GaR Tool: Technical Appendix[[1]](#footnote-2)

Overview

The GaR Excel tool is based on a Python backend[[2]](#footnote-3) which carries out estimations, optimization, distribution fit, simulations and plots.[[3]](#footnote-4) The Python codes are available in the subfolder GaR, packaged with the Excel spreadsheet; the Excel spreadsheet communicates with Python through a VBA macro interfaced with a dedicated Python package, “Xlwings”, which communicates Excel inputs to Python and retrieves Python output back to Excel. The full GaR approach comprises 5 steps:

1. Partition: Partition macrofinancial variables into financial indexes to reduce parameters dimensionality and screen financial information between market trends and idiosyncratic shocks; also, chained-partitions can be computed to mitigate attrition issues in the financial sample (see below).
2. Quantile regressions: Estimate the forecasting specification of future growth, xx-periods ahead on current macrofinancial conditions, via quantile regressions, for a given set of quantiles
3. Tskew fit: Based on the conditional quantiles estimated for a given date, fit a Tskew distribution, which is the parametric future growth distribution, conditional to a given set of macrofinancial conditions. Derive the growth at risk at a given threshold.
4. Historical distributions: the same process can be applied through time, rolling the estimation window and deriving the time-series of the growth at risk, as well as certain parameters of interest (variance, skewness, kurtosis)
5. Counterfactual scenarios: using the same coefficients of the quantile regressions estimated in step 2, this step derives counterfactual distributions based on simulated macrofinancial conditions.
6. Macrofinancial Variables Partitioning Using Dimensionality Reduction

*Partitioning macrofinancial variables: general principles*

The individual macrofinancial variables are aggregated into each category (for instance price of risk, leverage and external conditions) using either unsupervised (principal component analysis - PCA) or supervised (linear discriminant analysis - LDA) dimensionality reduction methods. Technically, this step is not necessary for estimating Growth at Risk, and the tool accommodates regressions based on raw variables.[[4]](#footnote-5) However, partitioning the variables before running the model has useful advantages: first, it allows to reduce dimensionality parameters, as traditionally used in the literature; this is even more important with macro data at the quarterly level, with a limited number of observations. Second, using partitioned data often improves the forecasting estimations by extracting common trends in financial variables, hence filtering information from idiosyncratic noises. Individual financial variables, especially in countries with illiquid markets, often exhibit noisy behavior and erratic volatility. However, co-movement across a sufficiently large number of financial variables often contains valuable information. Finally, it gives the possibility to compute chained-index partitions, which mitigates attrition issues in financial samples.

The literature often uses PCA-based methods to estimate FCIs.[[5]](#footnote-6) The tool gives the possibility to aggregate variables using a PCA within each partition. Note that by default the tool returns only the first component of the PCA for a given group of variables. For instance, if the user supplied into the “price of risk” partition the term spread, corporate spread, ted spread and sovereign spread, then the “price of risk” will be the first component of the PCA of the four spreads mentioned above. The process is repeated for each partition defined by the user in the “Partition groups” spreadsheet. The way the tool works is not standard in the literature: often, econometricians estimate PCA on the full set of variables and extract the first few components, for instance the first, second and third. Doing so is optimal from a statistical perspective, as it allows to maximize the variance decomposition and deliver orthogonal components. However, it is often the case that the first few components of a PCA are difficult to interpret, as the same set of variables can be repeated with different weights. Instead, extracting the first component from an ad-hoc set of variables ensures that each component has a straightforward economic interpretation.

The tool also provides the user with the possibility to use LDA (linear discriminant analysis) instead of PCA to compute the partitions, by extracting the first component of a LDA projection applied to each subgroup. One of the issues of the PCA approach is that data reduction is realized via a maximization variance principle among the set of individual variables, which might not be relevant information for predicting future GDP growth. The tool provides the user with the possibility of partitioning variables using a supervised data reduction method, more precisely Linear Discriminant Analysis (LDA). The goal of LDA is to project a dataset onto a lower-dimensional space with good class separability: LDA maximizes the common variance among a set of variables X – as a PCA does – but also ensures that the linear combination of the variables X discriminates the class of a categorical variable y. In the framework, the *y* variable is a dummy (two classes), defined at the country level: y equals 1 when future GDP growth at one-year level is below the xxth historical country-specific percentile and 0 else (the input spreadsheet gives the possibility to choose the level on which the variable is discretized from). In other words, the loadings of each individual variables on the LDA component maximizes the separability between low and normal growth regimes. As in a PCA, the input variables are normalized (through z-score) to avoid distortions due to heterogeneity in variables scale.

#### Formal derivation of the linear discriminant analysis[[6]](#footnote-7)

From a matrix of observations X of size (n\*c), composed of individual variables divided among classes defined by the categorical variable y taking values in a discrete set C= {1,…c},[[7]](#footnote-8) the within-class and between-class scatter matrices are defined respectively as:

1. where is a scatter matrix for every class, where is the mean vector of variables , for each category of y (in the case of the tool, low versus normal growth regime);

where *m* is the overall mean, and the sample size of each of the class.

Intuitively, the within scatter matrix represents the variability intra each class (low and normal growth regime), while the between scatter matrix represents the variability between each class. An appropriate classification strategy should therefore minimize the within variance (so that the groups are homogenous within themselves) while maximizing the between variance (so that the different groups are clearly discriminated among each other) in the new subspace composed of a projection of the features .

As in a PCA, the axes of the new feature subspace are given by the eigenvectors of the decomposition of the square matrix . Intuitively, the eigenvectors and eigenvalues measure the distortion of a linear transformation (here the projection onto the reduced subset); the eigenvectors are the direction of this distortion and the eigenvalues are the scaling factor that describes the magnitude of the distortion. The tool only uses the first components of the LDA as an explanatory variable to reduce parameter noise, so that the eigenvector associated with the largest eigenvalue is considered, given that the largest eigenvalues represents the axe bearing the most information about the features subset.

The projection matrix *P* (dimension *c\*1*) is therefore the eigenvector matrix associated with the most informative eigenpair for each class, and is used to transform the samples onto the reduced subspace *R*:

Where X is the matrix of observationsof dimension (*n\*c*), *P* the projection matrix (dimension *c\*1*) and R are the transformed observations onto the reduced subset (dimension *n\*1*, as the tool only considers the first largest eigenvector).

|  |
| --- |
| **Figure 1. Conceptual Differences Between PCA and LDA** |
| Source: sebastianraschka.com. |

#### Advantages of linear discriminant analysis

The linear discriminant analysis is very similar to the principal component analysis, as both these methods reduce the data dimensionality to summarize the information contained in a large set of variables into few components. Compared to PCA however, LDA also adds the classification approach which allows the tool to link financial variables with GDP growth in the data reduction process, while the PCA approach only aggregates information about common trend among financial variables.

LDA assumes i.i.d and normally distributed data, with homoscedastic variance among each class. However, LDA is considered as robust when these assumptions are violated: *“linear discriminant analysis frequently achieves good performances […], even though the assumptions of common covariance matrix among groups and normality are often violated.”[[8]](#footnote-9)*

***Interpretation***

The tool delivers two outputs: the partition itself, as well as the loadings associated with the partition. It is extremely important that the user takes time to check on whether each partition makes economic sense, based on his own understanding of the country economic history and recent developments. As explained above, for many countries, financial markets are rather illiquid and exhibit sometimes erratic behavior. It is important to make sure that the most important crisis episodes are captured by the partitions and the main trends make sense. Else, the user should either add or remove variables, try different partitions, look at different time periods, etc.

Concerning the loadings, as explained above, they represent the coefficients of the linear combination of variables used to generate the partition. As such, they should be interpreted as a whole and not separately: the loading of a given variable will also depends on whether other variables are included or not. This is why sometimes some coefficients might appear counter-intuitive, with correlations going in the wrong direction: it might be the case that other highly correlated variables in the PCA sample might already capture the main trend and correlation between the group of variables and the partition. Hence, as it is often the case in machine learning techniques, multi-collinear variables might have significant loadings as some of the variables might “load” more on the main trend while some other variable will load in a different way. Loadings should first be interpreted in absolute terms, as they advise about the most informative variables for the partition.

#### Retropolated partitions: the use of chained-index to mitigate attrition issues

#### For both PCA-based and LDA-based partitions, the tool provides the opportunity to create “retropolated” partitions, which are basically a chain-indexed partition from the original PCA or LDA partitions. In many countries, financial variables are not available at the same time, hence creating severe attrition issues: for instance, term and sovereign spreads are often available in the 1980s, while variables based on more sophisticated instruments, such as CDS, might only be available in mid-2000s. Hence, if the user wishes to estimate a PCA based on the full set of variables since the 1980s, it will have missing values in the sample corresponding to the dates when some variables are not yet available. Neither PCA nor LDA can accommodate missing values, as they are based on variance decomposition. Some authors in the literature (for instance Koop and Korobilis 2014) circumvent this issue by arbitrarily inputting “0” to the missing values and estimate the PCA on the entire time-series. Doing so creates major issues, as the variance-covariance matrix is severely biased – in particular when the attrition rate is high (variables only being available recently, hence inputting a large number of “0”). Instead, the tool adopts another approach based on chained-index partitions.

From the original sample of variables , defined as the imbricated set of variables such that each set contains all the variables available for a given period: where *C* is the number of cut-off points. For instance, if 3 variables are available from 1980 to 1995, and 2 supplementary variables are available starting 1995, and then 4 new variables arrive in 2002, then three subsets are defined as: with 3 variables from 1980 to 1995, from 1995 to 2002 with 5 variables; and finally from 2002 to the most recent date with 9 variables. Doing a backward recursive estimation, the retropolated partition is a chained-index, computed recursively as follows:

* for the last period where all the variables are available
* From the immediately preceding set , the tool computes the recursive first difference[[9]](#footnote-10) starting from the last value of the set . Then it applies the first difference backwards from the first value of towards the first value of . Therefore, where represents the backward first difference derived from set and applied to the first value of set
* Reapply the same principle for the set , using as the starting point the first value of the retropolated computed at the previous step , and continue up to the first set.

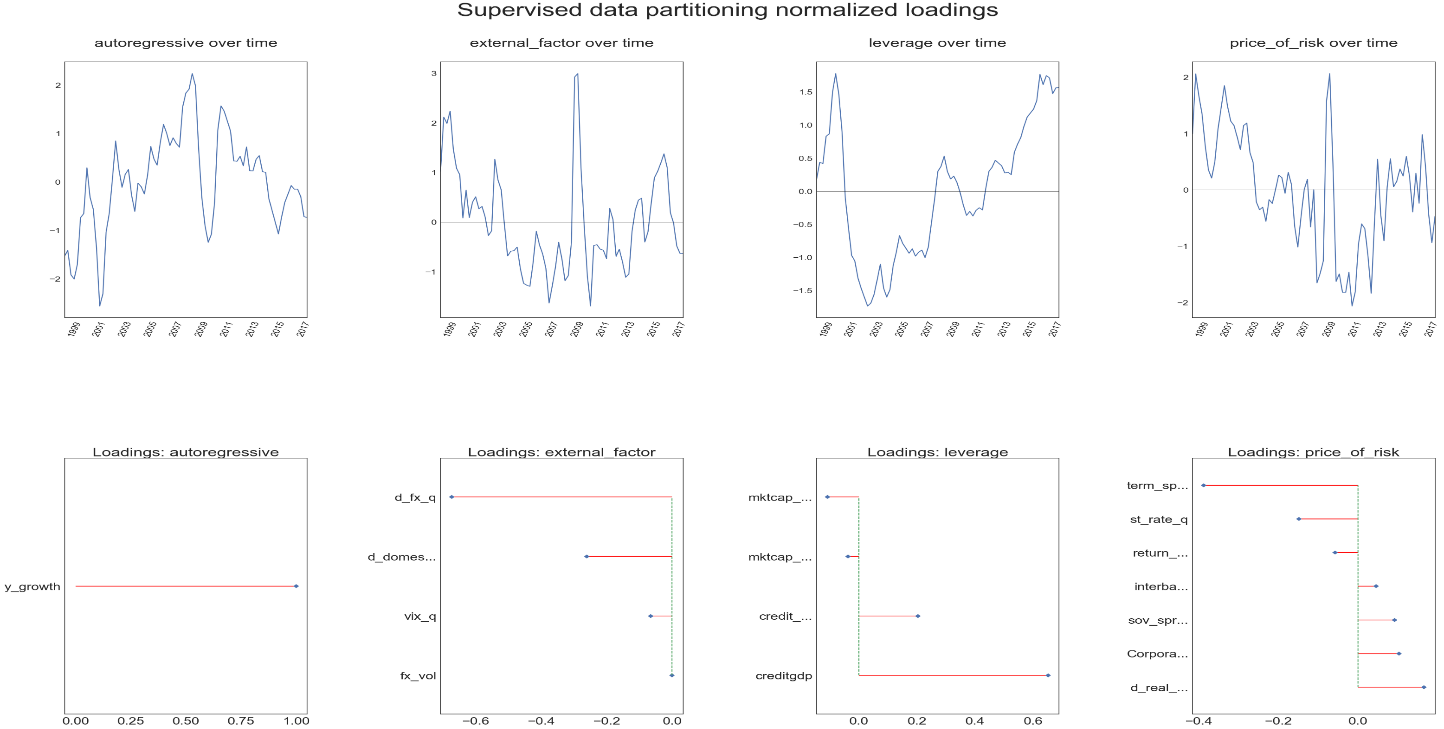
The recursive definition of the retropolated partition is therefore:

**Initial state:**

**Recursive value:**  with the operators defined above**.** Note that the tool accommodates any number of cutoffs when running the partition.

The chained-index partition tackles the problem of missing values at the beginning of the sample. For missing values at the end of the sample (for instance, if credit growth is only available with a one-period lag), the tool propagates the value forward (hot deck replacement) if the user wants to compute the partition over the entire time frame. This is a minor issue as usually only one or two observations are missing. If the user wishes to avoid this behavior, he should simply define the time range of the sample such that it ends when all the variables are available.

***Figure 2. Output Example: Partitions and Loadings***



1. Quantile Regressions and Quantile Coefficients

***Dependent variable***

The tool has been originally designed to estimate growth at risk: hence it estimates a growth rate for GDP, at different horizons. The user must provide the real GDP variable in level (for instance), and the tool will automatically compute the GDP growth rate depending on the horizon chosen by the user. The tool offers two different ways to compute GDP growth: either an annualized quarterly compound growth rate between period *t* and period *t+h*, where *h* is the horizon defined by the user, or the year-on-year growth rate, *h* periods ahead. Finally, the tool also offers the possibility to keep the variable in level in case some users wish to use their own growth rate calculations, or simply want to use another variable as the dependent variable and don’t want to compute a growth rate.

***Quantile regressions***

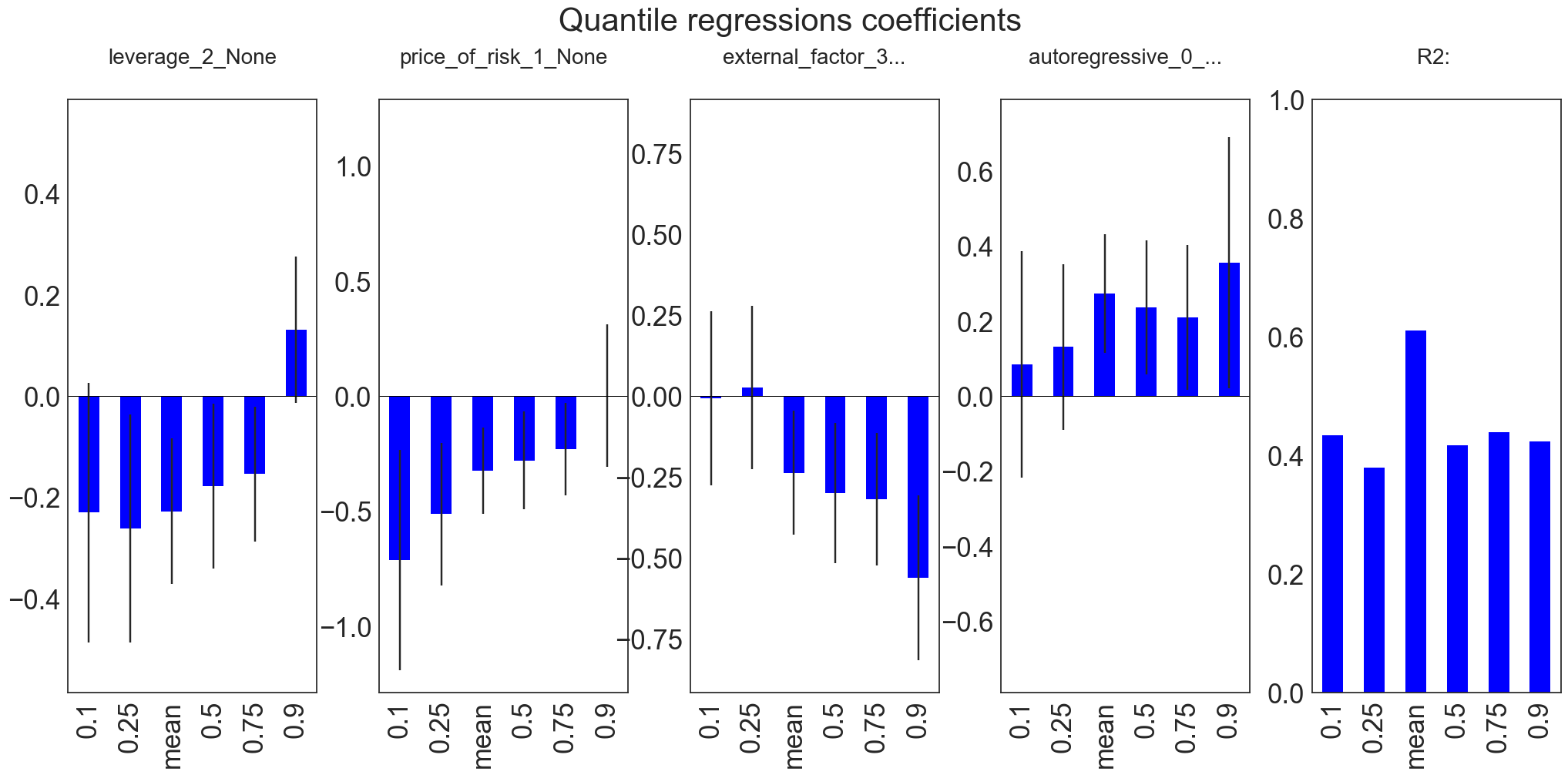
Using partitioned data instead of the original set of variables reduces the dimensionality of the regressors and therefore minimize parameters noise in the estimation. For a set of horizons where *h* represents the quarters ahead, the following specifications are estimated:

|  |
| --- |
| **Figure 3. Fitted OLS Curve vs. Fitted Quantile Regressions Curves** |
| Source: IMF staff |

Where represents future growth *h* quarters ahead, is the partition *i* (for instance price of risk, leverage or external factor), the coefficient of the quantile regression, the associated constant and the residual. The quantile regressions are estimated at different points of the distribution of , , by default. Each beta coefficients represents the macrofinancial linkage between the variable and future growth, at different points of the distribution of GDP growth (basically, the business cycle). The tool gives the choice to estimate the quantile regressions for more quantiles; however, for the extreme left and right quantiles, the data sample should be large enough to provide a reasonable fit. Note that for each quantile, the tool runs two quantile regressions: the first one with the variables as they stand and the second one where both the regressors and dependent variables have been z-scored, hence estimating normalized coefficients. The tool provides both estimated coefficients in the excel spreadsheet but plot the chart with normalized coefficients, to make them comparable and ease the interpretation. However, the original coefficients (non-normalized) are used to compute the conditional quantiles, so that the projected growth rate is at its original scale.

The tool also reports the confidence interval for the coefficients at the 10% level, using heteroskedastic robust standard errors for quantile regression (cf. Koenker 2005).

Finally, the tool also gives the possibility to transform the partitions before running the regressions: for instance, it is possible to add a leverage partition, as well as the lagged value, the moving average, first difference, etc. Users should keep in mind though that they should keep the model as parsimonious as possible to mitigate estimation errors.

***Figure 4. Output Example: Quantile Regressions Coefficients*** 

1. **Parametric fit of the conditional distribution of future GDP growth**

***Conditional quantiles***

Like an OLS regression provides an estimation of a **conditional mean**, quantile regression estimates **conditional quantiles** of the dependent variable , conditional on financial variables , for a given date *t*,[[10]](#footnote-11) based on the point estimates of the coefficients and

Using quantile regressions to estimate the conditional distribution has many advantages: first, under standard assumptions, quantile regressions provides the best unbiased linear estimator for the conditional quantile; second, quantile regressions are robust to outliers, which are frequent when dealing with countries with poor data coverage. Finally, the asymptotic properties of the quantile regression estimator are well-known and easy to derive.

The value of each conditional quantiles, estimated for a given date *t* is given in the “conditional quantiles” spreadsheet. It is important to stress that at this stage, growth at risk can be directly estimated from the conditional quantile at a given level (5%, 10%, etc.) providing that the level has been estimated in the quantile regression. However, as explained below, the tool provides a parametric fit of the distribution, from which the growth at risk is computed, to mitigate issues related with extreme tail quantiles and quantiles crossing.

***Parametric fit***

The conditional quantiles are a sufficient statistic for describing the conditional cumulative distribution function (cdf). From the cdf, deriving the probability distribution function (pdf) can be done in two ways, using either a parametric or a non-parametric fit. Although relying on very few assumptions, non-parametric fits are very sensitive to estimation noise, and in particular to quantiles crossing, when estimating quantiles regressions over a thin grid of quantiles. Also, because data coverage is limited, estimation of the extreme quantiles (5 percent and 95 percent) can be inaccurate, leading to large confidence error bands which might give inconsistent distribution fit on the tails.

For the sake of robustness with regards to quantiles crossing and extreme quantiles estimation, the tool uses a parametric method to fit the conditional quantiles estimated in the first step. Following Adrian et al. (2018), a parametric t-skew fit is used. Student t-distributions have been used extensively in financial econometrics for their good properties to represent more accurately fatter tails, as Student t-distributions have asymptotic Paretian tails. The skew version of the t-distribution (as in Azzalini 2003) has been proven useful to model tail events, as many distributions in finance are indeed skewed (cf. Andersen et al. 2001).

Fitting the cdf estimated from the quantile regressions represents another robust dimensionality reduction,after the data partitioning presented above. The Student t-skew distribution is fully characterized by 4 parameters (location, degree of freedom, scale and skewness) which represents a good compromise between describing the distribution as accurately as possible and keeping the number of parameters low to avoid over-fitting. Hence it is a parsimonious, yet rather exhaustive way, to summarize the information about variance (volatility), skewness and kurtosis contained in the sample.

***TSkew distribution***

Giot and Laurent (2002) proposes a closed form expression of both the quantile, cumulative distribution and probability functions of the t-skew, based on a reparametrization of Fernandez and Steel (1998) specification around the skewness parameters, via step functions. For instance, the non-standardized quantile function (also called the percentage point function or ppf) of the t-skew distribution is given by:

Where represents the quantile, df the degree of freedom, the skewness parameter (always positive, a value below 1 indicates skewness on the left and a value above 1 skewness on the right).

The standardized version of the t-skew percentage point function (ppf) is then straightforward to derive:

where *loc* is the location (the mean) and *scale* the variance of the distribution.

***From the empirical cdf to the Tskew parameters: unconstrained optimization***

The tool offers to estimate the t-skew distribution parameters in two different ways: either unconstrained in the location or with location constraint.

The unconstrained optimization derives the parameters of interest by minimizing the distance between the empirical quantiles and the quantiles of a Tskew:

Where represents the quantile of the Tskew with parameters (loc, df, scale, skew).

Because both the scale and the skewness parameters are naturally bounded, a Sequential Least Squares Programming (SLSQP) algorithm is used to perform the minimization program under the constraint that both these quantities should be strictly positive. Note that at this stage the tool is weighting each quantile equally; it could also be possible to improve the fit on the lower quantiles by weighting deviation on the lower quantiles more than on the rest of the distribution, to improve the crisis fit.

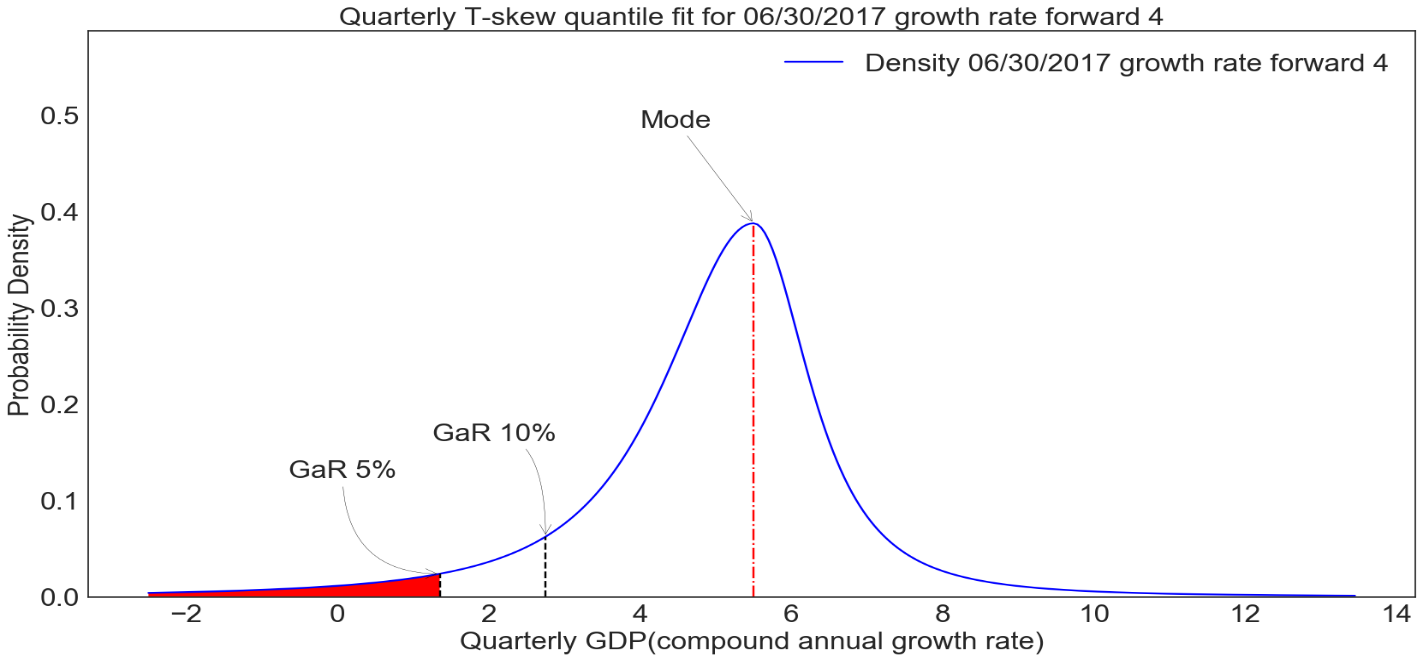
Once the optimal t-skew parameters have been estimated from the conditional quantiles, it is straightforward to derive the fitted t-skew cdf and pdf, therefore allowing to estimate both the associated Value at Risk (VaR) and Expected Shortfall (ES).

***From the empirical cdf to the Tskew parameters: location-constrained optimization***

## The tool also gives the possibility to fit the Tskew distribution by constraining the location (the mode) of the distribution. This option is useful if i) there is a more accurate forecast for the mode available or ii) to make the density forecast consistent with other point forecasts (for instance the IMF WEO forecast). In this case, the tails, skewness and kurtosis should be consistent with both the conditional quantiles from the quantile regressions, as well with the constrained location. Then the distribution fit must be optimized under the constraints, and not simply by translating the original distribution linearly. The optimization program is therefore ran on a 2-parameters space:

## Where is provided ex-ante by the user.

## Note that imposing the mode can have a large impact on the fitted Tskew distribution, in particular if the constrained location differs significantly from the unconstrained location corresponding to the conditional quantiles. In this case, the distribution will have to adjust its variance and skewness to meet the mode: hence, an optimistic ad-hoc mode (larger than the unconstrained mode) will result in inflated downside risks and vice-versa for pessimistic constrained mode. Hence, the GaR tool can also be used as a “reality-check” to put into perspective the results of a point-forecast model.

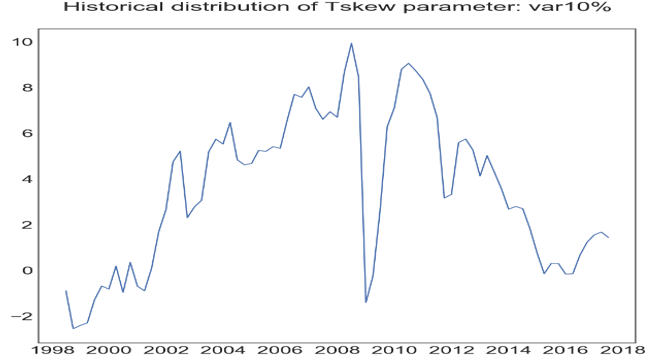
***Figure 5. Output Example: Tskew Fit and GaR at 5- and 10-percent*** 

1. **Parameters Historical Distribution**

The fourth step is optional: it gives the possibility to re-run the Tskew fit across time. The distribution fit is done with unconstrained parameters (else it would require the user to the entire vintage series of mode forecasts over the entire time horizon) based on the same approach as in step 3. However, as explained above, the conditional future growth density forecast depends on two sources of information: the beta coefficients from the quantile regression and the set of regressors from which the quantiles are computed upon. To make the GaR estimates easy to compare across time, and to avoid estimating quantile regressions on very limited sample (especially at the beginning of the time series), the tool uses the same quantile coefficients as the ones obtained from step 2, using the entire time-series. Hence, the only source of heterogeneity is in the regressors.

Technically, this approach assumes that there are no structural breaks in the data and therefore the quantile estimator is asymptotically consistent: by using the full time-series to estimate the beta coefficients, the tool assumes that the estimated beta coefficients will converge to the true “a-temporal” value, as the sample size increases. This is a reasonable assumption given the limited sample size in macro time series. However, should the user wish to recover the same beta as would be available to a forecaster in the past, it can simply restrict the input sample to a particular point in time.

***Figure 6. Output Example: Historical Time-Series of GaR at 10-percent***



1. **Counterfactual Scenarios Analysis**

Finally, the tool offers the possibility to run counterfactual scenarios analysis, in a static way. It simulates the impact on a shock of the raw variables on the future growth distribution: it is a comparative static exercise in the sense that other variables are held constant, the shock happens *ceteris paribus*. The tool gives the possibility to shock one or multiple variables, either the raw variables or the partitions. It provides two ways of shocking the variables: either in standard deviation term, or in percentage.

## *Shock on the partitions*

## This is the easiest case. Using the quantile regressions estimated in step 2, the tool estimates conditional quantiles based on counterfactual partitions = )

## Note that the estimated beta coefficients are the same as in the baseline regressions, hence ensuring that the scenarios are indeed comparable. Based on the counterfactual quantiles , the tool now estimates the Tskew distribution as in step 3 and also provides the possibility to have unconstrained and constrained mode.

## *Shock on the raw variables*

## Since the quantile regressions are estimated on the partitions, and not on the variables directly, it is necessary to add an extra step to estimate how much the partition will change if one (or many) of its variables are shocked. Again, this is a comparative static analysis, so the tool assumes that other variables in the partition will remained unchanged. In this case, the shock on the partition is defined as the shock on the variable times the correlation coefficient between the variable and its partition:

= ) where is the raw variable shocked entering into the partition .

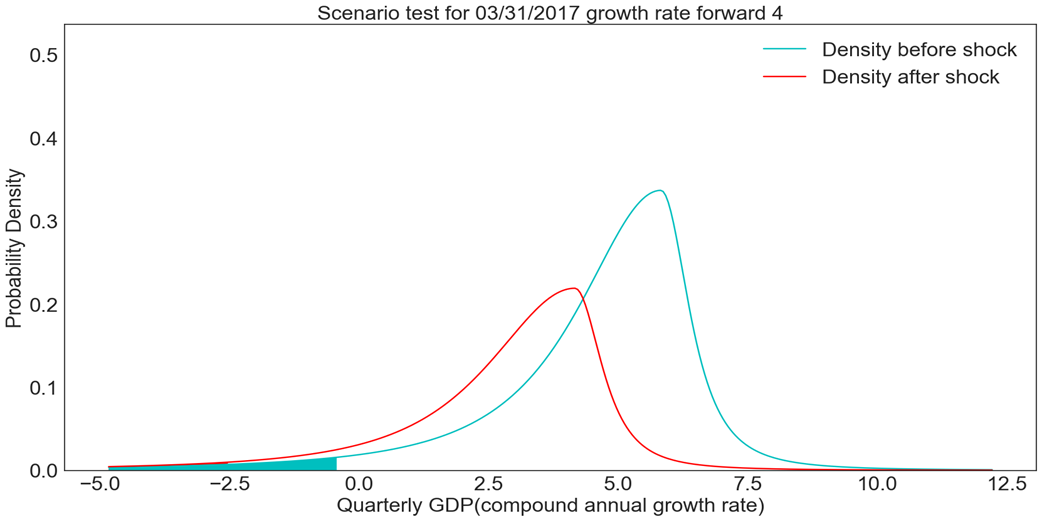
Using the correlation coefficient instead of the loading is necessary to avoid issues with the chained index partitions, where the loading of the variable and the partition changed across time.

## Then again, the conditional quantiles are estimated as follows:

And the counterfactual Tskew distribution is estimated as in step 3.

## *Interpretation*

## In both cases, the tool derives the counterfactual distribution and presents a plot with both the baseline and the shocked distributions. This approach is interesting as it uncovers the impact of non-linearities (different beta for each quantile) on the future growth distribution: the shock therefore propagates differently on different points of the distribution, for instance with different impacts on the tails and on the belly of the curve, hence delivering richer policy interpretation.

***Figure 7. Output Example: Counterfactual Scenario Analysis***

## References

Adrian, Tobias, Nina Boyarchenko, and Domenico Giannone, 2018 “Vulnerable Growth.” *American Economic Review,* forthcoming

York Staff Report 794. Azzalini, Adelchi and Antonella Capitanio. 2003. “Distributions Generated by Perturbations of Symmetry with Emphasis on a Multivariate Skew t-distribution.” *Journal of the Royal Statistical Society: Series B*, 65: 367-89.

Duda, Richard O., Peter E. Hart, and David G. Stork. 2001. *Pattern Classification*. Wiley-Interscience*.*

Fernandez, Carmen, and Mark FJ Steel. 1998. “On Bayesian Modelling of Fat Tails and Skewness.” *Journal of the American Statistical Association*, 93 (441): 359–71.

Giot, Pierre and Sébastien Laurent. 2003. “Value-at-Risk for Long and Short Trading Positions.” *Journal of Applied Econometrics* 18 (6): 641–64.

Izenman, Alan J. 2013. *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning*. Springer-Verlag.

Koenker, Roger. 2005. *Quantile Regression*. Cambridge University Press.

Koop, Gary and Dimitris Korobilis. 2014. “A New Index of Financial Conditions.” *European Economic Review* 71: 101–16.

Lambert, P. and S. Laurent. 2002. “Modelling Skewness Dynamics in Series of Financial Data using Skewed Location-Scale Distributions.” Discussion Paper, Institut de Statistique, Université Catholique de Louvain.

1. This version: September 2018. Drafted by Romain Lafarguette. Technical questions and clarifications about the empirical strategy should be addressed to: [rlafarguette@imf.org](mailto:rlafarguette@imf.org); questions about the Excel interface should be addressed to Changchun Wang at cwang2@imf.org [↑](#footnote-ref-2)
2. The Python version should be at least 3.5. The tool automatically manages paths and packages; the easiest way is to install the most recent Python Anaconda distribution, which contains by default all the packages needed for the GaR tool. For IMF users, the software center already has a full-fledged Anaconda distribution available which fits all the requirements of the GaR tool. [↑](#footnote-ref-3)
3. Plots are available in the Excel spreadsheet, and also in the dedicated subfolder “Figures” [↑](#footnote-ref-4)
4. Simply by creating a partition with only one variable in it: the tool will keep it unchanged. [↑](#footnote-ref-5)
5. Some papers have been using more advanced technics: for instance, the FCIs computed in chapter 3 of the GFSR April 2017 were derived from a FAVAR model (cf. Koop and Korobilis 2004) using Kalman filtering, which tries to capture the joint dynamic of GDP and FCI, over 20 variables; however, FAVAR-derived and PCA-derived FCIs are very similar , as the correlation between the two across time for 43 countries was above 95% [↑](#footnote-ref-6)
6. A thorough exposition of the linear discriminant analysis can be found in Izenman, *Modern Multivariate Statistical Techniques: Regression, Classification, and Manifold Learning* (2013). A practical and intuitive guide for implementation of the LDA using Python is available on Sebastian Raschka’s excellent website: http://sebastianraschka.com/Articles/2014\_python\_lda.html [↑](#footnote-ref-7)
7. In machine learning terminology, the X set is called the *training set*. The individual variables are called features or attributes. [↑](#footnote-ref-8)
8. Duda, Richard O, Peter E Hart, and David G Stork. 2001. Pattern Classification. New York: Wiley. [↑](#footnote-ref-9)
9. Given that the PCA components are z-scored, they are often close to 0. Using growth rate would in this case give very volatile values. [↑](#footnote-ref-10)
10. The date *t* for the variables should be provided by the user. [↑](#footnote-ref-11)