Online Appendices for

Time Varying Extremes

Ulrich K. Müller and Mark W. Watson

September 2024

1 Data Appendix

1.1 Weather Disaster Damages

The data are from the *Billion-Dollar Weather and Climate Disaster* dataset described at https://www.ncei.noaa.gov/access/billions/. We use all of the events in the database except for droughts and wildfires (because the duration of these events can extend for many months). The normalized data are adjusted by the value of the real U.S. capital stock. We use the chain-type quantity index *Table 1.2. Chain-Type Quantity Indexes for Net Stock of Fixed Assets and Consumer Durable Goods* from the BEA Fixed Asset Tables. The data are annual observations from 1979-2022. We interpolate the annual series to monthly observations from 1980:1-2022:12 and extrapolate through 2023:6 using the value from 2022:12. Let A_t denote the value of the capital stock at time t and D_t denote the unadjusted value of damages. The normalized damages are $D_t^{norm} = D_t/(A_t/A_{2023:6})$. The censoring threshold for the normalized damages is $\tau_t^{norm} = \tau/(A_t/A_{2023:6})$, where τ is the censoring threshold for D_t, which is $\tau = \$1$ billion.

Figure 1 plots the damages (shown in the published article as Figure 1) along with the normalized damages and the censoring thresholds.

1.2 Returns

We begin with value-weighted daily returns from the CRSP SP500 index available from January 2, 1926 through December 30, 2022. These returns are plotted in panel (a) of Figure 2. We fit a GARCH (1,1) model using the full sample period and standardize the daily returns by subtracting the sample mean and dividing by the fitted GARCH standard deviation. These standardized values are plotted in panel (b) of Figure 2. Panels (c) and (d) (also reported as Figure 2 in the published article) are the largest and smallest returns over non-overlapping 6-month periods.

1.3 Firm and City Size

Population values from the largest 100 cities for 1900, 1940, 1980 and 1920 are from https://www.biggestuscities.com/. These are divided by aggregate U.S. population. Employment levels are from Compustat for firms incorporated in the United States. These are divided by total U.S. private employment (series USPRIV from the FRBSL FRED database).

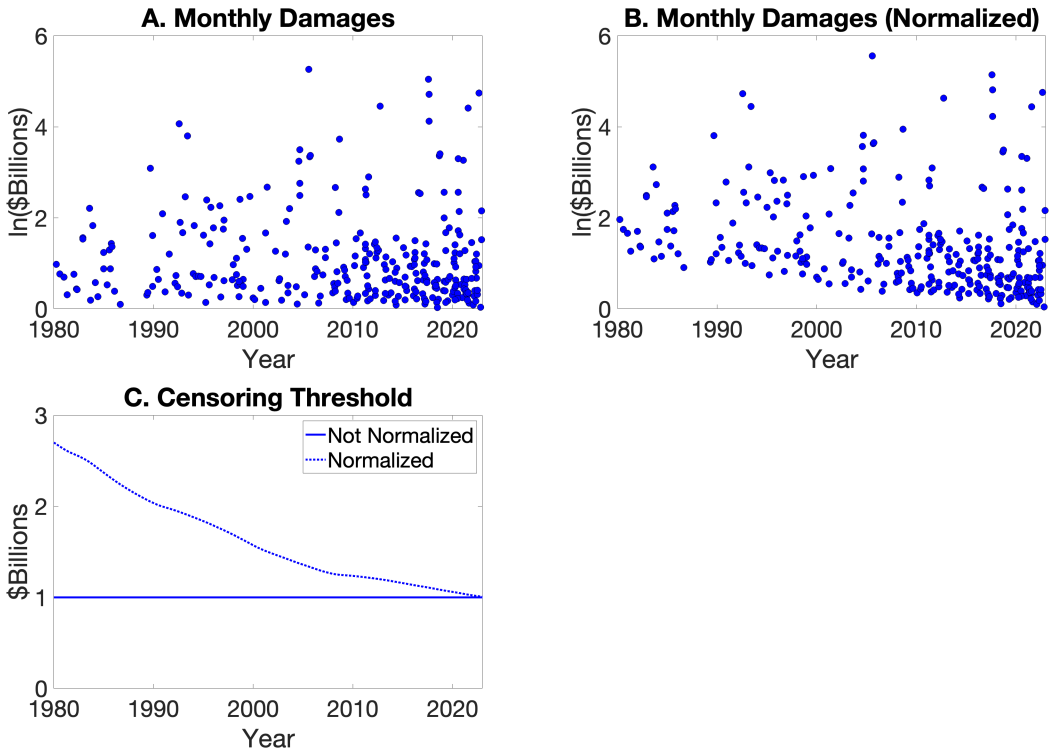

Figure 1: Damages and Normalized Damages

2 Computing the $\hat{\xi}$-adjusted LR Critical Values

2.1 LR Statistic for k Largest Observations

The distribution of the LR statistic shown in equation (6) in the published article only depends on ξ. The adjusted LR statistic depends on $\hat{\xi}$, the MLE of ξ, with $\mathrm{LR}^{\hat{\xi}\text{-adjusted}} = \exp(a_0 + a_1\hat{\xi} + a_2\hat{\xi}^2) \times \mathrm{LR}$. The goal is to find values of the parameters (a_0, a_1, a_2) to maximize $\min_{\xi_{\min} \leq \xi \leq \xi_{\max}} \mathbb{P}_\xi(\mathrm{LR}^{\hat{\xi}\text{-adjusted}} > 1)$ subject to the constraint $\max_{\xi_{\min} \leq \xi \leq \xi_{\max}} \mathbb{P}_\xi(\mathrm{LR}^{\hat{\xi}\text{-adjusted}} > 1) \leq \alpha$, where $\mathbb{P}_\xi$ denotes the probability under ξ and α is the level of the test. Throughout we set $\alpha = 0.05$. We approximate (a_0, a_1, a_2) as follows.

We subdivide the possible values of $\xi \in [\xi_{\min}, \xi_{\max}]$ into an equal spaced grid with 10 values ξ_j, $j = 1, \ldots, 10$. Here $\xi_{\min} = -0.5$ for the hypothesis concerning $q_{0.9}$, $\xi_{\min} = \varepsilon = 0.03$ for the hypothesis $H_0 : \mu = \sigma/\xi$, and $\xi_{\max} = 1.5$ in both cases. We generate $N = 10{,}000$ independent draws $(\hat{\xi}_j^{(l)}, \mathrm{LR}_j^{(l)})$, $l = 1, \ldots, N$ of the maximum likelihood estimator $\hat{\xi}$ and the LR statistic for each value of ξ in the grid. The underlying kTN independent exponential variables (cf. equation (2) in the published article) are held constant across the 10 values of ξ in these simulations.

We construct an estimate of $\mathbb{P}_{\xi_j}(\mathrm{LR}^{\hat{\xi}\text{-adjusted}} \leq 1)$ as $N^{-1} \sum_l \Phi(\frac{1-\mathrm{LR}_j^{\hat{\xi}\text{-adjusted},(l)}}{h})$, where Φ is the stan-

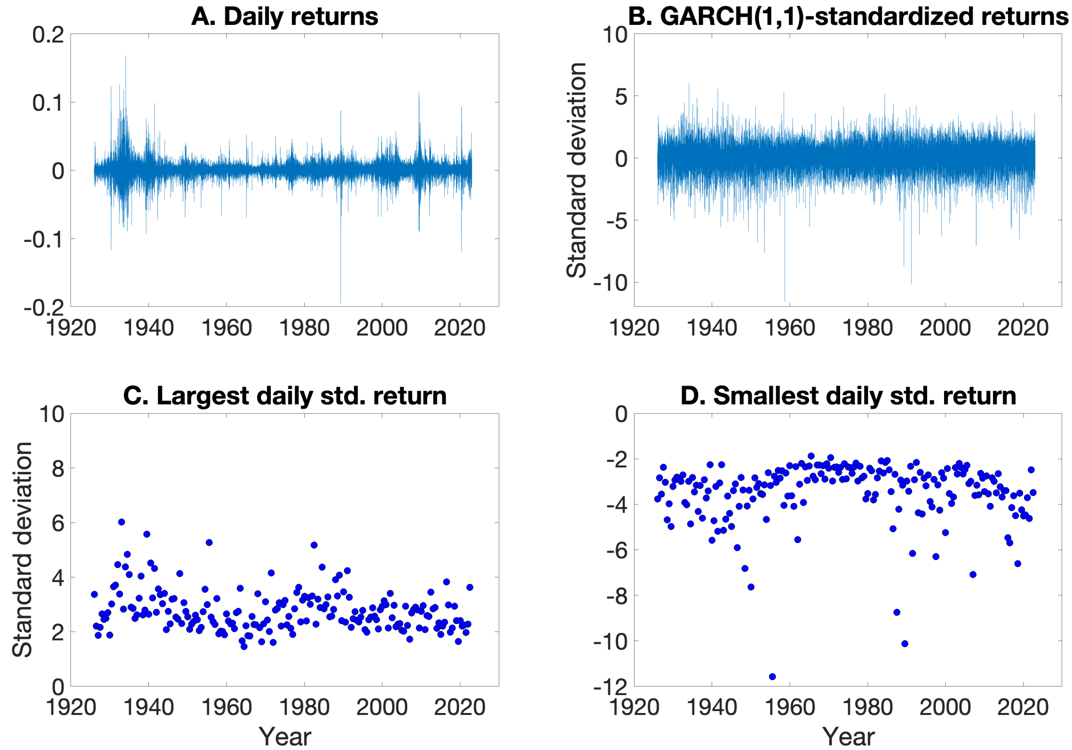

Figure 2: Daily Returns and 6-month Extremes

dard normal CDF, h is a bandwidth parameter. (This estimate is preferred to the naive estimate, $N^{-1}\sum_l \mathbf{1}[(\mathrm{LR}_j^{\hat{\xi}\text{-adjusted}} \leq 1]$ because $\Phi(\frac{1-\mathrm{LR}_j^{\hat{\xi}\text{-adjusted},(l)}}{h})$ is a smooth function of (a_0, a_1, a_2), and this facilitates numerical minimization.) We set the bandwidth h equal to 0.3 times the difference between the 97th and 93th percentile of the distribution of the unadjusted LR_j statistic for the 5th value of ξ in the grid.

The target for $\mathbb{P}_{\xi_j}(\mathrm{LR}_j^{\hat{\xi}\text{-adjusted}} \leq 1)$ is $1-\alpha$. We found it useful to smooth these probabilities using the logit transformation $\Psi(p) = \log(\frac{p}{1-p})$, and this yielded the minimization problem

$$\min_{a_0,a_1,a_2} \sum_{j=1}^{10} \ell \left\{ \Psi \left[N^{-1} \sum_{l=1}^{N} \Phi \left(\frac{1-\mathrm{LR}_j^{\hat{\xi}\text{-adjusted},(l)}}{h} \right) \right] - \Psi(1-\alpha) \right\} \tag{1}$$

where ℓ is a loss function penalizing deviations of $\Psi(N^{-1}\sum_l \Phi(\frac{1-\mathrm{LR}_j^{\hat{\xi}\text{-adjusted},(l)}}{h}))$ from $\Psi(1-\alpha)$. We used the linex-loss $l(x) = \exp(-12x) + 12x - 1$, where the asymmetry in the loss strongly penalizes overrejections.

After numerically solving (1), we adjust the value of a_0 so that the largest null rejection probability is exactly 5% on the grid. As a final check, we numerically explored potential overrejections for values of ξ that fall between grid values, and found them to be within Monte Carlo error.

2.2 LR Statistic for Exceedances

The initial level 99% confidence set for (μ, σ, ξ) is equal to

$$\mathcal{S}_0 = \{(\mu_0, \sigma_0, \xi_0) : \mathrm{LR}_{\mu_0,\sigma_0,\xi_0} < \mathrm{cv}^{0.99}_{\mu_0,\sigma_0,\xi_0}\} \subset \mathbb{R}^3$$

where $\mathrm{LR}_{\mu_0,\sigma_0,\xi_0} = \sup_{(\mu,\sigma,\xi)\in\Theta_1} \ln \prod_{t=1}^T f_{\tau_t}(\mathbf{Y}_t|\mu,\sigma,\xi) - \ln \prod_{t=1}^T f_{\tau_t}(\mathbf{Y}_t|\mu_0,\sigma_0,\xi_0)$ and $\mathrm{cv}^{0.99}_{\mu_0,\sigma_0,\xi_0}$ is the 99th percentile of the null distribution of $\mathrm{LR}_{\mu_0,\sigma_0,\xi_0}$. Our initial goal is to draw 50 points from $\mathcal{S}_0$ at random. To this end, we proceed as follows: We first compute the maximum likelihood estimator $(\hat{\mu}, \hat{\sigma}, \hat{\xi})$ and find the 99.5% level critical value $\overline{\mathrm{cv}}$ for the LR statistic of $H_0 : (\mu, \sigma, \xi) = (\mu_0, \sigma_0, \xi_0)$ where $(\mu_0, \sigma_0, \xi_0) = (\hat{\mu}, \hat{\sigma}, \hat{\xi})$. We then numerically determine $\mu_{\min}$ such that the LR statistic of $H_0 : (\mu, \sigma, \xi) = (\mu_0, \sigma_0, \xi_0)$ with $(\mu_0, \sigma_0, \xi_0) = (\mu_{\min}, \hat{\sigma}, \hat{\xi})$ is equal to $\overline{\mathrm{cv}}$. The idea here is that $\mathrm{cv}^{0.99}_{\mu_0,\sigma_0,\xi_0}$ does not vary a lot for relevant values of (μ_0, σ_0, ξ_0), so $\mathrm{cv}^{0.99}_{\mu_0,\sigma_0,\xi_0} \leq \overline{\mathrm{cv}}$, and $\mu_{\min}$ is smaller than what one would obtain had one used $\mathrm{cv}^{0.99}_{\mu_{\min},\hat{\sigma},\hat{\xi}}$. The same procedure is then applied to $\mu_{\max}$, $\sigma_{\min}$, $\sigma_{\max}$, $\xi_{\min}$ and $\xi_{\max}$, with the end result of a hypercube in $\mathbb{R}^3$ that contains $\mathcal{S}_0$. We then randomly draw values (μ_0, σ_0, ξ_0) uniformly in this hypercube, and check whether $\mathrm{LR}_{\mu_0,\sigma_0,\xi_0} < \mathrm{cv}^{0.99}_{\mu_0,\sigma_0,\xi_0}$, where $\mathrm{cv}^{0.99}_{\mu_0,\sigma_0,\xi_0}$ is obtained via simulation based on $N = 10,000$ draws, until we have collected 50 values for which we do not reject.

In the second step, we proceed just like in Section 2.1, except that the "grid" now consists of these 50 values.

3 Locally Best Test for the Null of Time Invariant Parameters with Known Values

To begin consider the following generic testing problem with $X \sim f(x|\zeta)$, $X \in \mathbb{R}^m$, with null hypothesis $H_0 : \zeta = \zeta_0$ and $H_a : \zeta \neq \zeta_0$, and interest focuses on weighted average power (WAP) for the values of ζ under the alternative using the weighting function G. By standard arguments, the best WAP test is the likelihood ratio test for $H_0 : \zeta = \zeta_0$ versus $H_a^* : \zeta \sim G$, that is, the test rejects for large value of the likelihood ratio statistic $\mathrm{LR} = \int f(X|\zeta)dG(\zeta)/f(X|\zeta_0)$. Suppose that G is the normal distribution with mean ζ_0 and variance $\tau^2 H$, so that under H_a^*, $\zeta = \zeta_0 + \tau Z$ with $Z \sim \mathcal{N}(0, H)$. With τ^2 small, $||\tau Z||$ is small with high probability, so $\ln[f(X|\zeta_0+\tau Z)/f(X|\zeta_0)] \approx \tau Z'S$, where S denotes the score $S = \partial \ln f(X|\zeta)/\partial\zeta|_{\zeta=\zeta_0}$. Thus, with τ^2 small and ϕ the density of Z,

$$\begin{aligned}
\mathrm{LR} &= \frac{\int f(X|\zeta_0 + \tau z)\phi(z)dz}{f(X|\zeta_0)} \\
&= \int \exp[\ln f(X|\zeta_0 + \tau z) - \ln f(X|\zeta_0)]\phi(z)dz \\
&\approx \int \exp[\tau z'S]\phi(z)dz \\
&= \mathbb{E}_Z[\exp(\tau Z'S)] = \exp[\tfrac{1}{2}\tau^2 S'HS]
\end{aligned}$$

where the final equality applies the familiar formula for the moment generating function of the normal distribution. Thus, the best local (small τ^2) test of H_0 versus H_a^* rejects for large values of $L^* = S'HS$.

The test statistic L_T^* shown in equation (13) in the published article is an application of this result: Let $X = (X_1, X_2, ..., X_T)$, $\theta_t = (\xi_t, \sigma_t, \mu_t)$, $\zeta = (\theta_1, \theta_2, ..., \theta_T)$, with $\zeta_0 = (\theta_{1,0}, ..., \theta_{1,0})$, and $f(X|\zeta) = \prod_t f(X_t|\theta_t)$. Under H_a^* assume that

$$\theta_t = \theta_{1,0} + \eta_0 + T^{-1/2} \sum_{j=1}^{t} \eta_j$$

where $\eta_j \sim iid\mathcal{N}(0, \tau^2\Omega)$. In this case, $H = [(ll' + JJ') \otimes \Omega]$ where l is a $T \times 1$ vector of 1s and J is a $T \times T$ lower triangular matrix of 1s. The resulting statistic has the form

$$L^* = S'HS = L_1 + L_2$$

where

$$L_1 = \left(\sum_{t=1}^{T} S_t \right)' \Omega \left(\sum_{t=1}^{T} S_t \right)$$

and

$$L_2 = \sum_{t=1}^{T} \left(T^{-1/2} \sum_{j=t}^{T} S_j \right)' \Omega \left(T^{-1/2} \sum_{j=t}^{T} S_j \right).$$

It is convenient to use $\Omega = \mathrm{Var}(S_t)^{-1}$, the inverse of the information matrix, so that L_1 is recognized as the LM test for testing that the constant parameter θ takes on the value $\theta_{1,0}$ and L_2 is the Nyblom statistic for testing for time variation assuming $\theta_1 = \theta_{1,0}$ is known.

4 Stan Script for the TV-GEV Model

The replication files for the published paper are available on the *Review's* replication archive and at `https://www.princeton.edu/~mwatson/publi.html`. As noted in Section 3.2 of the published article, the time varying parameter paths for the model parameters are computed using Hamilton Monte Carlo methods implemented in Stan. For convenience, we include the Stan script for the TVP-GEV model below. The Stan script for the TV-Exceedance model is similar and can be found in the replication files.

```
// Stan Script for TV-GEV model
functions {

  real GEVcpdf_lpdf(real x, real mu, real sig, real xsi){
    real z=(x-mu)/sig;
    real tau=exp(-6.0*abs(xsi));
    real lht;
    if(1+xsi*z<tau){
      lht=-log(tau)/xsi-(z-(tau-1)/xsi)/tau;
```

```
    }
    else{
      lht=-log1p(xsi*z)/xsi;
    }
    return (1+xsi)*lht-log(sig);
  }

  real GEVpdf_lpdf(real x, real mu, real sig, real xsi){
    real z=(x-mu)/sig;
    real tau=exp(-6.0*abs(xsi));
    real lht;
    if(1+xsi*z<tau){
      lht=-log(tau)/xsi-(z-(tau-1)/xsi)/tau;
    }
    else{
      lht=-log1p(xsi*z)/xsi;
    }
    return (1+xsi)*lht-log(sig)-exp(lht);
  }

}

data {

    real xi_min;
    real xi_max;
    int<lower=0> T;
    int<lower=0> nobs;
    matrix[nobs,T] y;
    real sg_xi;
    real sg_alpha;
    real sg_s;

}

parameters {

    // Level values for parameters
    real trans_xi_level;
    real ln_s_level;
    real m_level;
    // Innovation standard deviations (unscaled) for random walks
    real<lower=0.0,upper=5.0> g_xi;
    real<lower=0.0,upper=5.0> g_alpha;
    real<lower=0.0,upper=5.0> g_s;
    real<lower=0.0,upper=5.0> g_m;
    // Random walks
    vector [T] h_xi;
    vector [T] h_alpha;
    vector [T] h_s;
```

```
    vector [T] h_m;

}

transformed parameters {

    // set sg_m
    real sg_m = exp(ln_s_level);  // This preserves invariance to scale
    // Scale factors for innovations
    real gamma_xi = sg_xi*g_xi/sqrt(T);
    real gamma_alpha = sg_alpha*g_alpha/sqrt(T);
    real gamma_s = sg_s*g_s/sqrt(T);
    real gamma_m = sg_m*g_m/sqrt(T);
     // vector of zeros
    vector [T] zeros = rep_vector(0.0,T);
    // Random walks
    vector [T] drw_xi = cumulative_sum(gamma_xi*h_xi)
                       -mean(cumulative_sum(gamma_xi*h_xi));
    vector [T] drw_alpha = cumulative_sum(gamma_alpha*h_alpha)
                          -mean(cumulative_sum(gamma_alpha*h_alpha));
    vector [T] drw_s = cumulative_sum(gamma_s*h_s)
                      -mean(cumulative_sum(gamma_s*h_s));
    vector [T] drw_m = cumulative_sum(gamma_m*h_m)
                      -mean(cumulative_sum(gamma_m*h_m));
    //  Construct Basic Parameters
    vector [T] trans_xi = trans_xi_level + drw_xi;
    vector [T] ln_alpha = drw_alpha;
              // note that level of ln_alpha is fixed at 0
    vector [T] ln_s = ln_s_level + drw_s;
    vector [T] m = m_level + drw_m;
    // Transformed Parameters
    vector [T] alpha = exp(ln_alpha);
    vector [T] s = exp(ln_s);
    // Construct GEV Parameters
    vector [T] xi = xi_min + (xi_max-xi_min)*((exp(trans_xi))
                           ./(1+exp(trans_xi)));
    vector [T] sigma = s.*(alpha.^xi);
    vector [T] mu = m + (s./xi).*((alpha.^xi)-1);

}

model {

  // Priors
    g_xi ~ exponential(1.0);
    g_alpha ~ exponential(1.0);
    g_s ~ exponential(1.0);
    g_m ~ exponential(1.0);

    trans_xi_level ~ normal(0.0,1.5);
```

```
    // Flat prior on ln_s_level and m_level (imposed in Stan)

    h_xi ~ std_normal();
    h_alpha ~ std_normal();
    h_s ~ std_normal();
    h_m ~ std_normal();

    for (t in 1:T) {
      for (j in 1:nobs-1) {
        y[j,t] ~ GEVcpdf(mu[t],sigma[t],xi[t]);
      }
      y[nobs,t] ~ GEVpdf(mu[t],sigma[t],xi[t]);
    }

}
```