Supplementary Appendix index to

"Joint Extreme Value-at-Risk and Expected Shortfall Dynamics with a Single Integrated Tail Shape Parameter"

by Enzo D'Innocenzo, André Lucas, Bernd Schwaab, and Xin Zhang

Index

Files description

Indications to replicate the empirics

File index

`appendix/`

- appendix/Supplemantary appendix Joint Extreme VaR and ES Dynamics with a Single Integrated Tail Shape Parameter.pdf

Appendix with proofs and additional results.

`Data/`

- `Data/updated data/Bitfinex_BTCUSD_1h.csv`
- `Data/updated data/Bitfinex_ETHUSD_1h.csv`

Two downloaded files containing the data for BTC and ETH as used in the paper and as downloaded early Sep 2025; with thanks to Bitfinex to make such data publicly available. The data is downloaded as is, and any copyrights to the data (if any) befall to Bitfinex.

`code/r/`

This folder contains the files to replicate the empirical results. To run them, Rcpp and `Rcpp` and `RcppArmadillo` need to be installed, along with the appropriate C++ compilers, depending on your platform. See for instance https://teuder.github.io/rcpp4everyone_en/020_install.html.

- `code/r/EVTPZC_empirics.R`

  Main file to replicate the empirical analysis.

- `code/r/EVTPZC_empirics.log`

  Result from running `code/r/EVTPZC_empirics.R` on a MacBook Pro with Apple M3 Max, 48GB memory, Total Number of Cores: 16 (12 performance and 4 efficiency), macOS Tahoe version 26.1, from the command line using RScript, or from within R-Studio. Running the code on a Windows 11 (25H2) laptop, and R version 4.5.2 (from 2025-10-31), we got similar, though not identical results. Significance and order of magnitude of Table 2 did not appear to be affected, though.

- `code/r/Integrated_EVT_filter.R`
- `code/r/Integrated_EVT_filter.cpp`

  Core routines of the the dynamic EVT approach with an integrated single time-varying parameter, as explained in the paper.

- `code/r/PZC_simulation_aid.R`
- `code/r/PZC_simulation.cpp`

Core routines for the Patton/Ziegel/Chen methodology to obtain the thresholds for the dynamic EVT approach, or to use as a benchmark model. It also contains a large number of helper functions for tables and plots.

- `code/r/TVGPD_filter.R`

  Core routines for the D'Innocenzo/Lucas/Schwaab/Zhang time-varying GPD model as a benchmark model.

- `code/r/Report_fig3.R`
- `code/r/Report_fig4.R`

  Routines to generate the figures for the paper from the main output.

- `code/r/results`

  An (initially) empty folder used to write (intermediate and final) results to. Old results are overwritten.

`code/matlab/`

This folder contains MATLAB scripts to reproduce the Monte Carlo simulation results. Each script simulates a Gaussian GARCH(1,1) return series and then overwrites observations that exceed a time-varying threshold tau_t with synthetic exceedances of the form $tau_t * \exp(f_t * \epsilon_i)$, where $\epsilon_i \sim \text{Exp}(1)$ and f_t is a time-varying tail-shape parameter. Estimation is performed using `estimate_full_model` under three EVT option settings that differ in whether the parameter ω is estimated or fixed.

- `code/matlab/MC_simulation_SC1.m` (Experiment 1 – Gaussian GARCH + externally supplied threshold):

  Simulates `yg` and conditional variance `ht` via `code\matlab\tarch_simulate.m`. Defines $\tau_t = \sqrt{(h_t)} *$ norminv(0.90). When `yg(t) > tau_t(t)`, the observation is replaced by `tau_t(t) * exp(ff * epsi(t))`. The tail state `ff` is updated only on exceedance dates via

  ```
  ff = omega_f + ff + alpha_f * ff * (epsi(t) - 1);
  ```

  otherwise it remains constant. The script estimates tail parameters and the filtered tail path `EVT_ft` under three EVT option configurations, and saves results to `code\matlab\MC_simulations_SC1_<TT>.mat`.

- `code\matlab\MC_simulation_SC2.m` (Experiment 2 – same DGP + PZC threshold estimation step):

  Uses the same data-generating process as SC1.
  Additionally runs `code\matlab\PZC_Toolbox\PZC_optimize_safe(..., @PZC_filter, ...)` to obtain a VaR series at `alpha_tail = 0.10` and stores it in `my_data.tau`. Results are saved to `code\matlab\MC_simulations_SC2_<TT>.mat`.

- `code\matlab\MC_simulation_SC3.m` (Experiment 3 – driven tail state dynamics):

  Uses the same GARCH simulation and exceedance replacement mechanism as SC1, but the tail state evolves at every time step according to

  ```
  log(ff) = omega_f + beta_f * log(ff) + alpha_f * N(0,1)
  ```

  and then `ff = exp(log(ff))`.

  The script includes the same PZC VaR computation step as SC2. Results are saved to `code\matlab\MC_simulations_SC3_<TT>.mat`.

Indications to replicate the empirics

Open the file `code/r/EVTPZC_empirics.R` and make `code/r/` the working directory when executing this file, such that the data location and the output locations make sense. Ensure that there is a folder `code/results/` where the results will be written.

First, run the file `EVTPZC_empirics.R`. Given the options in the file, it will do recursive estimation for BTC and ETH at three different levels of κ and $\gamma = \kappa/10$. All results are written to the `code/results/` folder. On screen, an output if obtained that produces a raw version of Table 2 in the paper.

Second, run the file `Report_fig3.R`. It produces a number of `.png` files in the `code/results/` folder that correspond to the panels in Figure 3.

Third, run the file `Report_fig4.R`. It produces a number of `.png` files in the `code/results/` folder that correspond to the panels in Figure 4.