funds due to the regulations of the Basel accords.
- There are several reasons to consider stochastic volatility, as we have seen previously. Engle's consideration to create the ARCH model was that the conditional forecast error variance is no longer constant. We will consider directly the generalization of the ARCH model, the so-called GARCH models.

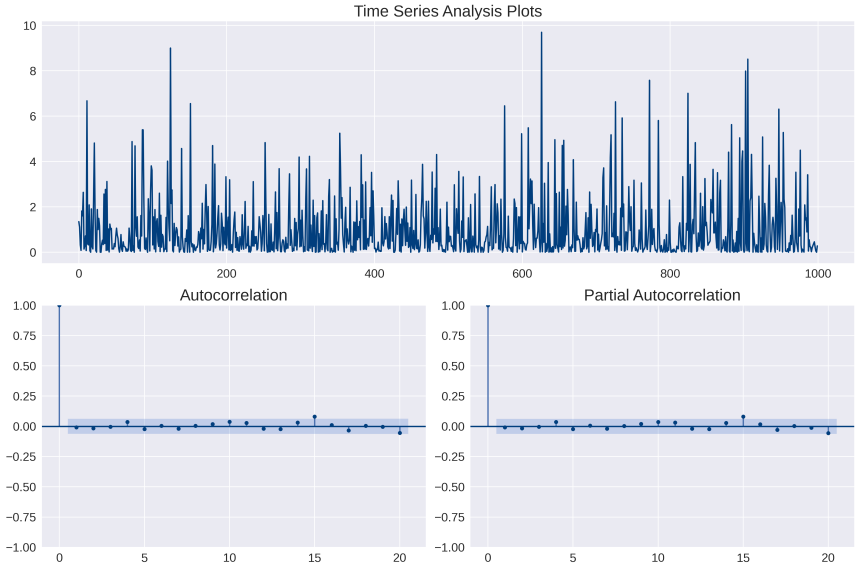
Volatility

- In order to derive some evidence of heteroskedasticity, we start by considering the residuals of a general ARMA model applied to X, denoted here by ε.
- Since we are interested in the (conditional) variance, we will consider the squared residuals, $(\varepsilon_t^2)_{t\in\mathcal{T}}$.
- We then compute the ACF and PACF of ε^2, and after visual inspection we check if it is consistent with a white noise model.
- If it were, it would be more evidence of iid residuals and hence, Homoscedasticity.

SPX - Difference Squared



White Noise Squared



GARCH

- We start with an AR(1) model: $X_t = a_0 + a_1 X_{t-1} + \varepsilon_t$, with $\varepsilon = (\varepsilon_t)_{t\in\mathcal{T}}$; more complex models for X could be considered.
- The GARCH(p, q) model assumes $\varepsilon_t = \sigma_t \eta_t$, where $\eta = (\eta_t)_{t\in\mathcal{T}}$ is an iid $N(0, 1)$ process, independent of ε, and

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2.$$

- We assume $\alpha_0, \alpha_i, \beta_j \geq 0$ in order to guarantee $\sigma_t^2 \geq 0$.
- To guarantee stationarity it is necessary that $\sum_{i=1}^{p} \alpha_i + \sum_{j=1}^{q} \beta_j < 1$.
- The simplest GARCH is the (1,1):

$$\sigma_t^2 = \alpha_0 + \alpha \varepsilon_{t-1}^2 + \beta \sigma_{t-1}^2.$$

EWMA

- Exponentially Weighted Moving Averages (EWMA) is a very pratical model to predict volatility.
- It could be seen as a GARCH(1,1) with $\alpha_0 = 0$ and $\alpha + \beta = 1$:

$$\sigma_t^2 = (1 - \lambda)\varepsilon_{t-1}^2 + \lambda\sigma_{t-1}^2.$$

- One can easily show that

$$\sigma_t^2 = (1 - \lambda)\sum_{k=0}^{+\infty} \lambda^k \varepsilon_{t-k-1}^2.$$

- Usually, one chooses $\lambda = 0.94$ for daily data, or similar values depending of the frequency of the data. These values for λ are usually calibrated to forecast volatility.

GARCH Generalizations

- An important generalization is the Threshold GARCH model. It is observed in real data that negative shocks in the stock price impacts more the volatility than positive shocks.
- For instance, the asymmetric GARCH(1, 1):

$$\sigma_t^2 = \alpha_0 + (\alpha_1 + \gamma\, 1_{\{\varepsilon_{t-1}<0\}})\varepsilon_{t-1}^2 + \beta_1\sigma_{t-1}^2.$$

- Another important generalization is so-called E(xponential)GARCH. The EGARCH(1, 1) is:

$$h_t = \alpha_0 + \beta h_{t-1} + \alpha|\eta_{t-1}| + \delta\eta_{t-1},$$

with $\varepsilon_t = e^{h_t/2}\eta_t$ (i.e. $h_t = \log\sigma_t^2 = 2\log\sigma_t$).

GARCH Estimation

- The most popular estimation of GARCH models is the maximum likelihood method.

- Let us consider the model specified previously:

$$\begin{cases} X_t &= a_0 + a_1 X_{t-1} + \varepsilon_t, \\ \varepsilon_t &= \sigma_t \eta_t, \quad \eta_t \overset{iid}{\sim} N(0,1), \\ \sigma_t^2 &= \alpha_0 + \sum_{i=1}^{p} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2. \end{cases}$$

- Under this model, one can easily see that, conditionally on the past, $\{X_{t-1}, \ldots\}$, X_t is Gaussian with

$$\begin{cases} \text{mean} &= a_0 + a_1 X_{t-1}, \\ \text{variance} &= \alpha_0 + \sum_{i=1}^{p} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=1}^{q} \beta_j \sigma_{t-j}^2, \end{cases}$$

where $\varepsilon_s = X_s - a_0 - a_1 X_{s-1}$.

GARCH Estimation

- Therefore, by conditioning, we can write the joint density of $(X_1, \ldots, X_T)$ as the product of the Gaussians from the previous slide:

$$p(x_1, \ldots, x_T) = p(x_1, \ldots, x_s) \prod_{t=s}^{T} p(x_t \mid x_{t-1}, \ldots),$$

  where $s > p$, since we need at least p observations to evaluate the volatility process.

- Writing in terms of log-likelihood and considering $p(x_1, \ldots, x_s)$ as a fixed constant level, we want to maximize

$$\log L(a_0, a_1, \boldsymbol{\alpha}, \boldsymbol{\beta} \mid x_1, \ldots, x_T) = -\frac{T}{2} \log 2\pi - \frac{1}{2} \sum_{t=s}^{T} \log \sigma_t^2 - \frac{1}{2} \sum_{t=s}^{T} \log \frac{\varepsilon_t^2}{\sigma_t^2}$$

GARCH Estimation

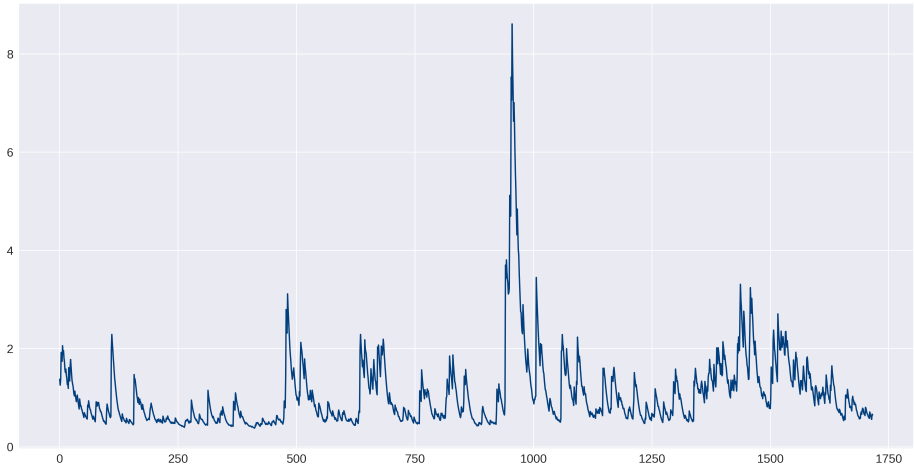
- One could consider other distribution, with fat-tails, instead of Gaussian.
- It is also possible to estimate the model of X separately from the model for the volatility and then, estimate a GARCH model for the residuals.
- The ML estimation requires numerical optimization method and because of that, the initial value for the parameters are very important. This could be done by the method of moments and the sample estimation of the ACF for the residual square.

GARCH Estimation

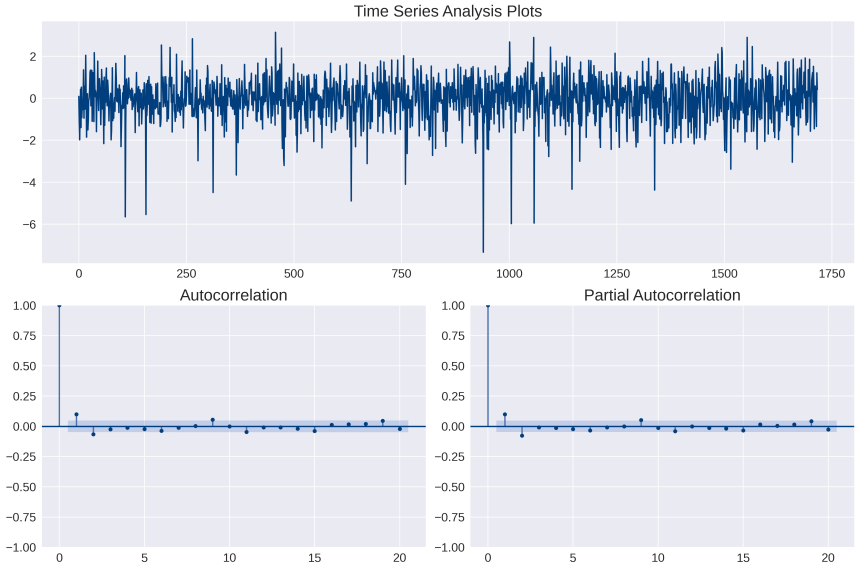
Dep. Variable:	y	**R-squared:**	0.000
Mean Model:	Constant Mean	**Adj. R-squared:**	0.000
Vol Model:	GJR-GARCH	**Log-Likelihood:**	-2210.49
Distribution:	Standardized Student's t	**AIC:**	4432.99
Method:	Maximum Likelihood	**BIC:**	4465.68
		No. Observations:	1717
Date:	Sun, Sep 24 2023	**Df Residuals:**	1716
Time:	20:21:07	**Df Model:**	1

	coef	std err	t	P> |t|	95.0% Conf. Int.
mu	0.0364	1.554e-02	2.345	1.902e-02	[5.986e-03,6.691e-02]
	coef	**std err**	**t**	**P> |t|**	**95.0% Conf. Int.**
omega	0.0230	7.268e-03	3.166	1.544e-03	[8.768e-03,3.726e-02]
alpha[1]	0.0522	2.395e-02	2.177	2.946e-02	[5.204e-03,9.910e-02]
gamma[1]	0.2390	5.834e-02	4.097	4.190e-05	[0.125, 0.353]
beta[1]	0.8204	3.290e-02	24.937	2.984e-137	[0.756, 0.885]
	coef	**std err**	**t**	**P> |t|**	**95.0% Conf. Int.**
nu	5.6029	0.788	7.107	1.183e-12	[4.058, 7.148]

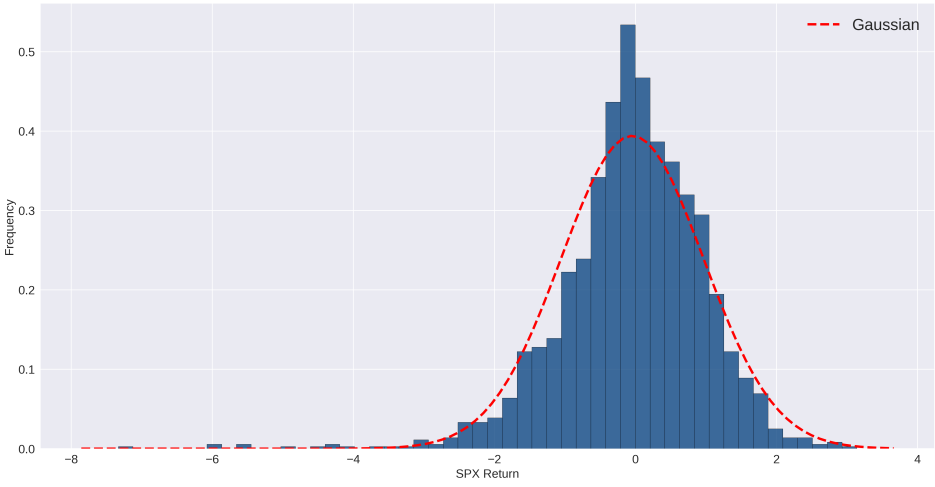
Conditional Volatility σ_t



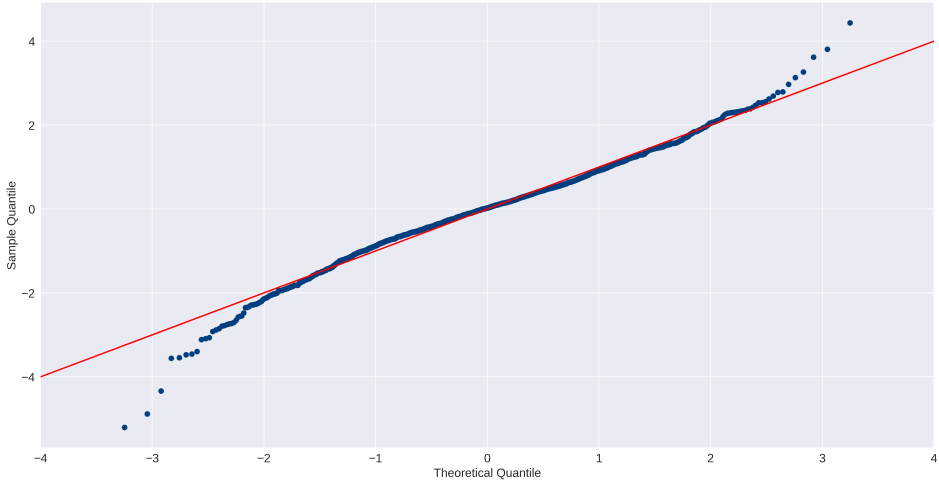
GARCH Standard Residual η_t



GARCH Standard Residual η_t



GARCH Standard Residual η_t



Do Question 5 on the Jupyter Notebook `Time Series - ARMA, GARCH and VaR`.

Forecast of ARMA and GARCH models

- We will consider that we are at time t and want to predict at time $t+h$ by minimizing the mean-squared error of the prediction and the observed value.
- This implies that the prediction, under the assumed model, must be $\mathbb{E}[X_{t+h} \mid X_t, X_{t-1}, \ldots]$, which we will denote by $\mathbb{E}_t[X_{t+h}]$.
- For instance, under an AR(1) model and for $h = 1$, we find

$$\mathbb{E}_t[X_{t+1}] = \mathbb{E}_t[a_0 + a_1 X_t + \varepsilon_{t+1}] = a_0 + a_1 X_t.$$

- In general, for any h, but still under AR(1), we find

$$\mathbb{E}_t[X_{t+h}] = a_0 \sum_{k=0}^{h-1} a_1^k + a_1^h X_t.$$

- Under the stationary case, this converges, as $h \to +\infty$, to $a_0/(1-a_1)$, which is the unconditional mean of X_t.

Forecast of ARMA and GARCH models

- In practice, we need to substitute the model parameters by their estimate.
- For the MA models, the computation is similar. For instance, for a MA(2), we find

$$\mathbb{E}_t[X_t] = \mathbb{E}_t[a_0 + a_1\varepsilon_t + a_2\varepsilon_{t-1} + \varepsilon_{t_1}] = a_0 + a_1\varepsilon_t + a_2\varepsilon_{t-1}.$$

- However, in this case, we need to estimate the model parameters and the residuals ε.
- In general, for an ARMA(p, q), similar computations could be performed.

Forecast of ARMA and GARCH models

- Under GARCH models, if we assume (as we did) that the standardized residuals, $\eta_t = \varepsilon_t/\sigma_t$ are iid with mean 0, then the GARCH assumption will not provide a different forecast under the MSE framework.
- More complicated GARCH assumptions would allow for more flexible estimation for X.
- Nonetheless, GARCH is very helpful to predict volatility. For instance, under the GARCH(1,1), we find

$$\mathbb{E}_t[\sigma_{t+1}^2] = \mathbb{E}_t[\alpha_0 + \alpha\varepsilon_t^2 + \beta\sigma_t^2] = \alpha_0 + \alpha\varepsilon_t^2 + \beta\sigma_t^2.$$

- Similar computations could be performed for the generalizations of the GARCH model we have considered. For prediction for longer time horizons, some computations could become more complicated.

Forecast of ARMA and GARCH models

- For the two-step ahead prediction, we find:

$$\mathbb{E}_t[\sigma^2_{t+2}] = \alpha_0 + \alpha\mathbb{E}_t[\varepsilon^2_{t+1}] + \beta\mathbb{E}_t[\sigma^2_{t+1}].$$

- We have already computed $\mathbb{E}_t[\sigma^2_{t+1}]$. We need to compute:

$$\mathbb{E}_t[\varepsilon^2_{t+1}] = \mathbb{E}_t[\eta^2_{t+1}\sigma^2_{t+1}] \overset{\text{indep.}}{=} \mathbb{E}_t[\eta^2_{t+1}]\mathbb{E}_t[\sigma^2_{t+1}] \overset{\mathbb{E}_t[\eta^2_{t+1}]=1}{=} \mathbb{E}_t[\sigma^2_{t+1}].$$

- Therefore

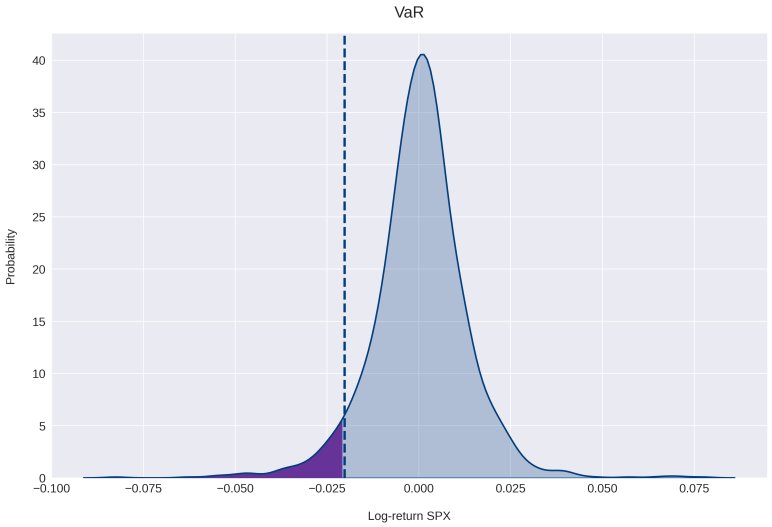
$$\mathbb{E}_t[\sigma^2_{t+2}] = \alpha_0 + (\alpha+\beta)\mathbb{E}_t[\sigma^2_{t+1}].$$

- For general h, one finds

$$\mathbb{E}_t[\sigma^2_{t+h}] = \alpha_0 \sum_{k=0}^{h-1}(\alpha+\beta)^k + (\alpha+\beta)^{h-1}(\alpha\varepsilon^2_t + \beta\sigma^2_t).$$

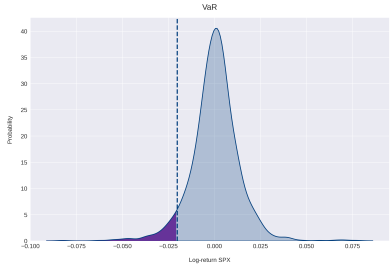
Value at Risk

Value at Risk



Value at Risk

- In words: the (daily) Value at Risk (VaR) at level α of (daily) returns X is the $(1-\alpha)$ of the losses $Y = -X$.
- The mathematical definition is $\text{VaR}^{\alpha} = \sup\{y \in \mathbb{R} \ ; \ \mathbb{P}(Y \leq y) > \alpha\}$



Value at Risk - AR-GARCH

- Suppose we have observed returns up to time t, $x_1, \ldots, x_t$, and we want to estimate the one period VaR:

$$\text{VaR}^{\alpha}_{t,t+1} = \sup\{y \in \mathbb{R} \ ; \ \mathbb{P}(-X_{t+1} \leq y \mid x_1, \ldots, x_t) > \alpha\},$$

  where X is the return of the portfolio or asset in one period.

- Under the AR(1) assumption, $X_{t+1} = a_0 + a_1 X_t + \varepsilon_{t+1}$, we find

$$\text{VaR}^{\alpha}_{t,t+1} = \sup\{y \in \mathbb{R} \ ; \ \mathbb{P}(-\varepsilon_{t+1} \leq y - a_0 - a_1 x_t \mid x_1, \ldots, x_t) > \alpha\}$$

- If we additionally assume GARCH model, we find

$$\text{VaR}^{\alpha}_{t,t+1} = \sup\left\{y \in \mathbb{R} \ ; \ \mathbb{P}\left(-\eta_{t+1} \leq \frac{y - a_0 - a_1 x_t}{\sigma_{t+1}} \mid x_1, \ldots, x_t\right) > \alpha\right\},$$

Value at Risk - Estimation

- We then estimate $\text{VaR}^{\alpha}_{t,t+1}$ as

$$\widehat{\text{VaR}^{\alpha}_{t,t+1}} = \sup\left\{ y \in \mathbb{R} \ ; \ \mathbb{P}\left(-\eta_{t+1} \leq \frac{y - a_0 - a_1 x_t}{\hat{\sigma}_{t,t+1}} \right) > \alpha \right\},$$

  where $\hat{\sigma}_{t,t+1}$ is the prediction of σ_{t+1} given $x_1, \ldots, x_t$.

- This implies that

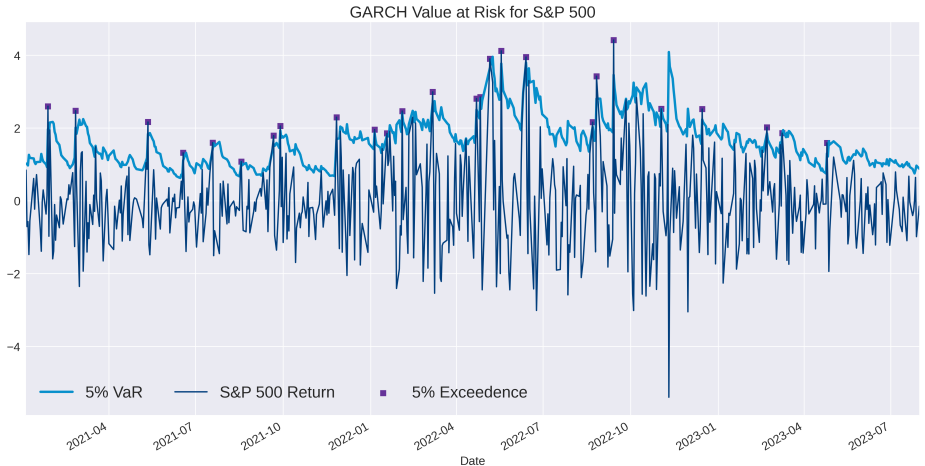
$$\widehat{\text{VaR}^{\alpha}}_{t,t+1} = -a_0 - a_1 x_t - \hat{\sigma}_{t,t+1} q^{\eta}_{\alpha},$$

  where q^{η}_{α} is the α-quantile of the standardized residual, η.

- With the model estimated with data up to the end of 2021, we forecast the model from that day onward. This gives us the VaR estimation above, where the α-quantile is computed with closed form using the chosen distribution of η, in our case, Student-t.

Value at Risk - Estimation

The estimated exceedance of the VaR 5% is around 4.43%. Do Question 6 on the Jupyter Notebook `Time Series - ARMA, GARCH and VaR`.



Time Series Prediction

Time Series Analysis

- Compute the risk of a financial portfolio. For this, one needs to understand the future distribution of the multivariate time series with stochastic volatility (hence, a statistical inference problem).
- Predict the interest rate curve. For this, one could use dimensionality reduction, classical prediction using classical time series models (e.g. VAR) and Machine Learning techniques (e.g. LSTM).
- Factor investing, the Fama–French factors and their capacity to explain the cross-sectional of returns. In this situation, one needs Time Series Regression techniques.

Data: US Treasury Zero-Coupon Yield Curve

Prediction of Time Series

- Prediction of financial time series is a very challenging subject because of the nature of financial markets.
- Usually, for stock prices daily data, the current price is the best predictor for future prices.
- In order to provide a meaningful example with more complex data structure we will consider US Treasury Zero-Coupon Yield Curve.
- From the website: "These yield curves are an off-the-run Treasury yield curve based on a large set of outstanding Treasury notes and bonds, and are based on a continuous compounding convention. Values are daily estimates of the yield curve from 1961 for the entire maturity range spanned by outstanding Treasury securities."

US Treasury Zero-Coupon Yield Curve

- Let's unravel the meaning of yield curves.
- A zero-coupon bond with maturity T is a financial contract that pays \$1 at time T.
- The US Treasury zero-coupon bond is issued (and paid at maturity) by the US Government.
- There are two main risks associated with this contract: the interest rate and the solvency of the issuer.

US Treasury Zero-Coupon Yield Curve

- Let the price of this bond at time $t < T$ be denoted by $P(t, T)$. It is clear that: (i) $P(T, T) = 1$ and (ii) if the interest rate is constant and deterministic equal r, then $P(t, T) = e^{-r(T-t)}$.

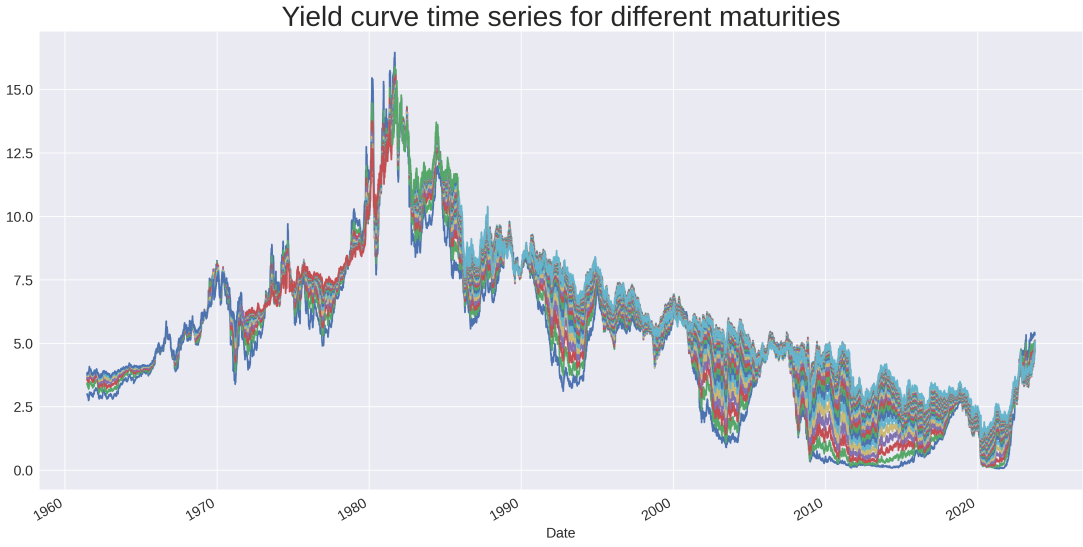
- The yield-to-maturity, or just yield, is the implied constant rate that calibrates the price $P(t, T)$:

$$P(t, T) = e^{-y(t,T)(T-t)}$$

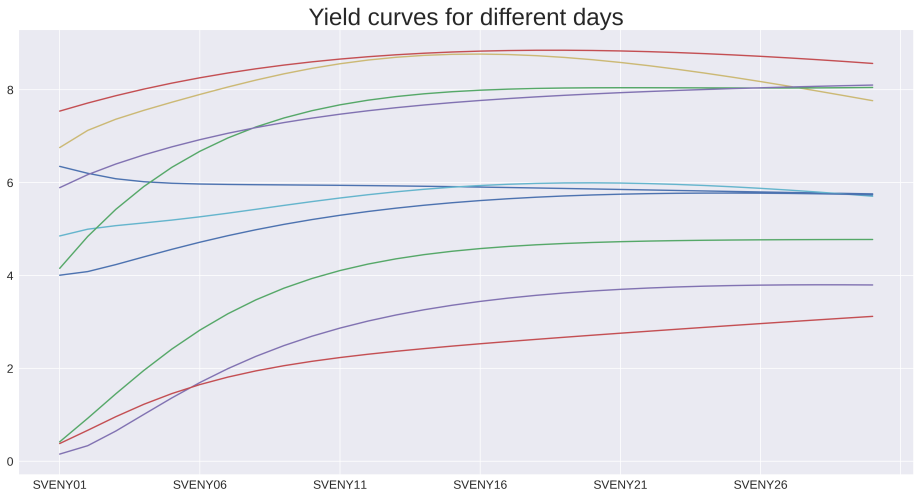
- This means that

$$y(t, T) = -\frac{\log P(t, T)}{T - t}.$$

US Treasury Zero-Coupon Yield Curve



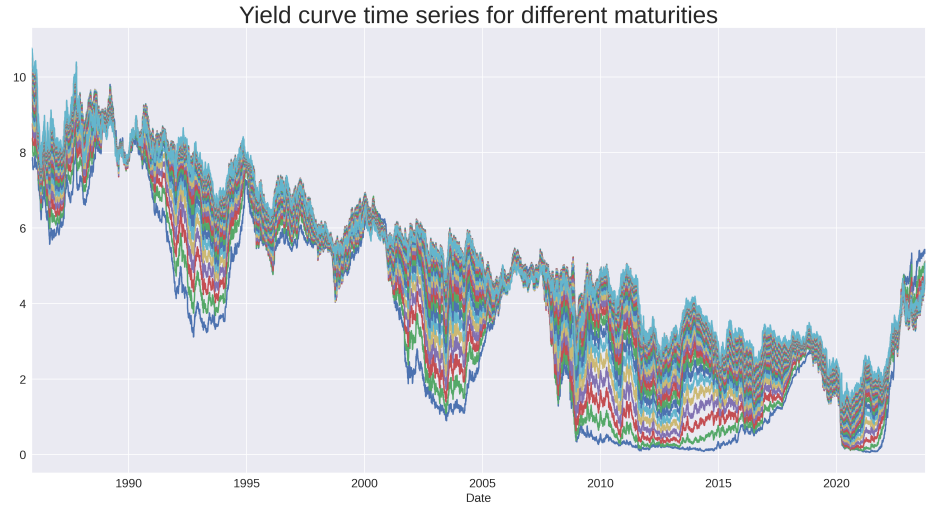
US Treasury Zero-Coupon Yield Curve



US Treasury Zero-Coupon Yield Curve

- `Quandl` provides yield curves from as early as 1961. However, only the first few maturities are available for the oldest data points.
- Each yield curve have maturities from 1 to up to 30 years.
- We have then decided to remove all dates that did not have the full yield curve available (all the 30 maturities).
- Hence, we are going to use the yield curves from the end of 1985 to today, a total of 9452 data points.

US Treasury Zero-Coupon Yield Curve

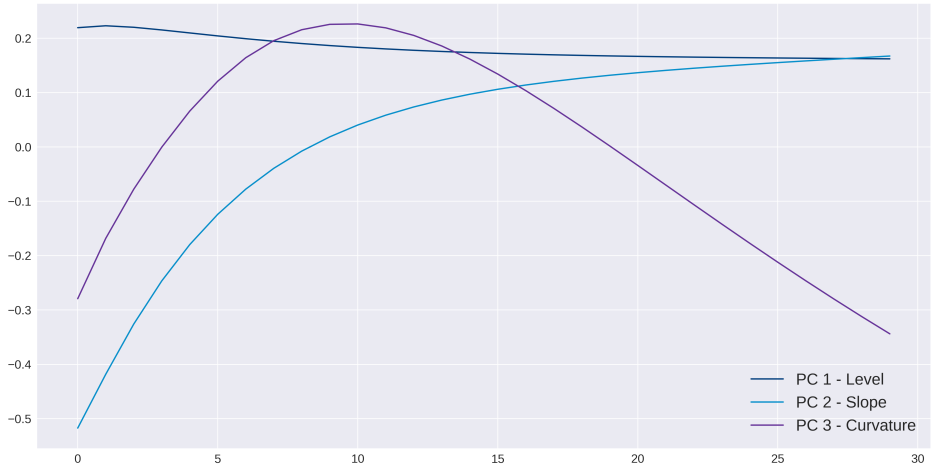


Principal Components Analysis

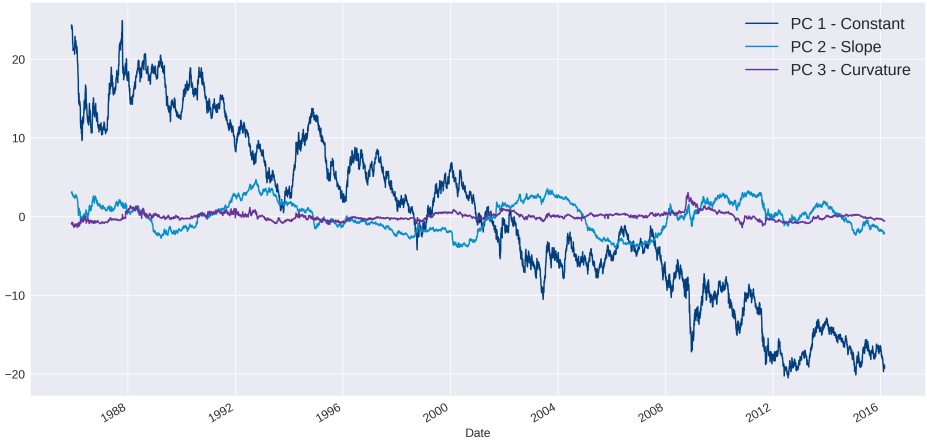
PCA for the Yield Curves

- From the sample above, we see that the curves vary in level, slope and curvature.
- If we hope to predict these time series using techniques similar to the ones we used on the last lecture, we will surely run into problems because the time series for each maturity are highly correlated.
- We then run an PCA with $n = 10$ components with standardizing the data.
- The idea is to choose the first factors that explain most of the variance. More than just dimensionality reduction, these factors will be orthogonal and better used for prediction.
- The idea can be used in a number of other cases as implied volatility surfaces, commodities futures, VIX futures, and others.

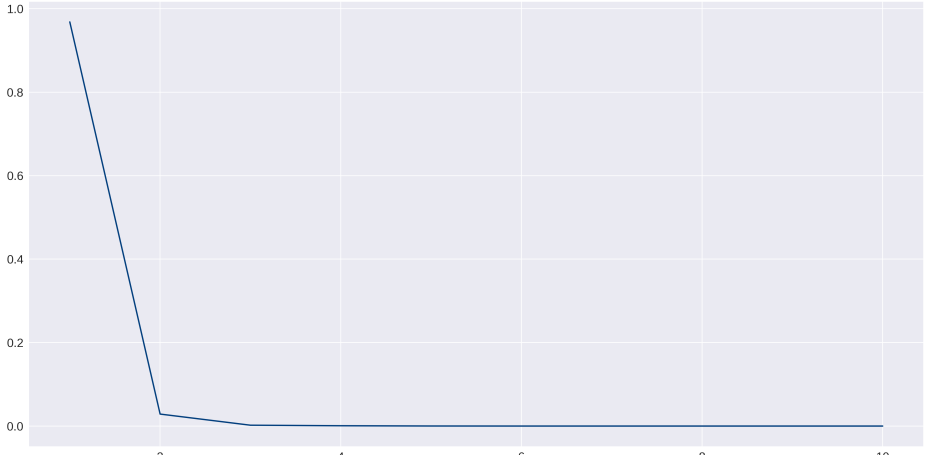
PCA for the Yield Curves



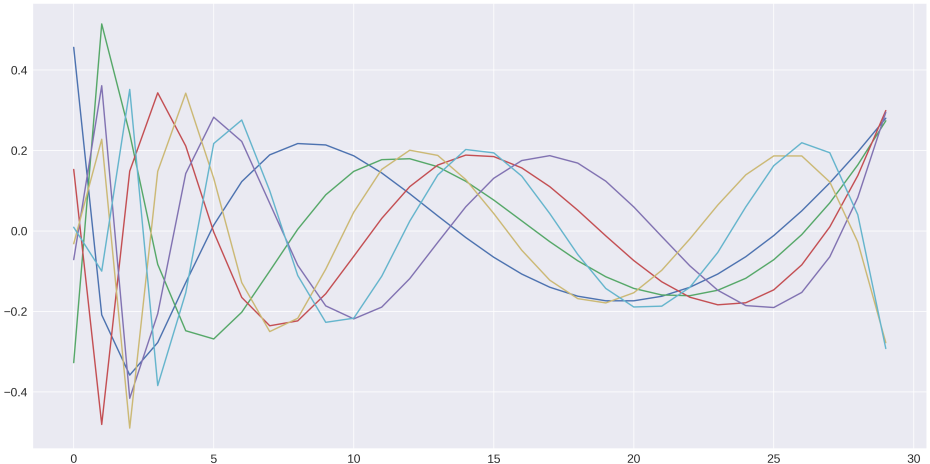
PCA for the Yield Curves



Explained Variance



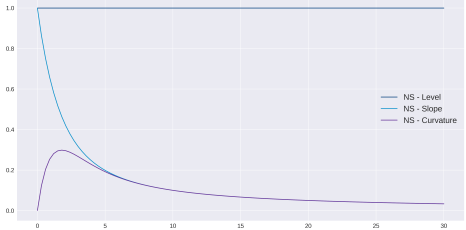
Other Components



PCA Factors

- As we see, this is consistent with the famous yield parameterization by Nelson-siegel[2]:

$$y(\tau) = \beta_0 + \beta_1 \frac{1 - e^{-\lambda\tau}}{\lambda\tau} + \beta_2 \left(\frac{1 - e^{-\lambda\tau}}{\lambda\tau} - e^{-\lambda\tau} \right)$$



- The PCA factors are interpretable and clearly related to the Nelson-Siegel model. Nelson and Siegel proposed their factors by financial-economical reasoning.

[2] Parsimonious Modeling of Yield Curves (1987) `https://www.jstor.org/stable/2352957`

Multidimensional Time Series Analysis

- After performing the dimensionality reduction by using the Nelson-Siegel factor, we will then study the dynamics and prediction of the 3 time series (Level, Slope and Curvature).
- We will first consider the Vector Autoregressive (VAR).
- We will then introduce and use Long Short-Term Memory (LSTM) deep learning architecture.

Multivariate Time Series Modeling
Vector Autoregressive

Vector Autoregressive

- Possibly, the most used and useful model from classical econometrics to predict multiple time series is the vector autoregressive models.
- Assume we want to model k time series, denoted by $\boldsymbol{X}_t = (X_t^{(1)}, \ldots, X_t^{(k)})$.
- The VAR(p) model for $\boldsymbol{X}$ is given by

$$\boldsymbol{X}_t = \boldsymbol{A}_0 + A_1 \boldsymbol{X}_{t-1} + \cdots + A_p \boldsymbol{Y}_{t-p} + \varepsilon_t,$$

  where $\boldsymbol{A}_0$ is a vector in $\mathbb{R}^k$, $A_1, \ldots, A_p$ are $k \times k$ matrices and $(\varepsilon_t)_{t \in \mathcal{T}}$ is a k-dimensional white noise process.
- In fact, we say $(\varepsilon_t)_{t \in \mathcal{T}}$ is a k-dimensional white noise process if $\mathbb{E}[\varepsilon_t] = 0$, $\mathbb{E}[\varepsilon_t \varepsilon_{t+s}^T] = 0$ and $\mathbb{E}[\varepsilon_t \varepsilon_t^T] = \Sigma$, where $s \neq 0$ and Σ is a covariance matrix.

Vector Autoregressive

- A VAR(p) process can always be transformed in a VAR(1) process by increasing the dimension of $\boldsymbol{X}$ by considering the last $p-1$ lags of $\boldsymbol{X}$.
- Hence, we will focus on the VAR(1) model:

$$\boldsymbol{X}_t = \boldsymbol{A} + B\boldsymbol{X}_{t-1} + \varepsilon_t$$

- Explicitly, the case $k = 2$ can be written as

$$\begin{cases} X_t^{(1)} = A_1 + b_{11}X_{t-1}^{(1)} + b_{12}X_{t-1}^{(2)} + \varepsilon_t^{(1)} \\ X_t^{(2)} = A_2 + b_{21}X_{t-1}^{(1)} + b_{22}X_{t-1}^{(2)} + \varepsilon_t^{(2)} \end{cases}$$

- The VAR(1) is stationary if the absolute value of eigenvalues of B are less than 1.

Vector Autoregressive - Model Selection

- ACF and PACF can be readily generalized to the multidimensional case by considering the cross-correlation between dimension of $\boldsymbol{X}$.
- AIC, BIC and HQIC are also similarly defined as in the one-dimensional case.
- Furthermore, estimation of VAR(p) can be performed by straightforward generalization of the estimation of AR(p), as using OLS.
- Forecasting follows a similar pattern:

$$\mathbb{E}_t[\boldsymbol{X}_{t+h}] = \sum_{k=0}^{h-1} A_1^k \boldsymbol{A}_0 + A_1^h \boldsymbol{X}_t.$$

- In practice, it is useful to compute prediction of deviations of the VAR, $\tilde{\boldsymbol{X}}_t = \boldsymbol{X}_t - \boldsymbol{\mu}$, where $\boldsymbol{\mu}$ is the mean of $\boldsymbol{X}$, which can be computed in closed form.

Vector Autoregressive - Model Selection and Forecasting

- ACF and PACF can be readily generalized to the multidimensional case by considering the cross-correlation between dimension of $\boldsymbol{X}$.
- AIC, BIC and HQIC are also similarly defined as in the one-dimensional case.
- Furthermore, estimation of VAR(p) can be performed by straightforward generalization of the estimation of AR(p), as using OLS.
- Forecasting follows a similar pattern:

$$\mathbb{E}_t[\boldsymbol{X}_{t+h}] = \sum_{k=0}^{h-1} A_1^k \boldsymbol{A}_0 + A_1^h \boldsymbol{X}_t.$$

- In practice, it is useful to compute prediction of deviations of the VAR, $\tilde{\boldsymbol{X}}_t = \boldsymbol{X}_t - \boldsymbol{\mu}$, where $\boldsymbol{\mu}$ is the mean of $\boldsymbol{X}$, which can be computed in closed form.

Granger causality

- Let $(X_t)_{t\in\mathcal{T}}$ and $(Y_t)_{t\in\mathcal{T}}$ be two one-dimensional time series.
- We say X does not Granger cause Y if

$$\mathbb{E}[Y_t \mid X_{t-1}, Y_{t-1}, \ldots, X_{t-k}, Y_{t-k}, \ldots] = \mathbb{E}[Y_t \mid Y_{t-1}, \ldots, Y_{t-k}, \ldots]$$

- In words, X does not Granger cause Y if forecasting Y using past values of Y is the same as also using past values of X.
- For instance, if $\boldsymbol{X}_t = (X_t, Y_t)^T$ follows a VAR(2) model

$$\boldsymbol{X}_t = \boldsymbol{A}_0 + A_1 \boldsymbol{X}_{t-1} + A_2 \boldsymbol{Y}_{t-2} + \varepsilon_t,$$

then X will not Granger cause Y if $(A_1)_{21} = (A_2)_{21} = 0$.

Testing Granger causality

- Consider the general VAR(p) model

$$\boldsymbol{X}_t = \boldsymbol{A}_0 + A_1 \boldsymbol{X}_{t-1} + \cdots + A_p \boldsymbol{Y}_{t-p} + \varepsilon_t,$$

- Then $X^{(j)}$ does not Granger cause $X^{(i)}$ if $(A_1)_{ij} = \cdots = (A_p)_{ij} = 0$.

- To test this hypothesis, one should the test statistics

$$(T - (Pk^2 - k)) \left(\log\det\left(\widehat{\Sigma_r}\right) - \log\det\left(\widehat{\Sigma_u}\right) \right),$$

  where $\widehat{\Sigma_r}$ is the estimated covriance matrix assuming the restricted model and $\widehat{\Sigma_u}$ is the estimated covriance matrix for the unrestricted model.

- Asymptotically, this test statistic is distributed as χ^2_p.

Impulse Response Function

- The impulse response function of $X^{(i)}$ with respect to residual $\varepsilon^{(j)}$ is the change in $X^{(i)}_{t+h}$, $h \geq 0$, for a shock of one standard deviation in $\varepsilon^{(j)}_t$.
- For a VAR(1), we can write it as a V(ector)MA(∞):

$$\boldsymbol{X}_t = \boldsymbol{A}_0 + \boldsymbol{\varepsilon}_t + A\boldsymbol{\varepsilon}_{t-1} + \cdots + A^k\boldsymbol{\varepsilon}_{t-k} + \cdots$$

- Then, if σ_j is the standard deviation of $\varepsilon^{(j)}$, the impulse response function of $X^{(i)}$ with respect to a shock in $\varepsilon^{(j)}$ at time h is given by

$$\text{IRF}^{(ij)}_h = \sigma_j (A^h)_{ij}$$

- Confidence intervals for the IRF can be computed using asymptotic normality, Monte Carlo or bootstrap.

Impulse Response Function

- Let us go over the 

Jupyter Noebook `Time Series - VAR`.
- Do all questions on the 

Jupyter Notebook `Time Series - VAR - Macroeconomic Indicators Data`.

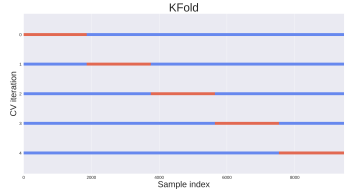
Cross-Validation for Time Series

ICs vs. CV

- We have used, for univariate data, information criteria (IC), such as AIC and BIC, to select model for inference and prediction.
- In this lecture, we will use another approach, the well-known cross-validation (CV).
- CV and ICs have the same goal of selecting models (or choosing hyper-parameters).
- CV is data-driven, almost model-free and of easy implementation. One of its disadvantage is the heavy computational cost.
- ICs depend more on theoretical assumptions and derivations of the particular model. In some particular cases, it is asymptotically equivalent to CV.

Cross-Validation for Time Series

- For time series, the train-validation-test split cannot be done as in the usual independent and identically distributed (iid) data.



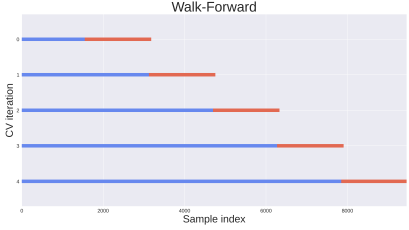
- This happen because of the obvious time ordering of data. The training of the model cannot look into the future.

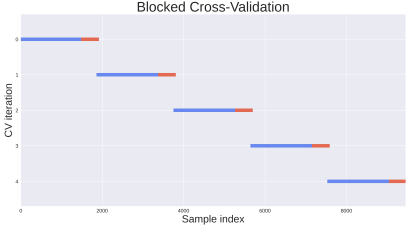
- There are several splitting techniques for times series:
  - Walk-forward splitting (available on `scikit-learn`).
  - Blocked and Rolling Window.
  - Combinatorial Purged CV (we will not use it here).

Cross-Validation for Time Series

Let us fix a model with hyper-parameter $\vartheta \in \mathbb{R}^d$, an error measure (such as MSE or MAE) and some possibilities for the hyper-parameters $\{\vartheta_i\}_{i=1,\ldots,K}$. The procedure works as follows:

- train (estimate) the model for each ϑ_i using the training set;
- For each trained model, compute the error measure in the validation set and choose the one with the smallest error.





Machine Learning for Time Series

Using Supervised Learning for Forecasting

- Assume we observe an one-dimensional time series X_t from $t = 1$ to $t = T$ and we want to predict X_{t+h} for time horizon $h \geq 1$.
- Denote $\boldsymbol{Z}_t = (X_{t-1}, \ldots, X_{t-p})$, where we are considering p lagged variable of X. We could also consider additional features and its lagged values in $\boldsymbol{Z}_t$.
- We then consider the following minimization problem

$$\arg\min_{f \in \mathcal{H}} \sum_{t=1}^{T-h} (X_{t+h} - f(\boldsymbol{Z}_t))^2$$

  for some class $\mathcal{H}$ that is defined by the Machine Learning model being considered.
- The main issue with this approach is that the sequential nature of the data is not taken into account. There are several new types of time series models in Machine Learning/Deep Learning, e.g. using LLM, see this link.
- Run Jupyter Notebook `Time Series - Sktime`.

Long Short-Term Memory - LSTM

Neural Networks for Sequence Data

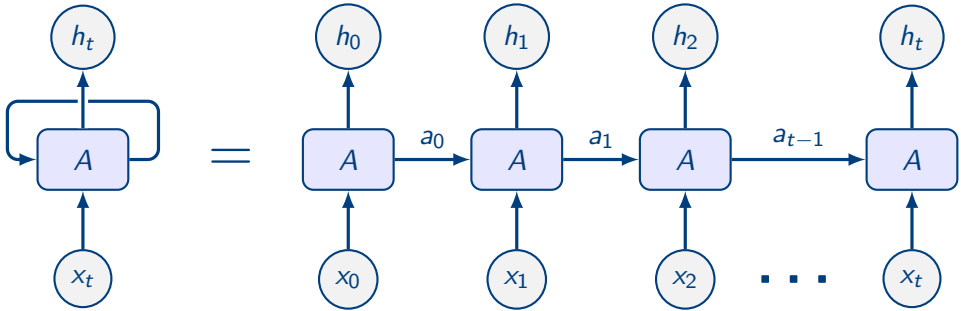
- Deep Learning and neural networks have clearly shown great capability of predicition in regression and classification problems from various areas of applications.
- We wish then to take advantage of these advances for time series data.
- Let's say we want to use the last 15 observations to observe the next one.
- We could then consider a feed-forward NN that takes 15-dimensional inputs and predicts an one-dimensional output.
- The clear problem of this approach is that we are not take into consideration the sequential nature of the data.

Neural Networks for Sequence Data

- We will go into the details of LSTM, but there are several other Machine/Deep Learning methods for time series.
- Several academic articles in Finance and Economics show that `XGBoost`, `Random Forests` and `Gradient Boosting` perform very well for time series data.
- `Prophet`, from Meta, is very easy-to-use and follows the Trend-Seasonality-Noise decomposition framework.
- `Scalecast` provides "uniform ML modeling" for time series.
- Unfortunately, we will not have the time to go over them.

Recurrent Neural Networks

- A class of neural network models that solves this problem is the so-called Recurrent Neural Networks:



- The description above is for the "many-to-many" problem with $T_x = T_h$, where T_x is the time length of the input, and T_h for the output.

- We could easily generalize for $(T_x = 1, T_h > 1)$, $(T_x > 1, T_h = 1)$ and $(T_x \neq T_h)$.

Recurrent Neural Networks

- Usually, inside the unit A the following operations are carried on:

$$a_t = g_1\left(\left(W^a \begin{bmatrix} a_{t-1} \\ x_t \end{bmatrix} + b^a\right),\right.$$

$$h_t = g_2(W^h a_t + b_h),$$

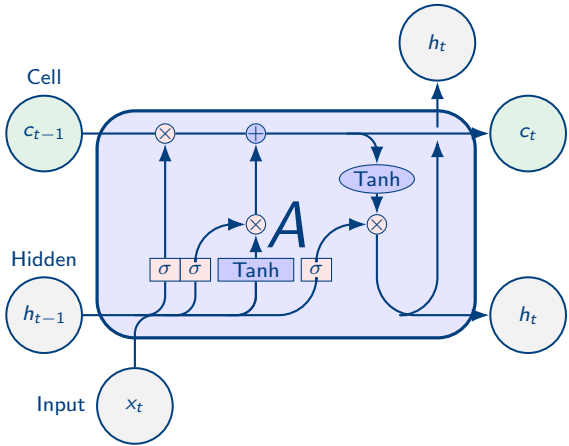
  where g_1 and g_2 are activation functions.

- The dimension of x_t and h_t can be chosen depending on the problem.

- The main issue when considering one-layer architecture in A is that during training, for long sequences, vanishing-exploding gradient phenomenons was observed.

LSTM

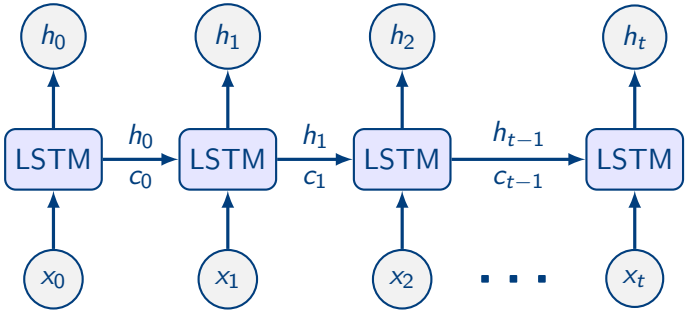
- The Long Short-Term Memory solves this problem by considering informational gates. An LSTM cell looks like:



- The gates are modeled through the operations $\boxed{\sigma}$ – $\boxed{\times}$.

LSTM

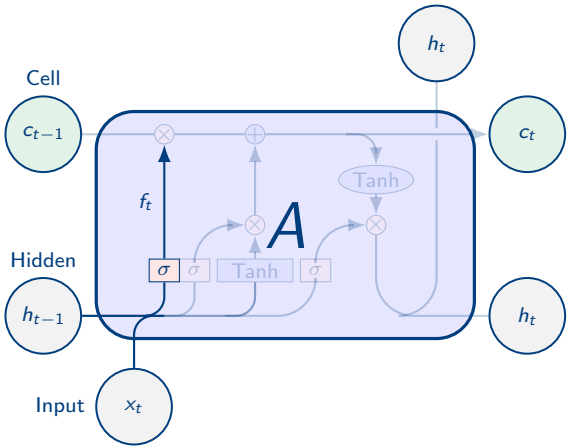
- We then stack the LSTM cells into the RNN architecture:



- It is common to consider `Dropout` layers between LSTM layers. Also, one can add `Dense` layer the output.

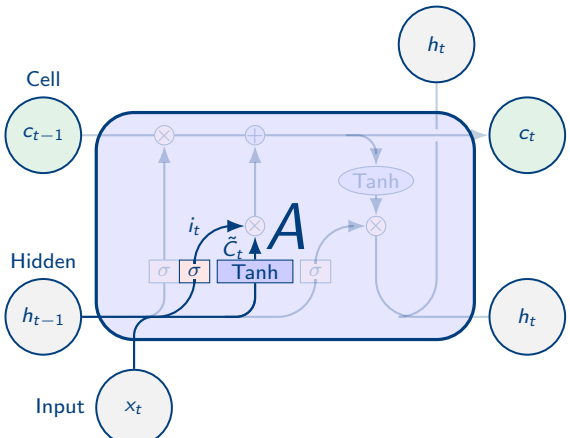
- Similarly to the RNN case, we could easily generalize for ($T_x = 1, T_h > 1$), ($T_x > 1, T_h = 1$) and ($T_x \neq T_h$).

Diving into LSTM - Forget gate



$$f_t = \sigma\left(W^f \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} + b^f\right)$$

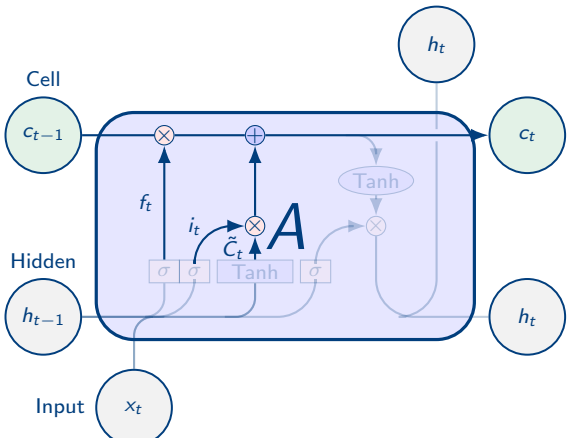
Diving into LSTM - Input Gate



$$i_t = \sigma\left(W^i \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} + b^i\right)$$

$$\tilde{c}_t = \tanh\left(W^c \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} + b^c\right)$$

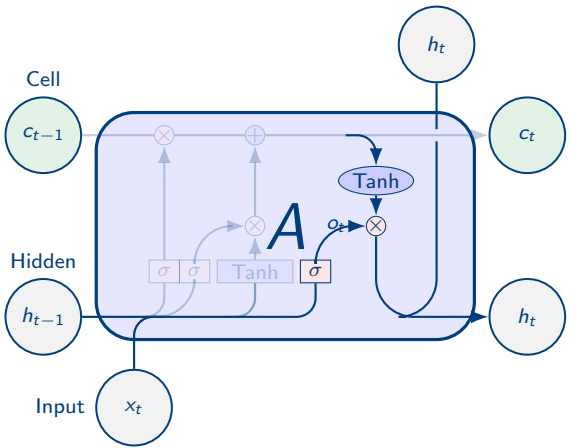
Diving into LSTM - Cell State



$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t$$

- f_t: should we forget the past cell state?
- i_t: how much of the past should impact the current cell state?
- c_t: this is the main idea of the LSTM architecture. The cell state it goes through the whole chain with simple interference from the past and current input

Diving into LSTM - Output Gate



$$o_t = \sigma\left(W^o \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix} + b^o\right)$$

$$h_t = o_t \odot \tanh(c_t)$$

Training and Prediction

- The loss function needs to consider the complete output and compare to the observed data. If we denote by $y = (y_1, \ldots, y_{T_h})$ the observed data, then the loss is given by

$$\mathcal{L}(y, h) = \sum_{t=1}^{T_h} \ell(y_t, h_t),$$

  for some point-to-point loss function ℓ.

- Training of LSTMs are done as usual: SGD-like optimization (`Adam`, for instance).

Backpropagation Through Time

- Notice that in our general RNN architecture, the networks parameters (weights and bias) do not change with time.

- Moreover, "many-to-many" case, the loss is composed by the sum of partial losses at each time t:

$$\mathcal{L}(y, h) = \sum_{t=1}^{T_h} \ell(y_t, h_t).$$

- Let ω be one of parameters of the networks. Then

$$\frac{\partial \mathcal{L}(y, h)}{\partial \omega} = \sum_{t=1}^{T_h} \frac{\partial \ell(y_t, h_t)}{\partial \omega} = \sum_{t=1}^{T_h} \partial_2 \ell(y_t, h_t) \frac{\partial h_t}{\partial \omega}.$$

$$\frac{\partial h_t}{\partial \omega} = \frac{\partial h_t}{\partial a_t} \sum_{k=1}^{t} \frac{\partial h_t}{\partial h_k} \frac{\partial h_k}{\partial \omega}$$

Gated Recurrent Unit (GRU)

- One of the issues with LSTM is the number of parameters and the computational cost for training.
- The Gated Recurrent Unit (GRU) removes some of the parameters of the LSTM, but keeping some of its essence.
- In fact, the forget gate f_t is replaced by $1 - i_t$ and the output gate is removed setting then $h_t = c_t$.

LSTM in Tensorflow

An LSTM layer in Tensorflow is

```
tf.keras.layers.LSTM(
    units,
    activation='tanh',
    recurrent_activation='sigmoid',...)
```

- units (dim of h_t): Positive integer, dimensionality of the output space.
- Activation: function to use. Default: hyperbolic tangent (tanh). If you pass None, no activation is applied (ie. "linear" activation: a(x) = x).
- recurrent_activation: Activation function to use for the recurrent step. Default: sigmoid (sigmoid). If you pass None, no activation is applied (ie. "linear" activation: a(x) = x).

LSTM in Tensorflow

Additional, important variable of LSTM:

- `dropout: Float between 0 and 1. Fraction of the units to drop for the linear transformation of the inputs. Default: 0.`
- `recurrent_dropout: Float between 0 and 1. Fraction of the units to drop for the linear transformation of the recurrent state. Default: 0.`
- `return_sequences: Boolean. Whether to return the last output in the output sequence, or the full sequence. Default: False.`
- `return_state: Boolean. Whether to return the last state in addition to the output. Default: False.`

LSTM in Tensorflow

```
lstm_model = tf.keras.models.Sequential([
    tf.keras.layers.LSTM(32, return_sequences=True),
    tf.keras.layers.Dense(units=1)
])
```

- The input shape x is `[batch, time, features]`, where time is T_x.
- The output shape h is `[batch, time, lstm_units]`.
- Let us go over the 

Jupyter Notebook `Time Series - LSTM`.
- Run 

Jupyter Notebook `Time Series - Scalecast` .

Time Series Regression

Time Series Analysis

- Compute the risk of a financial portfolio. For this, one needs to understand the future distribution of the multivariate time series with stochastic volatility (hence, a statistical inference problem).
- Predict the interest rate curve. For this, one could use dimensionality reduction, classical prediction using classical time series models (e.g. VAR) and Machine Learning techniques (e.g. LSTM).
- Fama–French factors and their capacity to explain the cross-sectional of returns. In this situation, one needs Time Series Regression techniques.

Fama-French Three Factor Model

Markowitz to CAPM

- In 1952, Henry Markowitz inaugurates the Modern Portfolio Theory that describes the decision of an investor as maximizing expected return of a portfolio while controlling for its variance.

- In 1964, William Sharpe showed the relation of Markowitz theory to his Capital Asset Pricing Model (CAPM):

$$R_t - R_f = \alpha + \beta_m(R_t^m - R_f) + \varepsilon_t,$$

where

  - R is the return (e.g. log return) of a given asset;
  - R_f is the risk-free rate;
  - R^m is the return of the market portfolio, i.e. a theoretical portfolio in which the weight of each asset is its market capitalization.

CAPM to APT to FF

- In 1976, Stephen Ross takes the next step and proposes the Arbitrage Pricing Theory (APT) where additional factors are considered to explain the return of a given asset:

$$R_t - R_f = \alpha + \beta_m(R_t^m - R_f) + \beta_1 f_t^1 + \cdots + \beta_k f_t^k + \varepsilon_t.$$

- Going to 1990s, a series of papers by Eugene Fama e Kenneth French proposed a methodology to structure specific factors.

- They started with the 3 factors Fama-French model.

FF Factors

- Let us first describe the Small minus Big (SMB) factor.
- We first sort the available companies by size, i.e. its market capitalization (price per share times the total number of shares).
- Divide the sorted list of companies in two equal parts. The first part is called Small and the second group is called Big.
- We now take the same list of companies and sort by the BM (book-to-market ratio), i.e. the ratio of the company's book value and its market capitalization. The book value is the value of all companies' assets minus the value of its liabilities.
- We then divide the sorted list in three parts: High ($> 70\%$), Medium ($< 70\%$ and $> 30\%$) and Low ($< 30\%$).

FF Factors

- Finally, we create 6 portfolios of companies by combining the groups above and compute their returns.
- SMB (Small minus Big):

$$\text{SMB} = \frac{1}{3}(\text{SL} + \text{SM} + \text{SH}) - \frac{1}{3}(\text{BL} + \text{BM} + \text{BH}).$$

- HML (High minus Low):

$$\text{SMB} = \frac{1}{2}(\text{SH} + \text{BH}) - \frac{1}{2}(\text{SL} + \text{BL}).$$

- FF 3 factors model:

$$R_t - R_f = \alpha + \beta_m(R_t^m - R_f) + \beta_{SMB}\text{SMB}_t + \beta_{HML}\text{HML}_t + \varepsilon_t.$$

Time Series Regression

Time Series Regression

- Let us consider a time series linear regression model with k regressors:

$$Y_t = \beta_0 + \beta_1 X_{t1} + \cdots \beta_k X_{tk} + \varepsilon_t,$$

  for $t = 1, \ldots, T$.

- In matrix form, we can write $\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}$, where $\boldsymbol{X}$ is a $T \times k + 1$ matrix.

- If $\boldsymbol{X}$ is full rank[3], then the OLS estimator for $\boldsymbol{\beta}$ is

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^\intercal \boldsymbol{X})^{-1} \boldsymbol{X}^\intercal \boldsymbol{Y}.$$

[3] the columns of $\boldsymbol{X}$ are linearly independent.

Assumptions

The usual assumptions on the model are:

A1 Linearity;

A2 Full rank: The rank of $\boldsymbol{X}$ is k;

A3 Exogeneity of the predictors: ε_t is uncorrelated with X_{sj} for all s and j;

A4 Zero mean: $\mathbb{E}[\varepsilon_t] = 0$;

A5 Homoscedastic: $\text{Var}(\varepsilon_t) = \sigma^2$;

A6 Uncorrelated: $\mathbb{E}[\varepsilon_t \varepsilon_s] = 0$, for $t \neq s$.

Assumption A3 might be replace by considering a conditional (on $\boldsymbol{X}$) version of A4-A6.

Properties of OLS estimator

- Under A1-A4, $\hat{\boldsymbol{\beta}}$ is unbiased: $\mathbb{E}[\hat{\boldsymbol{\beta}}] = \boldsymbol{\beta}$;
- Under A1-A6, $\text{Var}(\hat{\boldsymbol{\beta}}) = \sigma^2(\boldsymbol{X}^\intercal \boldsymbol{X})^{-1}$;
- Gauss-Markov Theorem: Under A1-A6, $\hat{\boldsymbol{\beta}}$ is the Best Linear Unbiased Estimator (BLUE).
- If we additionally assume that $\boldsymbol{\varepsilon} \mid \boldsymbol{X}$ is Gaussian, then a lot more could be said about $\hat{\boldsymbol{\beta}}$ under the small-sample assumption.

Violations of assumptions

The following are common violations of the assumptions previously discussed:

- Omitted variables;
- Endogeneity of predictors: ε_t is correlated with X_{sj}, for some s and j. A possible solution in this case is to find a so-called instrumental variable.
- Heteroskedasticity: the variance of ε_t is not constant. When the variance is known, a possible solution here is to consider generalized least squares with weight given by the variance.

Residual Diagnostics

The residual is defined as

$$\boldsymbol{e} = \boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}.$$

To verify the violations discussed above, one could perform residual diagnostics:

- plot the residuals;
- plot the auto-correlation function of the residuals (and perform the Durbin-Watson test or the Ljung-Box Q-test);
- use Engle's ARCH test for heteroscedasticity.

Spurious Regression

- When regressing non-stationary time series, one might observe the well-known phenomenon of spurious regression[4].
- This happens when, for instance, Y_t and x_t have a trending behavior and overall measures of fit become misleadingly "significant".
- An interesting example is when X and Y are independent random walks and we consider:

$$Y_t = \beta X_t + \varepsilon_t.$$

- By independence, $\beta = 0$, which implies that ε_t is also a random walk.
- In this case, we should observe $\hat{\beta} \approx 0$, but one can show that the OLS estimator converges to a non-Gaussian rv not centered in 0.

[4] For funny examples, see `https://www.tylervigen.com/spurious-correlations`.

Co-integration

- Therefore, regressing integrated time series is dangerous and can produce misleading relationships.
- One should always perform residual diagnostics or avoid regressing integrated series by considering first differences of them.
- We are allowed (in terms of OLS framework) to consider regression of integrated time series when the residual is stationary. In this case, we say the series are co-integrated.
- In other words, two time series X and Y are co-integrated if there exists β such that the time series

$$u_t = Y_t - \beta X_t$$

  is stationary.
- This is equivalent to saying that there exists a non-trivial linear combination of (X_t, Y_t) that is stationary.

Test for co-integration

- If β is known, one could test if $(u_t)_{t\in\mathcal{T}}$ is stationary using the ADF test.
- However, in practice, β will be estimated by regressing Y into X using OLS and the critical values for the ADF test are non longer valid.
- A possible alternative is the well-known Engle-Granger two-step method. It is based on Granger's Representation Theorem that says that if two or more non-stationary variables have a valid Error Correction Model (ECM), they are co-integrated.
  - Step 1: estimate β by OLS and compute the residual $\hat{u}_t = Y_t - \hat{\beta}X_t$ and test it if stationary;
  - Step 2: estimate an ECM:

$$\Delta Y_t = \gamma \Delta X_{t-1} + \underbrace{\alpha \hat{u}_{t-1}}_{\text{error correction}} + \varepsilon_t.$$

    All terms are stationary and hence OLS performs well. If α is significantly different from zero, we can argue that there is a valid ECM.

Dynamic regression models

- Let us assume that the assumption A6 uncorrelated errors ε_t is not verified.
- Instead, let us consider the following dynamic regression model:

$$Y_t = \beta_0 + \beta_1 X_{t1} + \cdots \beta_k X_{tk} + \varepsilon_t,$$

where $(\varepsilon_t)_{t\in\mathcal{T}}$ follows an ARIMA(p, q).

- For simplicity, let us assume that $(\varepsilon_t)_{t\in\mathcal{T}}$ follows an ARMA$(1, 1)$:

$$\varepsilon_t = a_0 + a_1\varepsilon_{t-1} + \eta_t + b_1\eta_{t-1},$$

where $(\eta_t)_{t\in\mathcal{T}}$ is a white noise.

Estimation of Dynamic regression models

- To estimate the model above, we cannot perform the usual OLS estimator for $\boldsymbol{\beta}$.
- The OLS estimator is no longer BLUE and inference on the parameters will no longer give meaningful results.
- Some predictor might appear important. This is known as spurious regression.
- To solve this, one needs to either minimize the square of the true residuals η or perform MLE.
- Additionally, in this situation we need to guarantee that the time series Y, $X_1, \ldots, X_k$ are all stationary. Otherwise, the estimator of $\boldsymbol{\beta}$ might not be consistent.
- One exception is when Y, $X_1, \ldots, X_k$ are co-integrated.

Estimation of Dynamic regression models

- When some of the time series Y, $X_1, \ldots, X_k$ are non-stationary, we consider taking the first difference to make then stationary.
- Taking the difference has another advatange because it does not change the nature of the regression model. We have then a "model in differences", instead of "model in levels":

$$\Delta Y_t = \beta_1 \Delta X_{t1} + \cdots \beta_k \Delta X_{tk} + \Delta \varepsilon_t.$$

- Regression models with ARIMA errors is equivalent to regression models in differences with ARMA errors.

Latent Models

Hidden Markov Models

Markov Chain

Definition (Markov chain)

Let $(X_n)_{n\in\mathbb{N}}$ be a stochastic process taking values in a countable set I. We say that $(X_n)_{n\in\mathbb{N}}$ is a *Markov chain* if it satisfies the Markov property:

$$\mathbb{P}(X_{n+1} = j \mid X_n = i, X_{n-1} = i_{n-1}, \ldots, X_0 = i_0) = \mathbb{P}(X_{n+1} = j \mid X_n = i),$$

for any $n \geq 0$ and any $i, i_0, \ldots, i_{n-1}, j \in I$.

A Markov chain is a process that the past realizations do not give more knowledge to forecast the next step than just knowing the present state. In other words, given the present, past and future are independent.

Markov Chain

A distribution on I is any sequence $(\lambda_i)_{i\in I}$ such that $\lambda_i \geq 0$ and $\sum_{i\in I} \lambda_i = 1$. A (possibly infinite) matrix $P = (p_{ij})_{i,j\in I}$ is said to be stochastic if each row $P = (p_{ij})_{j\in I}$, $i \in I$, is a distribution:

$$\sum_{j\in I} p_{ij} = 1.$$

We say that $(X_n)_{n\in\mathbb{N}}$ is Markov(λ, P) if

- $\mathbb{P}(X_0 = i) = \lambda_i$, for any $i \in I$, and
- for any $n \geq 0$ and any $i, i_0, \ldots, i_{n-1}, j \in I$,

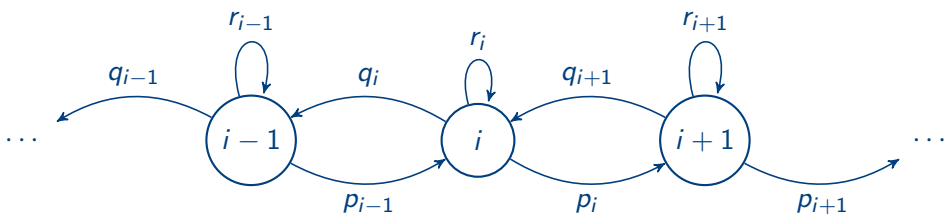
$$\mathbb{P}(X_{n+1} = j \mid X_n = i) = p_{ij}.$$

Birth-Death Chain

Let us consider a Markov chain with $I = \{0, \ldots, d\}$, where d could be $+\infty$, and

$$p_{ij} = \begin{cases} q_i, & \text{if } j = i-1, \\ r_i = 1 - p_i - q_i, & \text{if } j = i, \\ p_i, & \text{if } j = i+1, \end{cases}$$

where $q_0 = 0$ and $p_d = 0$, when $d < +\infty$.



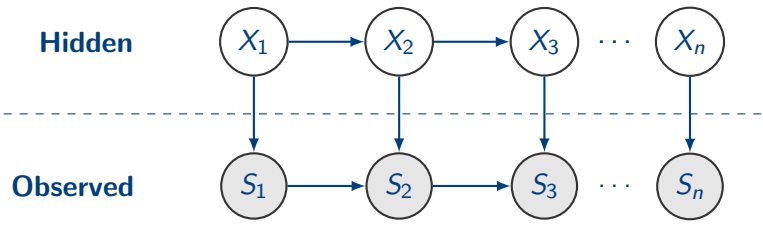
Hidden Markov Model

- Let $(X_n)_{n\in\mathbb{N}}$ be Markov(λ, P) defined on the state space I (hidden).
- Moreover, let $(S_n)_{n\in\mathbb{N}}$ be a stochastic process defined on a set of signals $\mathcal{S}$ (observed).
- We assume that

$$\begin{aligned}&\mathbb{P}(S_n = s \mid X_n = i, S_{n-1} = s_{n-1}, X_{n-1} = i_{n-1}, \ldots, S_0 = s_0, X_0 = i_0)\\ &= \mathbb{P}(S_n = s \mid X_n = i) = \alpha_{is}.\end{aligned}$$

- The sequence $(\alpha_{is})_{s\in\mathcal{S}}$ is a probability distribution on $\mathcal{S}$ and called output/emission probabilities. It is common to assume that these probabilities come from a Gaussian distribution.
- The parameters of this model are the matrices P, $A = (\alpha_{is})_{I,\mathcal{S}}$ and the initial distribution λ.

Hidden Markov Model



Hidden Markov Model

- Notice that S alone is clearly not a Markov chain, but, conditional on X_n, the future $S_n, X_{n+1}, S_{n+1}, \ldots$ is independent of the past $S_{n-1}, X_{n-1}, \ldots, S_0, X_0$.
- Define now $\boldsymbol{S}_n = (S_0, \ldots, S_n)$, $\boldsymbol{s}_n = (s_0, \ldots, s_n)$, $\boldsymbol{X}_n = (X_0, \ldots, X_n)$, $\boldsymbol{i}_n = (i_0, \ldots, i_n)$. By the assumption above, we find that

$$\mathbb{P}(\boldsymbol{S}_n = \boldsymbol{s}_n, \boldsymbol{X}_n = \boldsymbol{i}_n \mid \theta) = \lambda_{i_0} \prod_{k=1}^{n} p_{i_{k-1}, i_k} \alpha_{i_k, s_k}.$$

- We are going to analyze three problems under this model: calculating the probability of observing a sequence $\boldsymbol{s}_n$ (likelihood); finding the most likely hidden path $\boldsymbol{x}_n$ given an observation $\boldsymbol{s}_n$ (decoding); and estimating the parameters θ given an observation $\boldsymbol{s}_n$ (inference).

Likelihood

- Notice that calculating the probability of observing a sequence $\boldsymbol{s}_n$ means computing:

$$\mathbb{P}(\boldsymbol{S}_n = \boldsymbol{s}_n) = \sum_{\boldsymbol{i}_n} \mathbb{P}(\boldsymbol{S}_n = \boldsymbol{s}_n, \boldsymbol{X}_n = \boldsymbol{i}_n) = \sum_{\boldsymbol{i}_n} \lambda_{i_0} \prod_{k=1}^{n} p_{i_{k-1}, i_k} \alpha_{i_k, s_k}.$$

- However, there are too many terms in this sum to be computationally feasible. Define then

$$F_n(\boldsymbol{s}_n, j) = \mathbb{P}(\boldsymbol{S}_n = \boldsymbol{s}_n, X_n = j).$$

- Then

$$\mathbb{P}(X_n = j \mid \boldsymbol{S}_n = \boldsymbol{s}_n) = \frac{\mathbb{P}(\boldsymbol{S}_n = \boldsymbol{s}_n, X_n = j)}{\mathbb{P}(\boldsymbol{S}_n = \boldsymbol{s}_n)} = \frac{F_n(\boldsymbol{s}_n, j)}{\sum_{i \in I} F_n(\boldsymbol{s}_n, i)}.$$

Likelihood

- By conditioning on $X_{n-1} = i$ and summing over all possible $i \in I$, we find the following recursive formula for F_n

$$F_n(\boldsymbol{s}_n, j) = \alpha_{j,s_n} \sum_{i \in I} F_{n-1}(\boldsymbol{s}_{n-1}, i) p_{ij}.$$

- The first term is given by

$$F_0(s_0, j) = \mathbb{P}(X_0 = j, S_0 = s_0) = \lambda_j \alpha_{j,s_0}.$$

- One can use the probabilities F_n to compute the distribution of a given sequence of signals (i.e. the likelihood of $\boldsymbol{S}_n$ given the parameters θ).

- This is called the forward approach. A different way, called backward approach, uses the probabilities $\mathbb{P}(S_{k+1} = s_{k+1}, \ldots, S_n = s_n \mid X_k = i)$.

Most likely path

- We will now consider the problem of estimating the hidden states once we observed the $\boldsymbol{s}_n$, also called decoding.
- This will be accomplished by maximizing $\mathbb{P}(\boldsymbol{X}_n = \boldsymbol{i}_n \mid \boldsymbol{S}_n = \boldsymbol{s}_n)$ over $\boldsymbol{i}_n$.
- By the conditional probability formula, this is equivalent to maximizing $\mathbb{P}(\boldsymbol{X}_n = \boldsymbol{i}_n, \boldsymbol{S}_n = \boldsymbol{s}_n)$.

Most likely path

- To solve this problem, define

$$V_k(j) = \max_{\boldsymbol{i}_{k-1}} \mathbb{P}(\boldsymbol{X}_{k-1} = \boldsymbol{i}_{k-1}, X_k = j, \boldsymbol{S}_k = \boldsymbol{s}_k).$$

- Conditioning on $\boldsymbol{S}_{k-1} = \boldsymbol{s}_{k-1}$:

$$\mathbb{P}(\boldsymbol{X}_{k-1} = \boldsymbol{i}_{k-1}, X_k = j, \boldsymbol{S}_k = \boldsymbol{s}_k) \quad = \alpha_{j,s_k} p_{i_{k-1},j} \mathbb{P}(\boldsymbol{X}_{k-1} = \boldsymbol{i}_{k-1}, \boldsymbol{S}_{k-1} = \boldsymbol{s}_{k-1}).$$

- Hence

$$V_k(j) = \alpha_{j,s_k} \max_{i_{k-1}} V_{k-1}(i_{k-1}) p_{i_{k-1},j}.$$

- Moreover,

$$V_0(j) = \mathbb{P}(X_0 = j, S_0 = s_0) = \lambda_j \alpha_{j,s_0}.$$

- This is a recursion formula for $V_k(j)$ for each $j \in I$.

Most likely path

To find the sequence that maximizes $\mathbb{P}(\boldsymbol{X}_n = \boldsymbol{i}_n, \boldsymbol{S}_n = \boldsymbol{s}_n)$ proceed as follows:

- let i_n be the maximizing state for $V_n(j)$, and write

$$V_n(i_n) = \max_{\boldsymbol{i}_{n-1}} \mathbb{P}(\boldsymbol{X}_{n-1} = \boldsymbol{i}_{n-1}, X_n = i_n, \boldsymbol{S}_n = \boldsymbol{s}_n) = \alpha_{i_n, s_n} \max_i p_{i,i_n} V_{n-1}(i).$$

- Then, define i_{n-1} the value of i that attains the maximum above and perform a recursion of the equation above up to time 0.
- This is called Viterbi Algorithm.

Parameter Estimation

- We want to derive a procedure to find the maximum likelihood estimator:

$$\hat{\theta} = \arg\max_{\theta} \mathbb{P}(\boldsymbol{S}_n = \boldsymbol{s}_n | \theta),$$

where the parameters θ are given by P, $A = (\alpha_{is})_{I,\mathcal{S}}$ and the initial distribution λ.

Parameter Estimation

- Given the observed $\boldsymbol{S}_n = \boldsymbol{s}_n$ and the parameter θ, consider

$$\gamma_k(\boldsymbol{s}_n, i|\theta) = \mathbb{E}[1_{\{X_k=i\}} \mid \boldsymbol{S}_n = \boldsymbol{s}_n, \theta],$$

$$\xi_k(\boldsymbol{s}_n, i, j|\theta) = \mathbb{E}[1_{\{X_k=i, X_{k+1}=j\}} \mid \boldsymbol{S}_n = \boldsymbol{s}_n, \theta].$$

- Then,

$$\sum_{k=1}^{n} \gamma_k(\boldsymbol{s}_n, i|\theta) = \text{ expected number of visits to state } i,$$

$$\sum_{k=1}^{n} \xi_k(\boldsymbol{s}_n, i, j|\theta) = \text{ expected number of transitions from } i \text{ to } j.$$

- Notice we cannot estimate these quantities directly since we do not observe $(X_n)_{n\in\mathbb{N}}$.

Parameter Estimation

- However, assuming an estimator for γ_k and ξ_k (i.e. given the estimator of the most likely path), we are able to estimate P, A and λ as follows:

$$\bar{p}_{ij} = \frac{\sum_{k=1}^{n} \xi_k(\boldsymbol{s}_n, i, j|\theta)}{\sum_{k=1}^{n} \gamma_k(\boldsymbol{s}_n, i|\theta)},$$

$$\bar{\alpha}_{i,s} = \frac{\sum_{k=1}^{n} \gamma_k(\boldsymbol{s}_n, i|\theta) 1_{\{S_k = s\}}}{\sum_{k=1}^{n} \gamma_k(\boldsymbol{s}_n, i|\theta)},$$

$$\bar{\lambda}_i = \gamma_0(\boldsymbol{s}_n, i|\theta).$$

- Furthermore, our estimation of the most likely path were computed for a given parameter θ.

- The Baum–Welch algorithm to estimate θ is the iteration of the procedure above: given an initial parameter θ_0, iterate the procedure above to create sequence θ_k that should converge to $\hat{\theta}$.

Parameter Estimation

- We will now show an efficient procedure to estimate γ_k and ξ_k.
- Notice

$$\gamma_k(\boldsymbol{s}_n, i|\theta) = \mathbb{P}(X_k = i|\boldsymbol{S}_n = \boldsymbol{s}_n, \theta) = \frac{\mathbb{P}(\boldsymbol{S}_n = \boldsymbol{s}_n, X_k = i|\theta)}{\mathbb{P}(\boldsymbol{S}_n = \boldsymbol{s}_n|\theta)},$$

- Moreover,

$$\mathbb{P}(\boldsymbol{S}_n = \boldsymbol{s}_n, X_k = i|\theta) = \underbrace{\mathbb{P}(S_{k+1} = s_{k+1}, \ldots, S_n = s_n|X_k = i, \theta)}_{B_k(\boldsymbol{s}_{k+1:n}, i|\theta)} F_k(\boldsymbol{s}_k, i|\theta),$$

where $\boldsymbol{s}_{k+1:n} = (s_{k+1}, \ldots, s_n)$ and $F_k(\boldsymbol{s}_k, i) = \mathbb{P}(\boldsymbol{S}_k = \boldsymbol{s}_k, X_k = i|\theta)$.

Parameter Estimation

- An efficient way to compute B_k is similar to what we have done for F: define $B_n(\cdot, i) = 1$ and notice

$$B_k(\boldsymbol{s}_{k+1:n}, i|\theta) = \sum_{j \in I} p_{ij} \alpha_{j,s_{k+1}} B_{k+1}(\boldsymbol{s}_{k+2:n}, j|\theta).$$

- Moreover, the denominator of γ_k can be written as

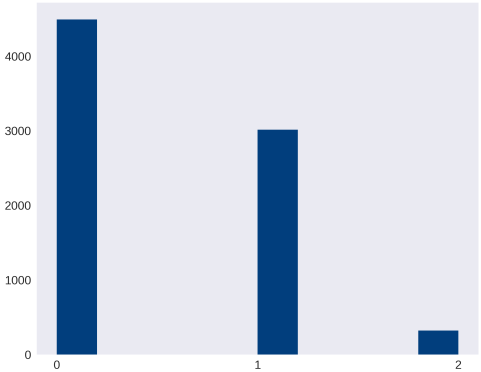
$$\mathbb{P}(\boldsymbol{S}_n = \boldsymbol{s}_n|\theta) = \sum_{i \in I} \mathbb{P}(\boldsymbol{S}_n = \boldsymbol{s}_n, X_k = i|\theta) = \sum_{i \in I} B_k(\boldsymbol{s}_{k+1:n}, i|\theta) F_k(\boldsymbol{s}_k, i|\theta).$$

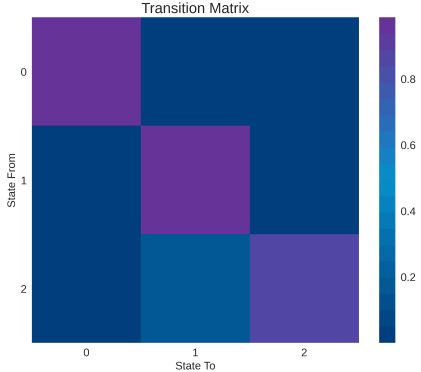
- Finally, for ξ_k, notice that

$$\begin{aligned} \xi_k(\boldsymbol{s}_n, i, j|\theta) &= \frac{\mathbb{P}(X_k = i, X_{k+1} = j, \boldsymbol{S}_n = \boldsymbol{s}_n|\theta)}{\mathbb{P}(\boldsymbol{S}_n = \boldsymbol{s}_n|\theta)} \\ &= \frac{F_k(\boldsymbol{s}_k, i|\theta) p_{ij} \alpha_{j,s_{k+1}} B_{k+1}(\boldsymbol{s}_{k+2:n}, j|\theta)}{\mathbb{P}(\boldsymbol{S}_n = \boldsymbol{s}_n|\theta)}. \end{aligned}$$

Application to stock price data

- Let us consider the S&P 500 index data.
- We compared several `GaussianHMM` models (emission probabilities are Gaussian) and the best model (using the `score`, i.e. log probability under the model).
- The best model has 3 hidden states.





HMM results

Run Jupyter Notebook `Time Series - Hidden Markov Model` for a different data set. Model parameters might need to change.

Kalman Filter

State Space Model

- Kalman filer is estimation (filtering) approach to estimate a Gaussian state space model:

$$\begin{cases} \text{Observation eq.} : \boldsymbol{S}_t = A\boldsymbol{X}_t + \boldsymbol{\eta}_t, \\ \text{(hidden) State eq.} : \boldsymbol{X}_t = B\boldsymbol{X}_{t-1} + \boldsymbol{\varepsilon}_t. \end{cases}$$

- $\boldsymbol{\eta}$ and $\boldsymbol{\varepsilon}$ are white noises with covariances R and Q, respectively, with

$$\mathbb{E}[\boldsymbol{\varepsilon}\boldsymbol{\eta}^\intercal] = 0, \text{ for all } t, s.$$

- $\boldsymbol{\eta}$ and $\boldsymbol{\varepsilon}$ are distributed as multivariate Gaussian.
- The initial value of the hidden state $\boldsymbol{X}_0$ are uncorrelated to $\boldsymbol{\varepsilon}$ and $\boldsymbol{\eta}$.
- VAR(p) and ARMA(p, q) can be written (with the lags of the process) can be written as a state space model.

Filtering and Smoothing

Assume we have observed $\boldsymbol{S}_1, \ldots, \boldsymbol{S}_T$. We can consider three estimation problems:

- Prediction: estimate $\boldsymbol{X}_t$ from $\boldsymbol{S}_1, \ldots, \boldsymbol{S}_{t-1}$;
- Filtering: estimate $\boldsymbol{X}_t$ from $\boldsymbol{S}_1, \ldots, \boldsymbol{S}_t$;
- Smoothing: estimate $\boldsymbol{X}_t$ from $\boldsymbol{S}_1, \ldots, \boldsymbol{S}_T$.

We are going to provide Kalman's solution for filtering and prediction since it is more financially interesting.

Necessary result

The following well-known theorem for Gaussian distribution is going to be paramount for the development of Kalman filter. It is also very useful in other areas of Machine Learning as, for instance, Gaussian Processes (GPs).

Theorem

Let $\boldsymbol{Z}$ be a n-dimensional Gaussian random variable with mean $\boldsymbol{\mu}$ and covariance Σ and write $\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{Z}_1 \\ \boldsymbol{Z}_2 \end{bmatrix}$, with each component's dimensions being n_1 and n_2 respectively. Write

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix} \text{ and } \Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22}. \end{bmatrix}$$

Then $\boldsymbol{Z}_1 \mid \boldsymbol{Z}_2$ is also Gaussian with mean and covariance given by:

$$\mathbb{E}[\boldsymbol{Z}_1 \mid \boldsymbol{Z}_2] = \boldsymbol{\mu}_1 + \Sigma_{12}\Sigma_{22}^{-1}(\boldsymbol{Z}_2 - \boldsymbol{\mu}_2),$$
$$Var(\boldsymbol{Z}_1 \mid \boldsymbol{Z}_2) = \Sigma_1 1 - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}.$$

Kalman Filter

- Let us consider first the filtering problem.
- Define the following notation:

$$\boldsymbol{X}_{t|h} = \mathbb{E}[\boldsymbol{X}_t \mid \boldsymbol{S}_1, \ldots, \boldsymbol{S}_h],$$

$$\boldsymbol{S}_{t|h} = \mathbb{E}[\boldsymbol{S}_t \mid \boldsymbol{S}_1, \ldots, \boldsymbol{S}_h],$$

$$P_{t|h} = \mathrm{Var}(\boldsymbol{X}_t \mid \boldsymbol{S}_1, \ldots, \boldsymbol{S}_h).$$

- By the dynamics of the state equation:

$$\begin{cases} \boldsymbol{X}_{t+1|t} = B\boldsymbol{X}_{t|t}, \\ P_{t+1|t} = BP_{t+1|t}B^\intercal + Q. \end{cases}$$

- Moreover, $\boldsymbol{S}_{t+1|t} = A\boldsymbol{X}_{t+1|t}$.
- These are called the forecasting step.

Kalman Filter

- Now we describe the updating step.
- By the Gaussian nature of the model, we find that

$$\begin{bmatrix} \boldsymbol{X}_{t+1} \\ \boldsymbol{S}_{t+1} \end{bmatrix} \mid \boldsymbol{S}_1, \ldots, \boldsymbol{S}_t$$

is Gaussian with mean

$$\begin{bmatrix} \boldsymbol{X}_{t+1|t} \\ \boldsymbol{S}_{t+1|t} \end{bmatrix}$$

and covariance

$$\begin{bmatrix} P_{t+1|t} & P_{t+1|t}A^\intercal \\ AP_{t+1|t} & AP_{t+1|t}A^\intercal + R \end{bmatrix}.$$

- This implies by the conditional property for Gaussian random variables:

$$\begin{cases} \boldsymbol{X}_{t+1|t+1} &= \boldsymbol{X}_{t+1|t} + P_{t+1|t}A^\intercal(AP_{t+1|t}A^\intercal + R)^{-1}(\boldsymbol{S}_{t+1} - \boldsymbol{S}_{t+1|t}), \\ P_{t+1|t+1} &= P_{t+1|t} - P_{t+1|t}A^\intercal(AP_{t+1|t}A^\intercal + R)^{-1}AP_{t+1|t}. \end{cases}$$

Kalman Filter

- Therefore, initializing $\boldsymbol{X}_{0|0}$ and $P_{0|0}$, we can use the filtering and updating step to compute $\boldsymbol{X}_{t|t}$ and $\boldsymbol{S}_{t|t}$ for every $t = 1, \dots, T$.
- We have also solved the prediction problem.
- Indeed, notice that

$$\begin{cases} \boldsymbol{X}_{t+1|t} &= B\boldsymbol{X}_{t|t}, \\ \boldsymbol{X}_{t|t} &= \boldsymbol{X}_{t|t-1} + P_{t|t-1}A^\intercal(AP_{t|t-1}A^\intercal + R)^{-1}(\boldsymbol{S}_t - \boldsymbol{S}_{t|t-1}), \\ \boldsymbol{S}_{t|t-1} &= A\boldsymbol{X}_{t|t-1}. \end{cases}$$

- Let us define the (Kalman) gain matrix:

$$K_t = BP_{t|t-1}A^\intercal(AP_{t|t-1}A^\intercal + R)^{-1}.$$

- Hence

$$\boldsymbol{X}_{t+1|t} = B\boldsymbol{X}_{t|t-1} + K_t(\boldsymbol{S}_t - A\boldsymbol{X}_{t|t-1}).$$

Estimation

- Notice that we can write, by conditioning:

$$p(\boldsymbol{s}_1, \ldots, \boldsymbol{s}_T) = p(\boldsymbol{s}_T \mid \boldsymbol{s}_1, \ldots, \boldsymbol{s}_{T-1}) p(\boldsymbol{s}_{T-1} \mid \boldsymbol{s}_1, \ldots, \boldsymbol{s}_{T-2}) \cdots p(\boldsymbol{s}_2 \mid \boldsymbol{s}_1) p(\boldsymbol{s}_1).$$

- We have seen that $p(\boldsymbol{s}_t \mid \boldsymbol{s}_1, \ldots, \boldsymbol{s}_{t-1})$ is Gaussian with mean $\boldsymbol{S}_{t|t-1}$ and variance $AP_{t|t-1}A^\intercal + R$.

- Since every distribution was assumed Gaussian, we can perform the Maximum Likelihood Estimator (MLE) for the parameters A, B, R and Q. For large systems, Expectation-Maximization (EM) is usually performed.

Fitting and Forecasting Brent and WTI future prices curve using Machine Learning

Data Analysis I

- This paper was co-authored by Mario Figueiredo (master student at IMPA) and published at Digital Finance in 2022.
- `https://link.springer.com/article/10.1007/s42521-022-00069-3`.
- We analyze two products: the ICE Brent Crude futures, referred as Brent, and the NYMEX West Texas Intermediate Crude Oil futures, referred as WTI.
- We have collected daily settlement prices for the first 36 contracts – whenever they were available – since 1985 up to the end of 2020.

Data Analysis II

We started by analyzing the number of available contracts

Figure: Number of Available Contracts - with settlement prices - for Brent from 1988 to 2020

Data Analysis III

We decided to use 15 contracts since year 2000

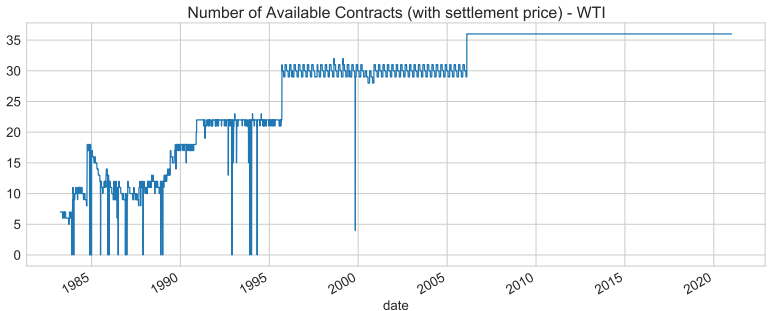


Figure: Number of Available Contracts - with settlement prices - for WTI from 1985 to 2020

Data Analysis IV

Here, we can see the price behaviour of the first contract rolled

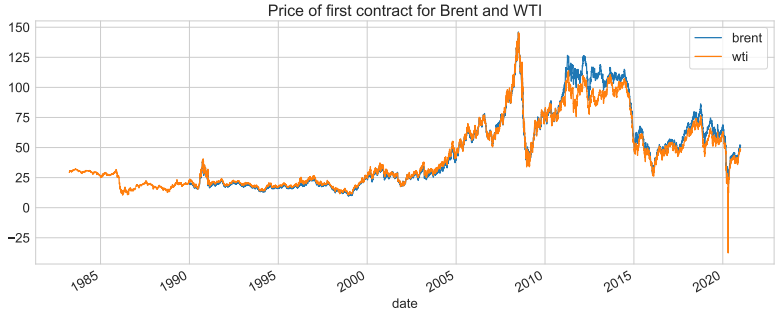


Figure: Price of the first contracts of Brent and WTI from 1985 to 2020

Dynamic Nelson-Siegel I

- The Dynamic Nelson-Siegel model is based on the Nelson-Siegel parametrization.
- Let $F_t(\tau)$ be the price of a contract with maturity $t+\tau$ at time t, with time measured in years. Then, the Nelson-Siegel parametrization of $F_t(\tau)$ is given as:

$$F_t(\tau) = L_t + S_t\left(\frac{1-\exp(-\lambda\tau)}{\lambda\tau}\right) + C_t\left(\frac{1-\exp(-\lambda\tau)}{\lambda\tau} - \exp(-\lambda\tau)\right) + \epsilon_t(\tau), \quad (1)$$

$$\epsilon_t \sim N(0, H_t).$$

Dynamic Nelson-Siegel II

- The error component ϵ_t is considered to follow a normal distribution with 0 mean and covariance matrix H_t.
- L_t, S_t and C_t are the three factors of the model, commonly referred to as the "level", "slope" and "curvature" factors.
- They are closely related to the first three principal components obtained by applying PCA.
- The parameter λ is considered to be fixed through time, although it could vary. This parameter controls the decaying rate of the factor loading.

Dynamic Nelson-Siegel III

The Level factor loading is a constant function, capturing the average level of the term structure.

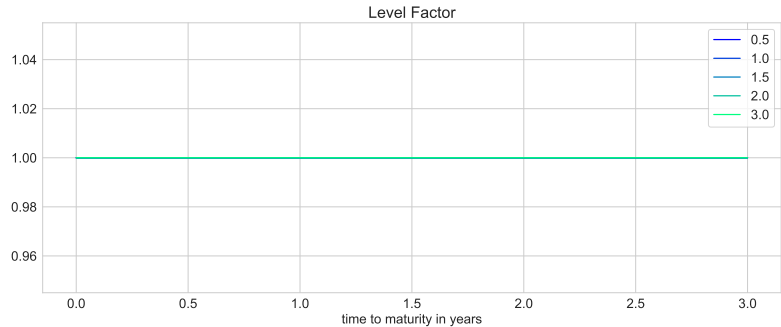


Figure: Level Factor versus Time to Maturity (τ) in years for increasing values of λ

Dynamic Nelson-Siegel IV

The Slope factor loading is a decaying function, capturing the slope of the term structure

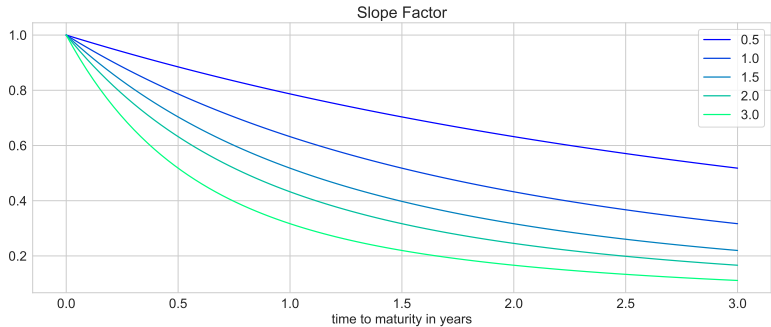


Figure: Slope Factor versus Time to Maturity (τ) in years for increasing values of λ

Dynamic Nelson-Siegel V

The Curvature factor loading is a function which achieves a maximum value, then decays, capturing the curvature of the term structure

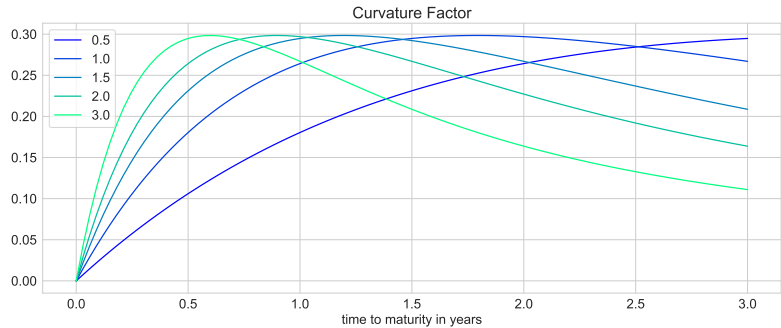
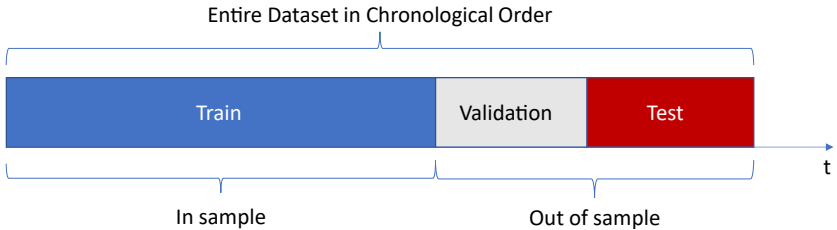


Figure: Curvature Factor versus Time to Maturity (τ) in years for increasing values of λ

Estimation Methods I

- For all models, we divide the data into train, validation and test.



- Train Data: used to estimate the models parameters. 80% of the total data.
- Validation Data: used to choose the best models, based on global estimation methods parameters. 10% of the total data.
- Test Data: used to test the model in unseen data. 10% of the total data.

OLS Approach I

- As the name suggests, in this approach, the factors - $\boldsymbol{\alpha}_t$ - are estimated via linear regression for each time step. Hence, we can write the model as the following:

$$\begin{aligned} \mathbf{y}_t &= Z_t(\Theta)\boldsymbol{\alpha}_t + \mathbf{d}_t(\Theta) + \boldsymbol{\epsilon}_t, \\ \boldsymbol{\epsilon}_t &\sim N(0, H_t), \end{aligned} \tag{2}$$

where

- $\mathbf{y}_t$ is the matrix of observations,
- $Z_t(\Theta)$ is a deterministic matrix of shape $p \times m$, whose values depend on the parameters Θ and $\mathbf{d}_t$ is a deterministic vector of shape $p \times 1$ that also depends on Θ.
- The component error $\boldsymbol{\epsilon}_t$ follows a normal distribution with mean 0 and variance matrix given by H_t.

OLS Approach II

For the dynamic Nelson-Siegel model, we can write:

$$\Theta = \lambda, \tag{3a}$$

$$Z_t(\Theta) = \begin{bmatrix} 1 & \left(\frac{1-exp(-\lambda\tau_{t,1})}{\lambda\tau_{t,1}}\right) & \left(\frac{1-exp(-\lambda\tau_{t,1})}{\lambda\tau_{t,1}} - exp(-\lambda\tau_{t,1})\right) \\ \vdots & \vdots & \vdots \\ 1 & \left(\frac{1-exp(-\lambda\tau_{t,p})}{\lambda\tau_{t,p}}\right) & \left(\frac{1-exp(-\lambda\tau_{t,p})}{\lambda\tau_{t,p}} - exp(-\lambda\tau_{t,p})\right) \end{bmatrix}, \tag{3b}$$

$$\boldsymbol{\alpha}_t = \begin{bmatrix} L_t \\ S_t \\ C_t \end{bmatrix}, \tag{3c}$$

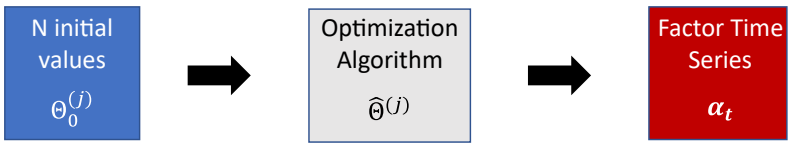
$$\mathbf{d}_t = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{3d}$$

OLS Approach III

- The estimation of Θ is denoted by $\hat{\Theta}$, which is calculated by minimizing the total error across the entire train time series, written as follows:

$$\hat{\Theta} = \underset{\Theta}{argmin} \left(\sum_{t=0}^{n_{train}-1} \sum_{j=0}^{p-1} \epsilon_{t,j}(\Theta)^2 \right). \tag{4}$$

- We present below the algorithm to obtain the estimates



OLS Approach IV

- After the values of $\boldsymbol{\alpha}_t$ are retrieved, they are modeled by Equation (5) below, where $T_{\Delta t}$ and c_{Δ_t} will vary depending on the estimation process applied.
- $\boldsymbol{\eta}_t$ is an error component with covariance matrix given by Q_t, which might depend on Θ. We will consider that this error component follows a normal distribution for simplicity.

$$\begin{aligned} \boldsymbol{\alpha}_{t+1} &= T_{\Delta t}(\Theta, \boldsymbol{\alpha}_t) + \mathbf{c}_{\Delta t} + \boldsymbol{\eta}_t, \\ \boldsymbol{\eta}_t &\sim N(0, Q_t). \end{aligned} \tag{5}$$

- Finally, after generating the forecast $\hat{\boldsymbol{\alpha}}_{t+1}$ for $\boldsymbol{\alpha}_{t+1}$, we calculate $\hat{\mathbf{y}}_{t+1}$ as follows:

$$\hat{\mathbf{y}}_{t+1} = Z_{t+1}(\hat{\Theta})\hat{\boldsymbol{\alpha}}_{t+1} + \mathbf{d}_{t+1}(\hat{\Theta}). \tag{6}$$

Random Walk - Baseline I

- It will serve as a benchmark, generating the simplest forecast.
- This model can be defined as follows:

$$\begin{aligned} \boldsymbol{\alpha}_{t+1} &= \boldsymbol{\alpha}_t + \boldsymbol{\eta}_t, \\ \boldsymbol{\eta}_t &\sim N(0, Q_t). \end{aligned} \tag{7}$$

- With that, the estimated value for $\boldsymbol{\alpha}$ in the next timestamp is given by:

$$\hat{\boldsymbol{\alpha}}_{t+1} = \boldsymbol{\alpha}_t. \tag{8}$$

VAR I

- The VAR model is one of the most used models for multi-dimensional time series.
- In this model, the P last values of the time series are used to predict the next value in a linear way.
- For this model, we differentiate the factors in order to obtain stationarity.
- We can write the VAR(P) model as the following:

$$\begin{aligned} \mathbf{\Delta}\boldsymbol{\alpha}_t &= \sum_{j=1}^{P} A_j \mathbf{\Delta}\boldsymbol{\alpha}_{t-j} + \mathbf{d} + \boldsymbol{\eta}_t, \\ \boldsymbol{\eta}_t &\sim N(0, Q), \\ \boldsymbol{\alpha}_t &= \boldsymbol{\alpha}_{t-1} + \mathbf{\Delta}\boldsymbol{\alpha}_t, \end{aligned} \tag{9}$$

where

VAR II

- A_j are fixed $m \times m$ coeffiecient matrices and $\mathbf{d}$ is a constant intercept vector. In this model, $\boldsymbol{\eta}_t$ is considered to be a white noise process with variance given by Q. For simplicity, we will consider that it follows a normal distribution as well.

- Once the parameters of the model - A_j, $j = 1, ..., P$, Q and $\mathbf{d}$ - are estimated, we can calculate the estimated value for $\boldsymbol{\alpha}$ in the next timestamp:

$$\hat{\boldsymbol{\alpha}}_{t+1} = \boldsymbol{\alpha}_t + \widehat{\boldsymbol{\Delta\alpha}}_{t+1} \tag{10}$$

LSTM

- From the LSTM architecture, we develop two different models: the Direct State Model and the Indirect State Model.
- The difference is that, in the Indirect State Model, instead of trying to forecast the state variable in the next timestamp, we try to forecast the value of the observation.
- The inputs of the LSTM are

$$\left[\Delta\alpha^*_{t-P} \quad \Delta\alpha^*_{t-P+1} \quad \cdots \quad \Delta\alpha^*_{t-1}\right] \tag{11}$$

  where P is the number of time steps used in the estimation, similarly to the VAR(P) model.
- The asterisk * indicates that the value was scaled between -1 and 1.

Filtering Approach I

In this approach, $\mathbf{y}_t$ and $\boldsymbol{\alpha}_t$ are modeled using a state-space representation. We consider $\mathbf{y}_t$ to be the output variable and $\boldsymbol{\alpha}_t$ to be the state variable.

$$\begin{aligned} \mathbf{y}_t &= Z_t(\Theta)\boldsymbol{\alpha}_t + \mathbf{d}_t + \boldsymbol{\epsilon}_t \\ \boldsymbol{\epsilon}_t &\sim N(0, H_t) \end{aligned} \tag{12}$$

$$\begin{aligned} \boldsymbol{\alpha}_{t+1} &= T_t(\Theta, \boldsymbol{\alpha}_t) + \mathbf{c}_t + \boldsymbol{\eta}_t \\ \boldsymbol{\eta}_t &\sim N(0, Q_t) \end{aligned} \tag{13}$$

for $t = 0, ..., n-1$, where the first equation is the measure equation and the second one is the transition equation. Notice that Z_t is a matrix that depends on Θ, while T_t is a function that depends on both Θ and $\boldsymbol{\alpha}_t$.

Filtering Approach II

- The estimation of Θ is generally given by maximization of log-likelihood.
- For the dynamic Nelson-Siegel model, we can calculate Z_t and d_t. As for T_t, c_t and Q_t, they can vary depending on the model used. Finally, Θ will also include H and the parameters for the transition equation model, which we will call Θ_T:

$$\Theta = \{\lambda, \Theta_T, H\} \tag{14}$$

- We will consider two different transition equation models for the dynamic Nelson-Siegel. The first one is the VAR(1), in which T_t is a linear function.
- And the second one is the LSTM, in which T_t is the output of an Indirect State Model.

Kalman Filter with VAR I

- In Kalman Filter with VAR the model can be estimated using the Kalman Filter algorithm.
- We choose to model the level factor as first differences, in order to make it stationary.
- In order to write the model with ΔL_t instead of L_t, we need to use a different specification for $\boldsymbol{\alpha_t}$, as written below:

$$\boldsymbol{\alpha}_t = \begin{bmatrix} L_t \\ L_{t-1} \\ S_t \\ C_t \end{bmatrix} \quad \text{and} \quad \boldsymbol{\alpha}_0 = \begin{bmatrix} L_0 \\ L_0 \\ S_0 \\ C_0 \end{bmatrix}. \tag{15}$$

Kalman Filter with VAR II

- The transition matrix T_t is given in Equation (16) below and the vector c_t is a vector with zeros:

$$T_t = \begin{bmatrix} 1+\phi_1 & -\phi_1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & \phi_2 & 0 \\ 0 & 0 & 0 & \phi_3 \end{bmatrix}, \tag{16}$$

where ϕ_1, ϕ_2 and ϕ_3 are parameters to be estimated. As we have modified $\boldsymbol{\alpha_t}$, we must also modify Z_t, by including a column of zeros in the second column.

Kalman Filter with VAR III

- Finally, for the matrix Q_t, we write:

$$Q_t = \begin{bmatrix} \sigma^2_{\eta_1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma^2_{\eta_2} & 0 \\ 0 & 0 & 0 & \sigma^2_{\eta_3} \end{bmatrix} \tag{17}$$

- Hence, $\Theta_T = \{\phi_1, \phi_2, \phi_3, \sigma^2_{\eta_1}, \sigma^2_{\eta_2}, \sigma^2_{\eta_3}\}$.

Kalman Filter with LSTM I

- The LSTM Kalman Filter (LSTM-KF) is the most complex model that will be considered. It combines an LSTM Deep Learning model with a Filtering framework in order to simultaneously estimate the state variables and the observations.
- We basically consider the transition function $T_t(\Theta, \boldsymbol{\alpha_t})$ to be the output of a Indirect State Model LSTM.
- As the transition equation is not linear, we use the Unscented Kalman Filter to estimate it.
- The diagram for the LSTM-KF algorithm is displayed in Figure 8 below.

Kalman Filter with LSTM II

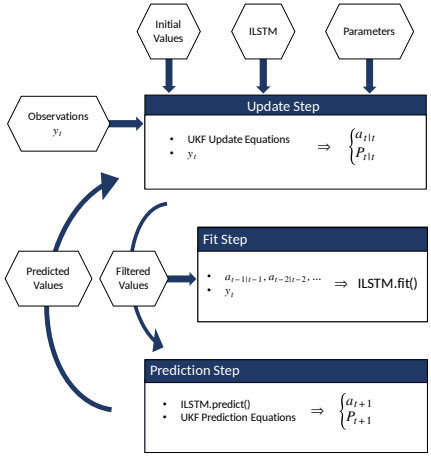


Figure: Diagram of LSTM-KF

Basic Model I

- The Basic Model does not use factors to make forecasts, hence has high dimension - from the prices and maturities.
- We want to test whether using hidden factors is in fact beneficial.
- The difference is basically the input:

$$X_j^{(t)} = \begin{bmatrix} \Delta \mathbf{y}^*_{t-P+j,\tau_1} \\ \Delta \mathbf{y}^*_{t-P+j,\tau_2} \\ \vdots \\ \Delta \mathbf{y}^*_{t-P+j,\tau_p} \\ \tau_1^* \\ \tau_2^* \\ \dots \\ \tau_p^* \end{bmatrix} \tag{18}$$

Results I

We then compare the total train RMSE for Brent and WTI for each maturity

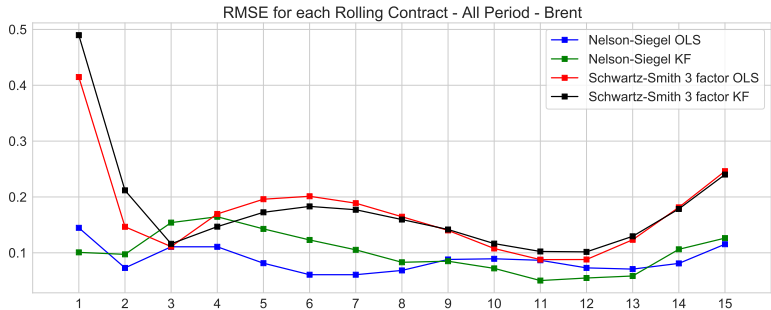


Figure: Fit RMSE for each Rolling Contract - All Period - Brent

Results II

WTI has shown a very similar behavior to Brent

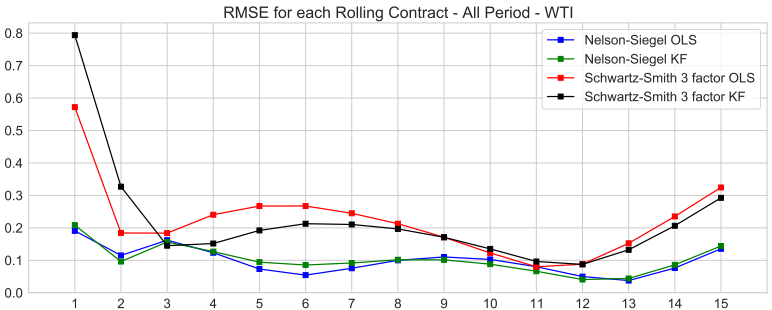


Figure: Fit RMSE for each Rolling Contrac - All Period - WTI

Forecasting I

We first show the Validation and Train RMSE for Brent with the VAR method to illustrate the method for choosing hyper-parameters to avoid over-fitting.

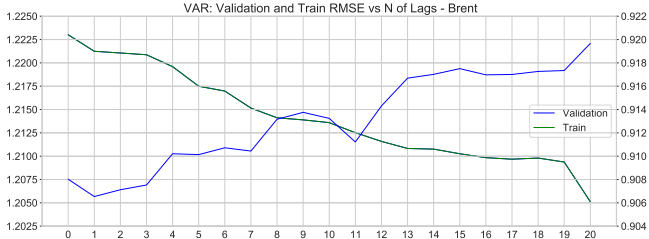


Figure: VAR over-fit: the train RMSE decreases monotonically, whereas the validation RMSE starts to increase with a higher number of lags - Brent

Forecasting II

Below we display the total relative RMSE results for Brent for each method of forecasting used.

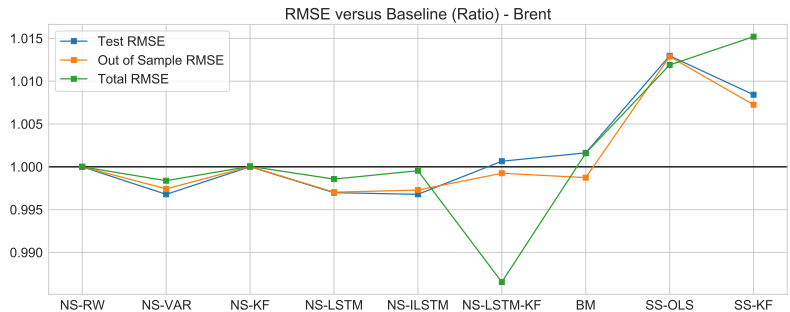


Figure: RMSE versus Baseline (Ratio) - Brent

Forecasting III

We also display the Test RMSE for each maturity for all the models except from the Schwartz-Smith ones.

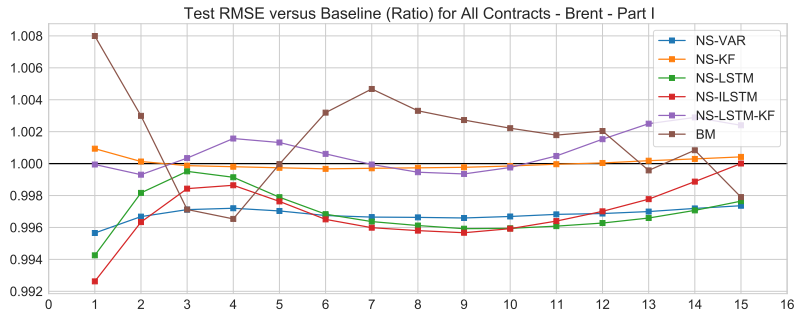


Figure: Test RMSE per Rolling Contract versus Baseline (NS-RW) - Brent

Forecasting IV

Below we display the total relative RMSE results for WTI for each method of forecasting used.

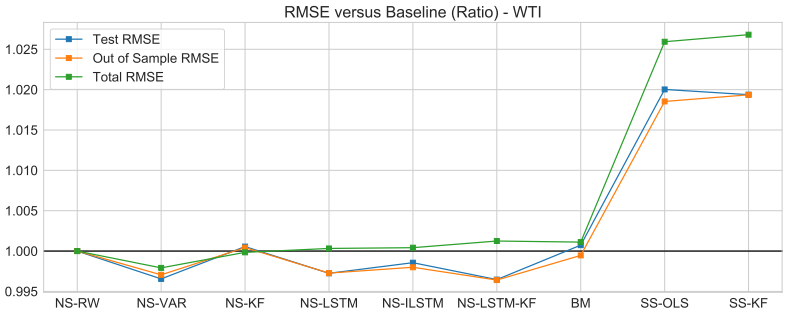


Figure: RMSE versus Baseline (Ratio) - WTI

Forecasting V

We also display the Test RMSE for each maturity for all the models except from the Schwartz-Smith ones.

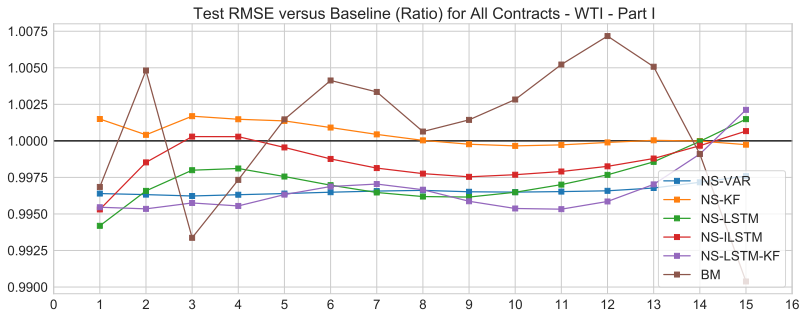


Figure: Test RMSE per Rolling Contract versus Baseline (NS-RW) - WTI

Conclusions

- The dynamic Nelson-Siegel was better at both fitting and forecasting the price curve for the two commodities.
- It was made clear that it is difficult to gain much advantage over the Random Walk in forecasting the curve prices.
- VAR(1), LSTM and ILSTM were the overall best models when considering RMSE and MAE for Brent and WTI, but also the robustness to parameter variation.
- As for the LSTM-KF, it has shown decent results and is certainly promising.
- Furthermore, we have also shown that converting the price curve into factors before the prediction step is certainly advantageous.