Final Project: Replicating Tail Risk of Contagious Disease article by Cirillo and Taleb with bootstrapping, frequentist and bayesian inference

Srihari Jaganathan

STAT 508, Fall 2021

1 Introduction

In this study we replicate the article "Tail Risk of Contagious Disease"(Cirillo and Taleb (2020)). Cirillo and Taleb use probabilistic approach and extreme value theory to show that pandemic deaths are "fat tailed". This is very important work, in risk domain, if a phenomena that we are studying is determined to be fat tailed then its most likely going to be **unpredictable** and will have destructive consequences. It is very important to note that they published this article in April 2020 before COVID-19 pandemic started to become widespread. Several researchers have realized this unpredictability phenomena unfortunately very late. For instance below is the quote from Carnegie Mellon University's Delphi Research group which serves as the basis of the CDC's official communications on COVID-19 forecasting and the blog post was recently published in International Institute of Forecasters(Reich et al. (2021)):

> **"**... forecasts of cases and hospitalizations showed repeated, **sustained lapses in accuracy** for longer-term forecasts, especially at key points during some the larger pandemic waves. Therefore, starting in September 2021, the Hub decided to **suspend** the inclusion of 2 through 4 week ahead case forecasts and 15 through 28 day ahead hospitalization forecasts in the official ensemble that is generated every week. Modelers and forecasters should continue to innovate and investigate so we can continue to build our understanding of how models can be used to anticipate changes in COVID trends and serve the needs of decision-makers and the general public.**"**

Pasquale Cirillo Source:Website

Nassim Taleb Source:Wikipedia

Science is about understanding properties, not forecasting single outcomes. - Nassim Taleb

As illustrated in Figure 1, most consequential events are not seen in bulk of probability distribution, instead it resides in the tail of the distribution (Taleb, Bar-Yam, and Cirillo (2020)). Therefore its important to study the tails. In this study we use tools developed by Cirillo and Taleb (2020) and replicate their article. We will develop codes from scratch and not rely on standard packages including, graphical tools to determine if the data generating process is from fat-tailed distribution, maximum likelihood estimation, estimating uncertainty such as profile likelihood. We will employ techniques learnt in this class such as bootstrapping to estimate parameter uncertainty. The organization of this study is as follows: in the next section, we will provide brief overview of the data, followed by exploratory data analysis. Then we discuss Cirillo and Taleb (2020) idea of dual distribution which makes it possible to calculate moments and therefore expected value. Next we will employ maximum likelihood estimation and visual approaches to assess the fit of the distribution. To assess the uncertainty of parameter estimates, we will use parametric, non-parametric and bayesian approaches. Finally we conclude the article with implications.

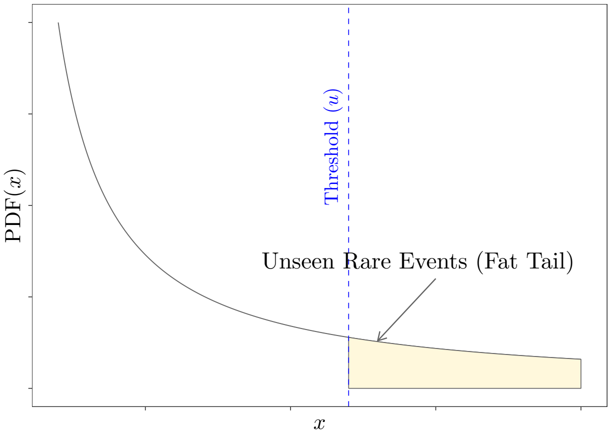


Figure 1: Probability Density function with unseen rare events beyond threshold u

2 Data

Cirillo and Taleb (2020) compiled a list of 72 pandemics with greater than 1000 casualties from 3 different sources (Mark (2021), Wikipedia (2021), Franklin (2021)). The pandemic dataset (see Table 6 in Appendix)has name of the pandemic, start and end year, average estimated death, upper and lower estimates, rescaled death to today's population, and population at the time of pandemic. We will use average estimated death for the analysis. Table 1 shows the descriptive statistics of the average estimated deaths. As we can observe the mean and median are farther apart, in addition kurtosis is extremely high indicating that this is a skewed dataset. Table 2 shows a very large difference between 75^{th} percentile and the 100^{th} percentile indicating heavy skewness. Figure 2 shows time series of average estimated deaths in log scale and the histogram of pandemic casualties. One could observe from this figure that most of the pandemics happen after year 1500 and in addition, histogram indicates that there is extreme skewness in the data. In the rescaled death data, some pandemics scaled to current population such as Plague of Justinian and Black Death have over 2 billion casualties which is approximately 30% of current world population.

Table 1: Descriptive statistics of average estimated deaths ($\times 10^3$) in 72 Pandemics

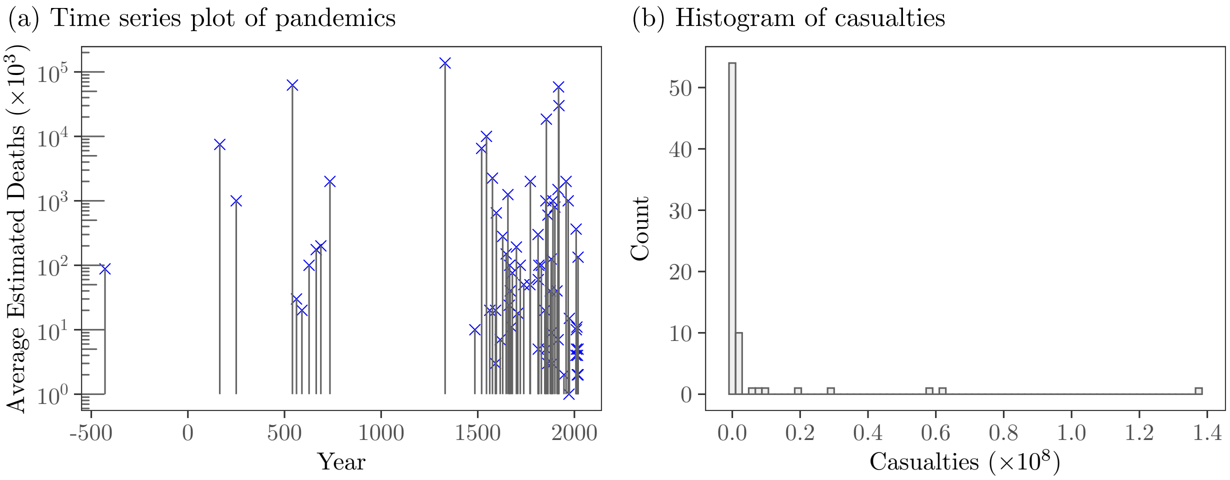
	n	Mean	SD	Median	Trimmed	Min	Max	Range	Skew	Kurtosis
Average Estimated Death	72	4878	19132	82	459.9	1	137500	137499	5.34	31.11

Table 2: Quantile of average estimated deaths ($\times 10^3$) in 72 Pandemics

	0%	25%	50%	75%	100%
Average Estimated Deaths	1	10	82	850	137500

3 Exploratory Data Analysis

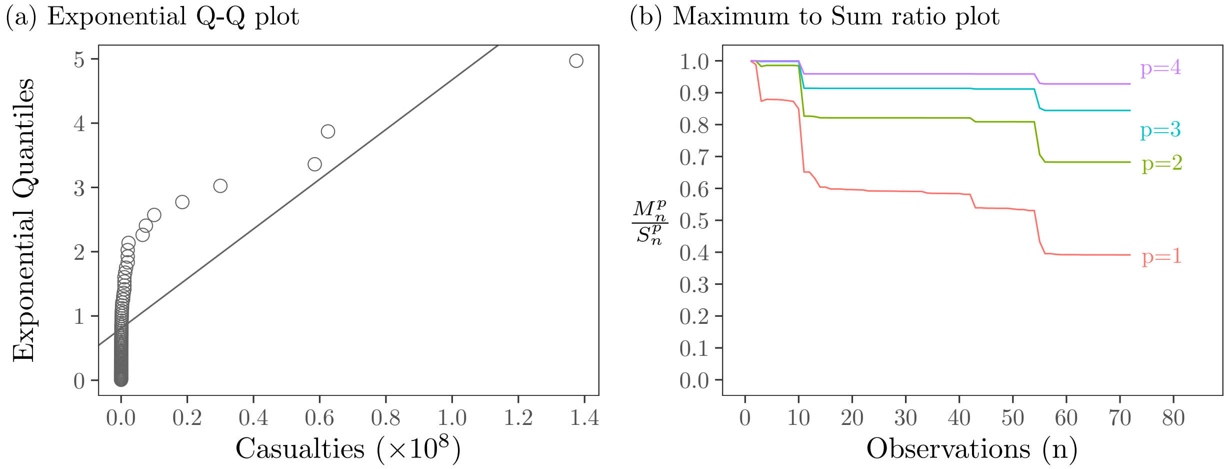
When we suspect heavy skewness and also large outlying observations, one needs to further conduct expolratory analysis to determine of the underlying data generating process is from heavy tailed distribution such as Pareto distribution.Cirillo (Cirillo (2013)) provides an excellent overview and diagnostics to determine of the distribution has a heavy tailed phenomena and rule out other confounding distributions such as Log-normal distribution. In this article we will use diagnostic plots as proposed by Cirillo and Taleb (2020). If the data has a heavy tail, one would observe a convex shape in an exponential Q-Q plot. Figure 3 (a) shows the Exponential Q-Q plot against the observed casualties, as we could clearly see that there is a convex shape which may suggest presence of heavy tail. The second useful plot is the maximum to sum ratio plot. For a random variable that is identical and independently distributed (i.i.d), according to law of large numbers if $E[X^p] < \infty$ the ratio M_n^p/S_n^p would converge to zero for order $p = 1, 2, 3, ...$ as $n \to \infty$. Here p is the moments, $M_n = max(X_1^p, X_2^p, ..., X_n^p)$ is partial maximum value at n for p similarly $S_n^p = \sum_{i=1}^n X_i^p$.


(a) Time series plot of pandemics

(b) Histogram of casualties

Figure 2: Figure (a) shows time series plot of pandemics average estimated deaths in log scale by starting year. Figure (b) shows the histogram of 72 pandemics average estimated deaths which clealy shows heavy skewness in the dataset

Exponential Q-Q, maximum to sum ratio, survival and mean excess (residual life) plots are useful diagnostic tools to determine if the data is from a heavy tail distribution.


(a) Exponential Q-Q plot

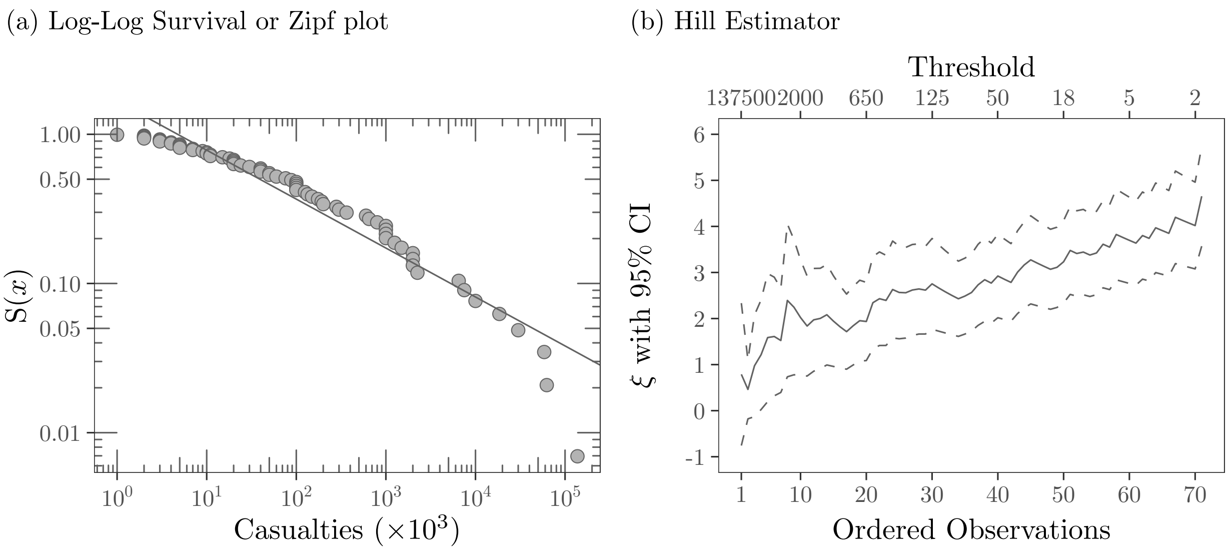
(b) Maximum to Sum ratio plot

Figure 3: (a) shows Exponential Q-Q plot against observed data. Convex shape indicates heavy tail. (b) shows Maximum to Sum plot, and all the moments do not converge to 0 indicating strong presence of heavy tail

Log-Log survival plot or Zipf plot is another useful diagnostic tool to determine if the data is heavy tailed. Figure 4 a. The Zipf plot shows a clear linear downward trend, indicating the presense of heavy tailed distribution. Non parametric hill estimator plot can be used to estimate the tail parameter $\hat{\xi}_n$ which will be discussed in subsequent sections. Let $X_1, X_2, X_3, ..., X_n$ be i.i.d random variable, and the corresponding order statistics be $X_{n,n} \leq ... \leq X_{1,n}$, then the tail parameter $\hat{\xi}_n$ can be estimated as follows:

$$\hat{\xi}_n = \frac{1}{k}\sum_{i=1}^{k} log(X_{i,n}) - log(X_{k,n}), 2 \leq k \leq n. \tag{1}$$

Mean excess plot or mean residual life plot can be used to determine presence of fat tail in addition to also determining threshold value u. An upward linear trend indicates presence of fat tail. Mean excess plot is calculated using following equation.

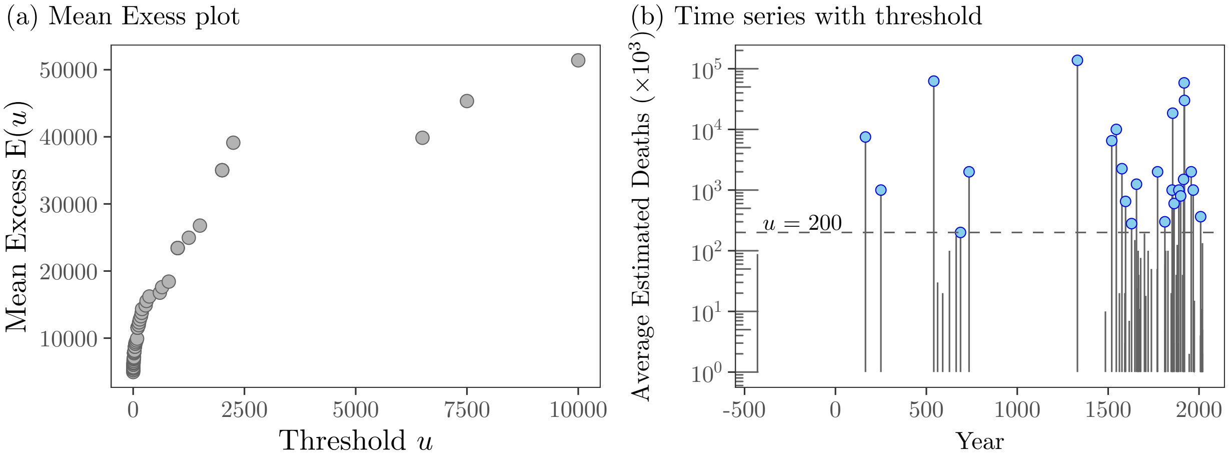

(a) Log-Log Survival or Zipf plot

(b) Hill Estimator

Figure 4: (a) shows Log-Log survival or Zipf plot. A linear trend down indicates a heavy tailed distribution. (b) shows the hill plot, a value of greater than 1 indicates a heavy tailed distribution.

$$\hat{M}(u) = \frac{\sum_{i=1}^{n} X_i \mathbb{1}_{[X_i>u]}}{\sum_{i=1}^{n} \mathbb{1}_{[X_i>u]}} \tag{2}$$

Figure 5 a shows that there is a strong upward trend. Whenever there is change in direction or plateauing that may be use to select threshold. In this instance Cirillo and Taleb (2020) choose the threshold of 200 as on optimal threshold. Figure 5 b shows data that is above the threshold of 200. Table 3 provides a summary of all diagnostic tools used to identify fat-taildness in the data generating process.


(a) Mean Exess plot

(b) Time series with threshold

Figure 5: (a) shows the mean-excess plot at various threshold u. An upward linear trend indicates presence of fat tail. (b) shows time series plot with choosen observation above the threshold $200(\times 10^3)$ casualties

Table 3: Charectersitic of fat tail of various diagnostic plots

Diagnostic Plot	Characteristic of Fat Tail
Maximum to Sum Ratio Plot	Non-convergence to zero of moments
Exponential Q-Q Plot	Convex Shape
Log-Log Survival plot/Zipf Plot	Linear downward trend
Mean Excess Plot	Upward linear trend
Hill Estimator of ξ	Estimator ξ >1

4 Dual Data

Lack of moments as shown in the maximum to sum plot in Figure 3 (b) does not mean that we have infinite mean, as the world is bounded by the population. Cirillo and Taleb (2016b) and Cirillo and Taleb (2016a) proposed dual data approach with special log transformation. Let L be the lower bound of pandemic fatalities and H be the maximum possible fatalities which could be the world population, then the dual data is obtained by following transformation.

$$\varphi(Y) = L - H\ log\left(\frac{H-Y}{H-L}\right) \tag{3}$$

from which $\varphi(Y) \in C^{\infty}$, $\varphi^{-1}(Y) = H$ and $\varphi^{-1}(L) = \varphi(L) = L$. Figure 6 shows what would happen if ignores the upper existence of upper bound H, since only M is observed. The new random variable is defined as $Z = \varphi(Y)$ with lower bound L and infinite upper bound. The expected value for random variable Y can be obtained as follows:

$$E[Y] = (H-L)e^{\frac{\frac{1}{\xi}\sigma}{H}}\left(\frac{\sigma}{H\sigma}\right)^{\frac{1}{\xi}}\Gamma\left(1-\frac{1}{\xi}, \frac{\sigma}{H\xi}\right) + L \tag{4}$$

where $\sigma = \beta(n_u/n)^{\xi}$, n_u is the number of opeservations abover threshold and n is the total number of observations. Parameters β and ξ can be estimated using maximum likelihood estimation.

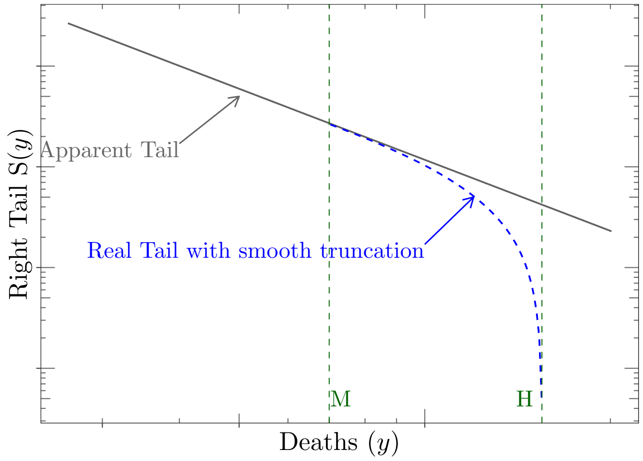


Figure 6: Figure showing a Log-Log plot of what would happen if one ignores the upper bound H, since only M is observed

Dual data approach via special log transformation helps us calculate 'shadow' moments such as expected pandemic fatalaties which is not possible when using raw data.

5 Modeling with Extreme Value Theory using threshold models

As Cirillo and Taleb (2020) mention in the article, most important information is in tail of the distribution. They employ tools from extreme value theory to study pandemic casualties. Generalized Pareto distribution was used to fit the data for dual transformed random variable Z. The cumulative distribution function $F(z)$ and the probability density function $f(z)$ is given by following equation.

$$F(z) = \begin{cases} 1 - \left(1 + \frac{\xi(z-u)}{\beta}\right)^{-1/\xi} & \text{for } \xi \neq 0, \\ 1 - e^{\left(-\frac{z-u}{\beta}\right)} & \text{for } \xi = 0, \end{cases} \tag{5}$$

$$f(z) = \frac{1}{\beta}\left(1 + \frac{\xi(z-u)}{\beta}\right)^{\left(-\frac{1}{\xi}-1\right)} \tag{6}$$

Cirillo and Taleb (2016b) recommend using $u = 200,000$ victims as appropriate threshold. This leads us to $nu = 25$ (34.7 %)observations that are above 200,000 observations. The key parameter of interest is the shape parameter ξ. As noted in previous sections, if $\xi > 1$ then there is no finite moments such as mean. With dual transformations we are able to estimate the mean since random variable Z has finite support between L and H. As far as threshold u,one could use graphical tools to determine appropriate values. As outlined in the exploratory data analysis, one could rely on graphical tools such as Zipf plot and mean excess plot.

5.1 Fitting the data with Maximum likelihood

Maximum Likelihood estimation (Coles et al. (2001)) methods are appropriate for cases where $\xi > 0$. Excellent overview of various estimation methods to fit generalized Pareto distribution are provided in de2010parameter1 and de2010parameter2. Following is the log likelihood equation to estimate parameters $\theta = (\xi, \beta)$.

$$\mathcal{L}_n = \sum_{i=1}^{n} \log f(X_i; \xi, \beta) \tag{7}$$

where ξ and β can be estimated by $(\hat{\xi}, \hat{\beta}) = \text{argmax}_{\xi,\beta}\mathcal{L}_n$. After applying numerical optimization to maximize the log likelihood, we obtain the estimates and standard errors (SE) as outlined in Table 4. The estimates are identical to the one obtained in the Cirillo and Taleb (2016b). 3D surface and contour plot is shown in 7. 3D surface area around the optimum estimates are very flat and not steep which indicates that there will be high level of uncertainty and its reflected in large standard errors.

Table 4: Comparison of estimates from this analysis and Cirillo and Taleb (2020)

	This Analysis		Cirillo and Taleb (2020)	
	Estimate	SE	Estimate	SE
$\beta \times 10^3$	1174.72	535.08	1174.7	536.5
ξ	1.62	0.52	1.62	0.52

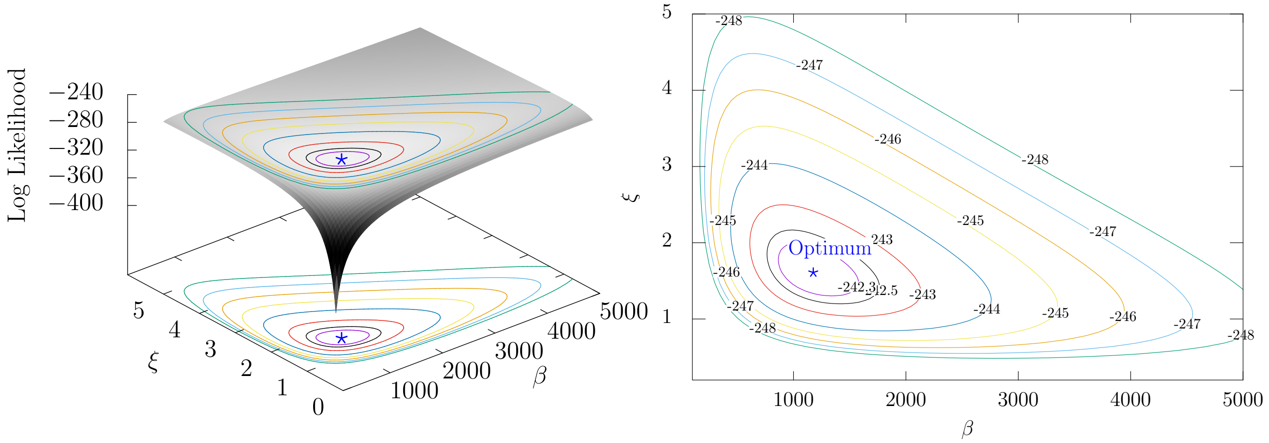


Figure 7: Left figure shows the 3D surface plot of log likelihood function with contours, as can be seen the surface plot is flat and not steep which will indicate there will be high level of uncertainity in estimates. Right figure shows the contour plot with optimum values highlighted at $\beta = 1174.72$ and $\xi = 1.62$.

5.2 Assesing the fit of the distribution

There are two approaches to assess the fit of generalized Pareto distribution (GPD) to empirical data. First one is visual inspection of fit vs actual data and the second is goodness of fit test (Choulakian and Stephens (2001)). In this analysis we use visual inspection to assess the fit of the distribution. Figure 8 (a) shows the Log-Log survival plot of fitted vs actual data. The fit appears to be very good. Similarly fit is realized when we use CDF of the fitted vs actual data as shown in Figure 8 (b).

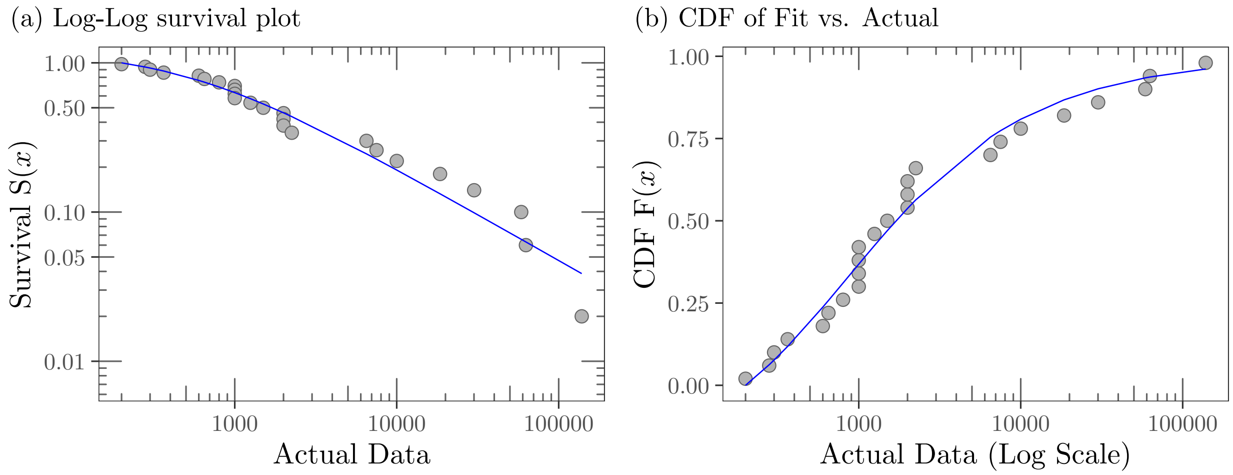


Figure 8: (a) shows the mean-excess plot at various threshold u. An upward linear trend indicates presence of fat tail. (b) shows time series plot with choosen observation above the threshold $200(\times 10^3)$ casualties

Maximum likelihood estimation is appropriate when shape parameter $\xi > 0$. Visual inspection of fitted vs actual and goodness of fit test statistic such as Anderson-Darling or bootstrap test can be used to assess the fit of GPD to the data.

6 Uncertainity of shape parameter ξ

In this section, we study the uncertainty of shape parameter ξ. We employ non-parametric bootstrapping, profile likelihood, parametric bootstrapping, simulation using asymptomatic normality property of maximum likelihood inference, and finally bayesian inference to understand the uncertainty of shape parameter ξ.

6.1 Non-parametric bootstrapping

The pandemic dataset was based on historical reporting which can be inaccurate. There are two forms of inaccuracies that could occur, one could be under/over reporting and the other could be missing information. Recognizing this, Cirillo and Taleb (2016b) perturbed data by randomly varying the data by ±20% and resampled 10,000 times and estimated parameter $\hat{\theta} = (\hat{\xi}, \hat{\beta})$. Frequency histogram of this estimate is shown in Figure 9 (a). Results are very similar to the one reported in Cirillo and Taleb (2016b) and its robust data perturbation i.e., the estimates are very close to maximum likelihood estimation. In addition, Cirillo and Taleb (2016b) jackknifed data by making randomly missing 1% to 10% and estimating the parameters is shown in Figure 9 (b). The estimates are closer to the MLE estimates and very the parameter is very robust to missing values.

6.2 Profile likelihood

Profile likelihood is an excellent approach to estimate confidence interval of estimates. Subba Rao (2021) provides an excellent review of profile likelihood. Wilks theorm gives us the following equation:

$$2\{\mathcal{L}_n(\hat{\xi}_n, \hat{\beta}_n) - \mathcal{L}_n(\xi_0, \beta_{\xi_0})\} \xrightarrow{D} \chi^2_p \tag{8}$$

where ξ_0 is a true parameter, and χ^2_p is Chi-square distribution with p degrees of freedom representing ξ_0 paramater. Using this equation we can construct the confidence interval with $(1-\alpha)$ level of significance and is given by:

$$2\Big\{\mathcal{L}_n(\hat{\xi}_n, \hat{\beta}_n) - \mathcal{L}_n(\xi, \beta_{\xi})\Big\} \leq \chi^2_p(1-\alpha) \tag{9}$$

Figure 10 (a) shows the profile log likelihood. Confidence interval at 95% for ξ parameter is (0.84,3.04) highlighted in blue.

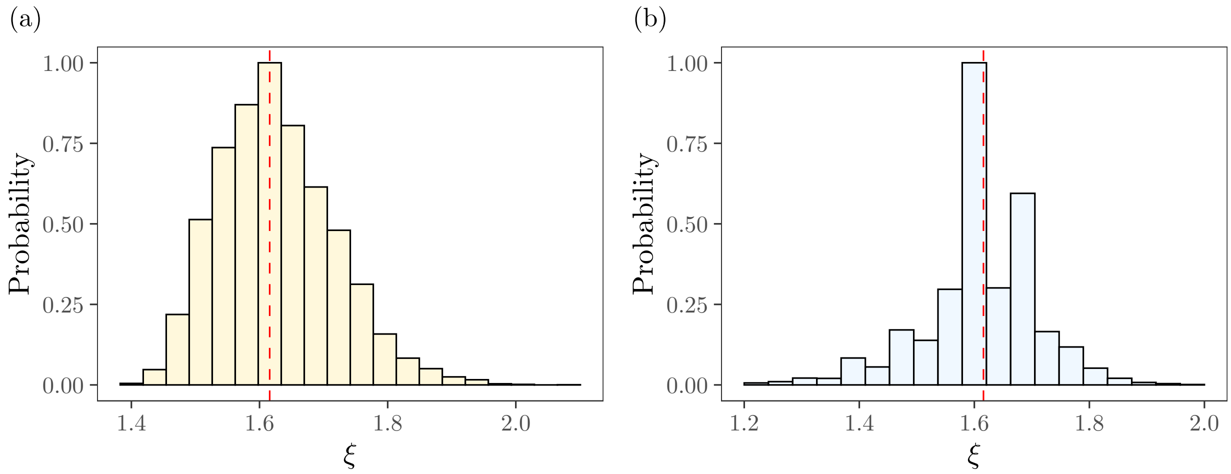


Figure 9: (a) Frequency histogram of estimated ξ parameter from 10,000 resampled data perturbed by ±20% (b)Frequency histogram of jackknifed data by making randomly missing 1% to 10%. MLE estiamtes are shown in red dashed lines at 1.62

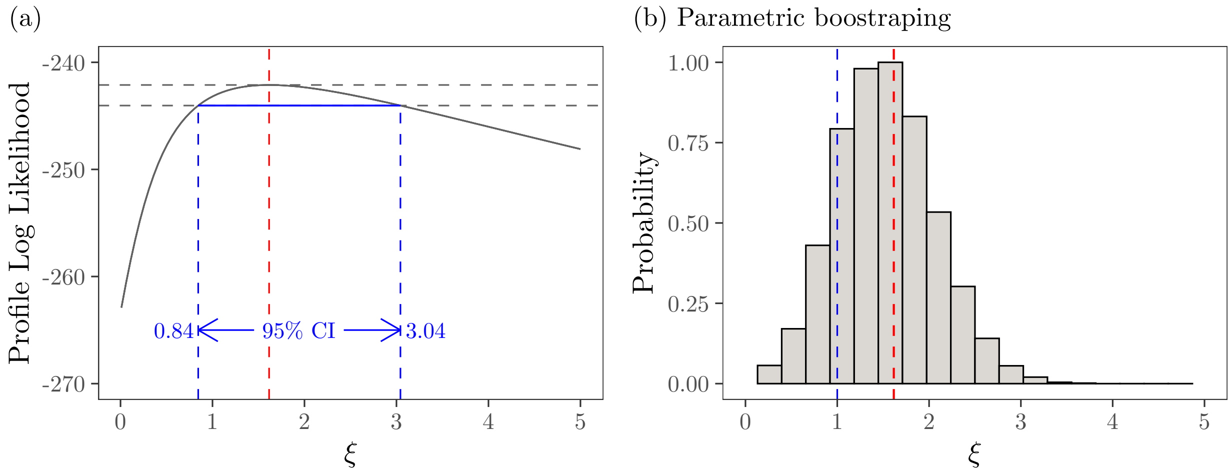


Figure 10: (a) Profile likelihood estimation of ξ with 95% confidence interval (b) Paramteric bootstrapping: 84% of observations are $\xi > 1$

6.3 Parametric Bootstrapping

Parametric bootstrapping is a computationally intensive method, to estimate uncertainity of ξ parameter. The steps involved parametric bootstrapping are as follows:

1. Estimate $\hat{\theta} = (\hat{\xi}, \hat{\beta})$ using MLE.
2. Based on the MLE estimates $\hat{\theta}$, generate random number of sample size $n_u = 25$ (number of observations above threshold u from GPD.
3. Apply MLE and store the estimated parameter values as $\hat{\theta}_i^* = (\hat{\xi}_i^*, \hat{\beta}_i^*)$.
4. Repeat steps 2 to 3 for $i = 1, 2, ..., n$ times, here we set $n = 10,000$.
5. Calculate the quantity of interest based on step 4. In our instance it is the frequency distribution of ξ.

Figure 10 (b) shows the frequency histogram of paramteric bootstrapping. We observe that approximately 84% of observations are > 1.

6.4 Asymptomatic normality

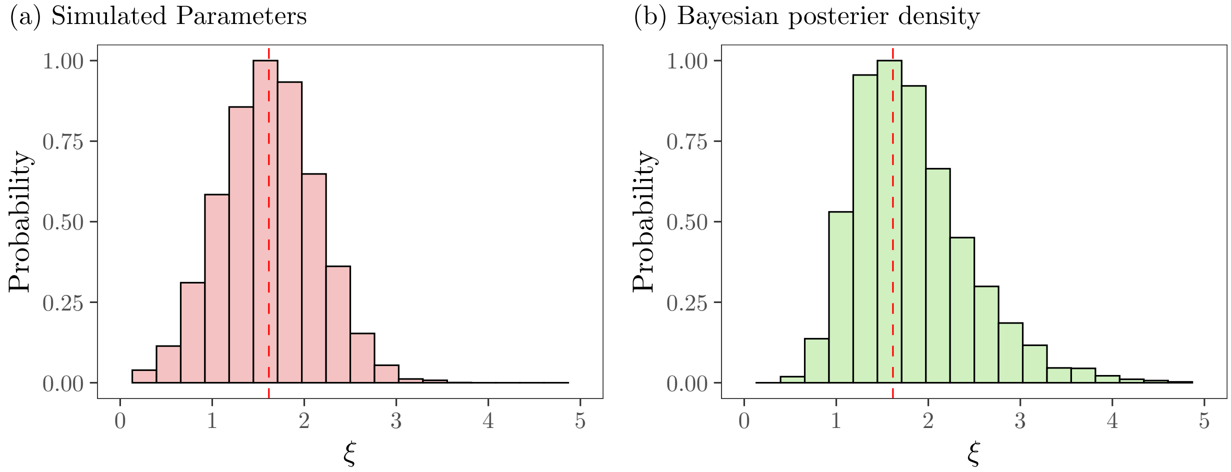
Based on the properties of maximum likelihood estimation we can assume asymptomatic normality of the estimates when the sample size is large. Using this property we simulate using multivariate random normal distribution with mean as $\hat{\theta} = (\hat{\xi}, \hat{\beta})$ and variance as $\hat{V}(\hat{\theta})$ which is the inverse of Fisher information matrix (Hessian at optimum) which is readily available in any optimization routine. The following equation gives us the simulated estimates of parameters $\tilde{\theta} = (\tilde{\xi}, \tilde{\beta})$

$$\tilde{\theta} \sim \mathcal{MVN}(\hat{\theta}, \hat{V}(\hat{\theta})) \tag{10}$$

Figure 11 (a) shows the frequency histogram of $\hat{\xi}$. Greater than 95% of values are greater than 1, and almost $> 99.99\%$ values are greater than 0.

6.5 Bayesian Inference

We use Bayesian estimation to estimate parameter uncertainty. `rstan` package (Stan Development Team (2020)) in `R` to estimate posterior density of parameters. Vehtari (2017) has programmed all functions related to GPD in `Stan`. We used this code to model GPD for Bayesian inference. Figure 11 (b) shows the posterior of $\hat{\xi}$. 95% of values are greater than 1, and 100% values are greater than 0.


(a) Simulated Parameters

(b) Bayesian posterier density

Figure 11: (a) Shows the simulated estimates ξ based on asymptomatic normality and (b) shows the bayesian posterier density

Non-parametric, parametric and bayesian inference all demonstrate that the shape paramater ξ is >0 and almost 85% to 95% probability that they are greater than 1. This analysis confirms that pandemics are fat tailed phenomena.

7 Expected Value of Pandemic Fatalaities

We can estimate the Expected value due to dual transformation even though the shape parameter $\xi > 1$. Plugging in estimated values into equation 4 we obtain an estimate of 20.2 million casualties compared to a naive estimate of 13.96 million for data greater than 200,000. These values are summarized in Table 5. If we had used naive estimates we would have significantly underestimated the casualties.

Table 5: Table Comparing naive sample mean and expected value from GPD dual data. We observe that naive sample mean under estimates the casualties by 35% less than the GPD's expected value.

Estimate (in millions)	Naive Sample	GPD (dual data)	Difference
μ (>200,000)	13.96	20.2	-35%

8 Conclusions and Implications

In this study we successfully replicated the work of Cirillo and Taleb (2020). We developed almost all the analysis from scratch and not relying on any canned functions from standard packages.

Pandemics exhibit fat tail properties and therefore very hard to predict outcomes. This should be considered for risk management and decision making.

9 References

Choulakian, Vartan, and Michael A Stephens. 2001. "Goodness-of-Fit Tests for the Generalized Pareto Distribution." *Technometrics* 43 (4): 478–84.

Cirillo, Pasquale. 2013. "Are Your Data Really Pareto Distributed?" *Physica A: Statistical Mechanics and Its Applications* 392 (23): 5947–62.

Cirillo, Pasquale, and Nassim Nicholas Taleb. 2016a. "Expected Shortfall Estimation for Apparently Infinite-Mean Models of Operational Risk." *Quantitative Finance* 16 (10): 1485–94.

———. 2016b. "On the Statistical Properties and Tail Risk of Violent Conflicts." *Physica A: Statistical Mechanics and Its Applications* 452: 29–45.

———. 2020. "Tail Risk of Contagious Diseases." *Nature Physics* 16 (6): 606–13.

Coles, Stuart, Joanna Bawa, Lesley Trenner, and Pat Dorazio. 2001. *An Introduction to Statistical Modeling of Extreme Values*. Vol. 208. Springer.

Franklin, George K. 2021. "List of Epidemics Compared to Coronavirus (Covid-19)." In. https://listfist.com/list-of-epidemics-compared-to-coronavirus-covid-19; (Date accessed: 12.09.2021).

Mark, Joshua J. 2021. "Plague in the Ancient & Medieval World." In. https://www.worldhistory.org/article/1528/plague-in-the-ancient--medieval-world/; (Date accessed: 12.09.2021).

Reich, Nicholas, Ryan Tibshirani, Evan Ray, and Roni Rosenfeld. 2021. "On the Predictability of Covid-19." In. https://delphi.cmu.edu/blog/2021/09/30/on-the-predictability-of-covid-19/; (Date accessed: 12.09.2021).

Stan Development Team. 2020. "RStan: The R Interface to Stan." http://mc-stan.org/.

Subba Rao, Suhasini. 2021. "The Profile Likelihood." In. https://web.stat.tamu.edu/~suhasini/teaching613/chapter3.pdf; (Date accessed: 12.09.2021).

Taleb, Nassim Nicholas, Yaneer Bar-Yam, and Pasquale Cirillo. 2020. "On Single Point Forecasts for Fat-Tailed Variables." *International Journal of Forecasting*.

Vehtari, Aki. 2017. "Extreme Value Analysis and User Defined Probability Functions in Stan." In. https://mc-stan.org/users/documentation/case-studies/gpareto_functions.html; (Date accessed: 12.09.2021).

Wikipedia. 2021. "List of Epidemics." In. https://en.wikipedia.org/wiki/List_of_epidemics; (Date accessed: 12.09.2021).

10 Appendix

`R` code to reproduce models.

10.1 Replicate Figure 1

```r
x <- seq(1, 10, 0.01)
y <- dpareto(x, 1, 0.1)
df_p <- data.frame(x, y)

shade <- data.frame(x = c(x[x >= 6], max(x), 6), y = c(y[x >=
    6], 0, 0))
```

```
ggplot(data = df_p, aes(x = x, y = y)) + theme_bw() +
    geom_line(size = 0.6, color = "grey39") + geom_polygon(data = shade,
    aes(x, y), col = "grey39", fill = "#FFF8DC") +
    theme(plot.background = element_blank(), axis.text.x = element_blank(),
        axis.text.y = element_blank(), axis.title = element_text(size = 16),
        panel.grid.major = element_blank(), aspect.ratio = 0.7,
        panel.grid.minor = element_blank()) + xlab("$x$") +
    ylab("PDF($x$)") + annotate("segment", x = 7.5,
    xend = 6.5, y = 0.03, yend = 0.013, col = "grey39",
    arrow = arrow(length = unit(2.5, "mm")), lwd = 0.8) +
    geom_vline(xintercept = 6, linetype = "dashed",
        color = "blue") + annotate(geom = "text", x = 4.5,
    y = 0.035, label = "Unseen Rare Events (Fat Tail)",
    hjust = 0, size = 6) + annotate("text", x = 5.7,
    y = 0.05, label = "Threshold ($u$)", col = "blue",
    hjust = 0, size = 5, angle = 90)
```

10.2 Replicate Figure 2

```
## Get Data

import_data <- read.xlsx(file = "/mnt/home/u044338/stat508/final_project/data/long_fat_tail.xlsx",
    1)
dat <- import_data$avg.est



## Maximum Sum Plot
cum_sum = cumsum(import_data$avg.est)

x_1 <- data.frame(start.year = import_data$start.year,
    avg.est = import_data$avg.est, cum_events = 1:length(dat))

p1 <- ggplot(x_1, aes(x = start.year, y = avg.est)) +
    geom_point(shape = 4, size = 1.5, col = "blue") +
    geom_segment(aes(x = start.year, xend = start.year,
        y = 1, yend = avg.est), colour = "grey39") +
    scale_y_log10(breaks = trans_breaks("log10", function(x) 10^x),
        labels = trans_format("log10", math_format(10^.x))) +
    annotation_logticks(sides = "l", colour = "grey39") +
    ylab("Average Estimated Deaths ($\\times10^3$)") +
    xlab("Year") + theme_bw() + theme(plot.background = element_blank(),
    panel.grid.major = element_blank(), aspect.ratio = 0.7,
    panel.grid.minor = element_blank(), axis.title = element_text(size = rel(0.85))) +
    ggtitle("(a) Time series plot of pandemics") +
    theme(plot.title = element_text(size = 10), plot.title.position = "plot")


p2 <- ggplot(x_1, aes(x = start.year, y = cum_events)) +
    geom_line(col = "grey39") + geom_point(shape = 4,
    size = 1.5, col = "blue") + ylab("Cumulative Number of Pandemics") +
    scale_y_continuous(breaks = seq(0, 72, 10)) + xlab("Year") +
    theme_bw() + theme(plot.background = element_blank(),
    panel.grid.major = element_blank(), aspect.ratio = 0.7,
    panel.grid.minor = element_blank(), axis.title = element_text(size = rel(0.85))) +
    theme(plot.title = element_text(size = 10), plot.title.position = "plot")


p3 <- ggplot(data = x_1, aes(x = avg.est/10^5)) + geom_histogram(bins = 72,
    col = "grey39", fill = "grey39", alpha = 0.1) +
    labs(x = "Casualties ($\\times10^8$)", y = "Count") +
    scale_x_continuous(breaks = seq(0, 1.4, 0.2)) +
    scale_y_continuous(breaks = seq(0, 72, 10)) + theme_bw() +
    theme(plot.background = element_blank(), panel.grid.major = element_blank(),
        aspect.ratio = 0.7, panel.grid.minor = element_blank(),
        axis.title = element_text(size = rel(0.85))) +
    ggtitle("(b) Histogram of casualties") + theme(plot.title = element_text(size = 10),
    plot.title.position = "plot")

plot_grid(p1, p3, nrow = 1, rel_widths = c(1, 1), align = "vh")
```

10.3 Replicate Figure 3

```
exp_qq <- data.frame(x=sort(dat)/10^5,y=qexp(ppoints(dat)))
mod <- lsfit(sort(dat/10^5),qexp(ppoints(dat)))

p4 <- ggplot(exp_qq, aes(x=x, y=y)) +
        geom_point(shape = 1,size = 2,col="grey39") + scale_x_continuous(breaks=seq(0.0, 1.4, 0.2)) +
        labs( x= "Casualties ($\\times10^8$)", y="Exponential Quantiles") +
              theme_bw() + theme(plot.background = element_blank(),
              panel.grid.major = element_blank(),aspect.ratio = 0.7,
              panel.grid.minor = element_blank())+
  geom_abline(intercept = mod$coefficients[1],
              slope= mod$coefficients[2],col="grey39")+ ggtitle("(a)")+
  theme(plot.title = element_text(size = 10)) +
  ggtitle("(a) Exponential Q-Q plot") +
  theme(plot.title = element_text(size = 10),plot.title.position = "plot")

msrd <- data.frame( x = seq_along(dat),
                    p1 = (cummax(dat^1)/cumsum(dat^1)),
                    p2 = (cummax(dat^2)/cumsum(dat^2)),
                    p3 = (cummax(dat^3)/cumsum(dat^3)),
                    p4 = (cummax(dat^4)/cumsum(dat^4)))
msrd.gg <- melt(msrd ,  id.vars = 'x', variable.name = 'series')
msrd.gg$series <- factor(msrd.gg$series, levels=c("p1", "p2", "p3","p4"),
                         labels=c("p=1", "p=2", "p=3","p=4"))

p5 <-   msrd.gg %>%
  mutate(label = if_else(x == max(x), as.character(series), NA_character_))  %>%
  ggplot(aes(x = x, y = value, group = series, colour = series)) +
  geom_line() + scale_x_continuous(limits=c(0,85),breaks=seq(0.0, 100, 10))+
  scale_y_continuous(limits=c(0,1),breaks=seq(0.0, 1, 0.1))+
  xlab("Observations (n)")+
  ylab('$\\frac{M_n^p}{S_n^p}$')+ #expression(paste(frac(M[n]^p, S[n]^p)))
  geom_label_repel(aes(label = label),size = 3,
                  nudge_x = 1,box.padding = 0,label.size=NA,fill = NA,
                  na.rm = TRUE) +theme_bw() + theme(plot.background = element_blank(),
              panel.grid.major = element_blank(),aspect.ratio = 0.7,
              panel.grid.minor = element_blank(),legend.position = "none",
              axis.title.y = element_text(angle = 0,vjust = 0.5))+
  ggtitle("(b)") + theme(plot.title = element_text(size = 10))+
  ggtitle("(b) Maximum to Sum ratio plot") +
  theme(plot.title = element_text(size = 10),plot.title.position = "plot")


plot_grid(p4,p5,nrow = 1,align = "vh")
```

10.4 Replicate Figure 4

```
ldat = log(sort(dat, decreasing = T))
n <- length(dat)
ns = 1:(n - 1)
alpha = 0.05

xi = cumsum(ldat[-n])/ns - ldat[-1]

xi_se = xi/sqrt(ns)

lci <- xi - qnorm(1 - alpha/2) * xi_se
hci <- xi + qnorm(1 - alpha/2) * xi_se

hill_est <- data.frame(ns, xi, lci, hci)

hill_est_df <- melt(hill_est, id.vars = "ns", variable.name = "series")


label <- sort(dat, decreasing = T)[c(1, seq(10, 70,
    10))]

p6 <- ggplot(hill_est_df, aes(x = ns, y = value, group = series)) +
    geom_line(aes(linetype = series, colour = series)) +
    scale_linetype_manual(values = c("solid", "dashed",
        "dashed"), labels = c(xi = "$\\xi$", lci = "LCL",
        hci = "HCL")) + theme_bw() + scale_color_manual(values = c("grey39",
    "grey39", "grey39")) + theme(plot.background = element_blank(),
    panel.grid.major = element_blank(), legend.position = "none",
    aspect.ratio = 0.7, panel.grid.minor = element_blank()) +
    scale_x_continuous(limits = c(1, 71), breaks = c(1,
        seq(10, 70, 10)), sec.axis = dup_axis(name = "Threshold",
        breaks = c(1, seq(10, 70, 10)), labels = label),
        guide = guide_axis(check.overlap = TRUE)) +
    scale_y_continuous(limits = c(-1, 6), breaks = seq(-1,
        6, 1)) + xlab("Ordered Observations") + ylab("$\\xi$ with 95$\\%$ CI") +
    guides(colour = "none") + theme(legend.title = element_blank()) +
    ggtitle("(b) Hill Estimator") + theme(plot.title = element_text(size = 10),
    plot.title.position = "plot")


s_eq <- 1 - ppoints(dat)

data_final <- data.frame(x = sort(dat), y = s_eq)

mod1 <- lsfit(s_eq, sort(dat))

mod1 <- lm(log10(s_eq) ~ log10((sort(dat))))

p7 <- ggplot() + geom_point(data = data_final, aes(x = x,
    y = y), color = "grey39", shape = 21, size = 2,
    fill = "grey70") + scale_x_log10(breaks = scales::trans_breaks("log10",
    function(x) 10^x), labels = scales::trans_format("log10",
    scales::math_format(10^.x))) + scale_y_log10(breaks = c(1,
    0.5, 0.1, 0.05, 0.01, 0), limits = c(0.0069, 1)) +
    annotation_logticks(sides = "trbl", colour = "grey39") +
    theme_bw() + theme(plot.background = element_blank(),
    panel.grid.major = element_blank(), aspect.ratio = 0.7,
    panel.grid.minor = element_blank()) + xlab("Casualties ($\\times 10^3$)") +
    ylab("S($x$)") + geom_abline(intercept = mod1$coefficients[1],
    slope = mod1$coefficients[2], col = "grey39") +
    ggtitle("(a) Log-Log Survival or Zipf plot") +
    theme(plot.title = element_text(size = 10), plot.title.position = "plot")

plot_grid(p7, p6, nrow = 1, rel_widths = c(1, 1), align = "vh")
```

10.5 Replicate Figure 5

```
meplot = function(data, cut = 5) {
    # In cut you can specify the number of maxima
    # you want to exclude.  The standard value is
    # 5
    data = sort(as.numeric(data))
    n = length(data)
    mex = c()

    for (i in 1:n) {
        mex[i] = mean(data[data > data[i]]) - data[i]
    }
    data_out = data[1:(n - cut)]
    mex_out = mex[1:(n - cut)]

    return(cbind(data_out, mex_out))

}

me_data <- data.frame(meplot(dat))

p8 <- ggplot() + geom_point(data = me_data, aes(x = data_out,
    y = mex_out), color = "grey39", shape = 21, size = 2,
    fill = "grey70") + theme_bw() + theme(plot.background = element_blank(),
    panel.grid.major = element_blank(), aspect.ratio = 0.7,
    panel.grid.minor = element_blank()) + xlab("Threshold $u$") +
    ylab("Mean Excess E($u$)") + ggtitle("(a) Mean Exess plot") +
    theme(plot.title = element_text(size = 10), plot.title.position = "plot")


p9 <- ggplot(x_1, aes(x = start.year, y = avg.est)) +
    geom_segment(aes(x = start.year, xend = start.year,
        y = 1, yend = avg.est), colour = "grey39") +
    geom_point(data = x_1[x_1$avg.est >= 200, ], shape = 21,
        size = 1.5, col = "blue", fill = "skyblue") +
    scale_y_log10(breaks = trans_breaks("log10", function(x) 10^x),
        labels = trans_format("log10", math_format(10^.x))) +
    annotation_logticks(sides = "l", colour = "grey39") +
    geom_hline(yintercept = 200, col = "grey39", linetype = "dashed") +
    ylab("Average Estimated Deaths ($\\times10^3$)") +
    annotate("text", x = -400, y = 300, label = "$u$ = 200",
        hjust = 0, size = 3) + xlab("Year") + theme_bw() +
    theme(plot.background = element_blank(), panel.grid.major = element_blank(),
        aspect.ratio = 0.7, panel.grid.minor = element_blank(),
        axis.title = element_text(size = rel(0.85))) +
    ggtitle("(b) Time series with threshold") + theme(plot.title = element_text(size = 10),
    plot.title.position = "plot")

plot_grid(p8, p9, nrow = 1, rel_widths = c(1, 1), align = "vh")
```

10.6 Replicate Figure 8

```
### Dual Data ###

L <- 1
H <- 7700000
dual = L - H * log((H - dat)/(H - L))

### Log Likelihood of GPD ####
llfun <- function(parms, threshold = 200, indata = dual) {

    scale <- parms[1]
    shape <- parms[2]
    loc <- 0

    x <- indata[indata >= threshold] - threshold

    llik <- log(1/scale * (1 + shape * (x - loc)/scale)^(-1/shape -
        1))

    return(-sum(llik))
}

### optimize LLik ###
opt.gpd <- optim(fn = llfun, par = c(1000, 1), lower = c(0.001,
    0.001), method = "L-BFGS-B", threshold = 200, indata = dual,
    hessian = T)

### Extract Paramaeters and calculate SE ###
opt.par <- opt.gpd$par
vcovx <- solve(opt.gpd$hessian)
opt.se <- sqrt(diag(vcovx))

beta <- opt.par[1]
xi <- opt.par[2]

### Fit of distribution ###


## Actual Vs Theoretical Plot ##
x_200 = sort(dual[dual >= 200])

qf <- evir::qgpd(ppoints(x_200), xi = xi, beta = beta,
    mu = 200)
```

```
p10 <- ggplot2::qplot(x_200, qf) + geom_point(aes(x = x_200,
    y = qf), color = "grey39", shape = 21, size = 2,
    fill = "grey70") + scale_x_log10(breaks = scales::trans_breaks("log10",
    function(x) 10^x), labels = scales::trans_format("log10",
    scales::math_format(10^.x))) + scale_y_log10(breaks = scales::trans_breaks("log10",
    function(x) 10^x), labels = scales::trans_format("log10",
    scales::math_format(10^.x))) + annotation_logticks(sides = "trbl",
    colour = "grey39") + theme_bw() + geom_smooth(method = "lm",
    se = FALSE, col = "blue", size = 0.4) + theme(plot.background = element_blank(),
    panel.grid.major = element_blank(), aspect.ratio = 0.7,
    panel.grid.minor = element_blank()) + xlab("Actual Data") +
    ylab("Fitted Data") + ggtitle("(a)") + theme(plot.title = element_text(size = 10),
    plot.title.position = "plot")

### Survival Plot ###

surv_emp <- 1 - ppoints(x_200)
surv_fit <- 1 - evir::pgpd(x_200, xi = xi, beta = beta,
    mu = 200)

p11 <- ggplot2::qplot(x_200, surv_emp) + geom_point(aes(x = x_200,
    y = surv_emp), color = "grey39", shape = 21, size = 2,
    fill = "grey70") + scale_x_log10() + scale_y_log10(breaks = c(1,
    0.5, 0.1, 0.05, 0.01, 0), limits = c(0.0069, 1)) +
    annotation_logticks(sides = "trbl", colour = "grey39") +
    theme_bw() + theme(plot.background = element_blank(),
    panel.grid.major = element_blank(), aspect.ratio = 0.7,
    panel.grid.minor = element_blank()) + xlab("Actual Data") +
    ylab("Survival S($x$)") + geom_line(data = data.frame(x = x_200,
    y = surv_fit), mapping = aes(x = x, y = y), col = "blue",
    size = 0.4) + ggtitle("(a) Log-Log survival plot") +
    theme(plot.title = element_text(size = 10), plot.title.position = "plot")



p12 <- ggplot2::qplot(x_200, (1 - surv_emp)) + geom_point(aes(x = x_200,
    y = (1 - surv_emp)), color = "grey39", shape = 21,
    size = 2, fill = "grey70") + scale_x_log10() +
    annotation_logticks(sides = "tb", colour = "grey39") +
    theme_bw() + theme(plot.background = element_blank(),
    panel.grid.major = element_blank(), aspect.ratio = 0.7,
    panel.grid.minor = element_blank()) + xlab("Actual Data (Log Scale)") +
    ylab("CDF F($x$)") + geom_line(data = data.frame(x = x_200,
    y = (1 - surv_fit)), mapping = aes(x = x_200, y = y),
    col = "blue", size = 0.4) + ggtitle("(b) CDF of Fit vs. Actual") +
    theme(plot.title = element_text(size = 10), plot.title.position = "plot")

plot_grid(p11, p12, nrow = 1, rel_widths = c(1, 1),
    align = "vh")
```

10.7 Replicate Figure 9

```
## Non Parametric

n <- 10000
```

```
tol <- 0.2
len <- length(dual)

## Perturbed simulated data

sim_per <- matrix(NA, n, 3)

for (i in 1:n) {

    runi <- runif(len, -0.2, 0.2)
    x_200_dist <- dual + dual * runi

    sim_opt <- optim(fn = llfun, par = c(1000, 1),
        threshold = 200, indata = x_200_dist, hessian = T)
    sim_per[i, 1] <- sim_opt$par[1]
    sim_per[i, 2] <- sim_opt$par[2]
    sim_per[i, 3] <- sim_opt$value
}


df.sim <- data.frame(sim = sim_per[, 2])

p13 <- ggplot(df.sim, aes(x = sim)) + geom_histogram(aes(y = ..ndensity..),
    colour = "black", fill = "#FFF8DC", bins = 20) +
    geom_vline(xintercept = xi, col = "red", linetype = "dashed") +
    theme_bw() + theme(plot.background = element_blank(),
    panel.grid.major = element_blank(), aspect.ratio = 0.7,
    panel.grid.minor = element_blank()) + xlab("$\\xi$") +
    ylab("Probability") + ggtitle("(a)") + theme(plot.title = element_text(size = 10),
    plot.title.position = "plot")

# FFF8DC Jackknife data ###

sim_jk <- matrix(NA, n, 3)

for (i in 1:n) {
    sel <- sample(1:72, sample(65:71, 1), replace = F)
    x_dist <- dual[sel]
    # yu <- fit.gpd(average_200_dist, threshold =
    # 200, method = 'Grimshaw', show = F)
    yu <- optim(fn = llfun, par = c(1000, 1), threshold = 200,
        indata = x_dist, hessian = T)
    sim_jk[i, 1] <- yu$par[1]
    sim_jk[i, 2] <- yu$par[2]
    sim_jk[i, 3] <- length(sel)
}

df.sim.jk <- data.frame(sim = sim_jk[, 2])



p14 <- ggplot(df.sim.jk, aes(x = sim)) + geom_histogram(aes(y = ..ndensity..),
    colour = "black", fill = "#F0F8FF", bins = 20) +
    geom_vline(xintercept = xi, col = "red", linetype = "dashed") +
    theme_bw() + xlim(1.2, 2) + theme(plot.background = element_blank(),
    panel.grid.major = element_blank(), aspect.ratio = 0.7,
    panel.grid.minor = element_blank()) + xlab("$\\xi$") +
    ylab("Probability") + ggtitle("(b)") + theme(plot.title = element_text(size = 10),
    plot.title.position = "plot")

plot_grid(p13, p14, nrow = 1, rel_widths = c(1, 1),
    align = "vh")
```

10.8 Replicate Figure 10

```
# insipired by:
# https://www.r-bloggers.com/2015/11/profile-likelihood/

### Pofile Log Likelihood ###
prof_log_lik = function(a) {
    b = (optim(1, function(z) llfun(c(z, a)), method = "Brent",
        lower = 10, upper = 10000000))$par
    return(llfun(c(b, a)))
}

vx = seq(0.01, 5, length = 101)

vl = -Vectorize(prof_log_lik)(vx)

## Max Like via profile likelihood ##

v1 = optim(1, prof_log_lik)
v_par = v1$par
v_val = -v1$value

## Likelihood ratio at 95% confidence interval
h1 = -optim(1, prof_log_lik)$value - qchisq(0.95, 1)/2
b1 = uniroot(function(z) Vectorize(prof_log_lik)(z) +
    h1, c(0.5, 1.5))$root
b2 = uniroot(function(z) Vectorize(prof_log_lik)(z) +
    h1, c(1.6, 4))$root

p15 <- ggplot2::qplot(vx, vl, geom = "line") + geom_line(col = "grey39") +
    theme_bw() + theme(plot.background = element_blank(),
    panel.grid.major = element_blank(), aspect.ratio = 0.7,
    panel.grid.minor = element_blank()) + xlab("$\\xi$") +
    ylab("Profile Log Likelihood") + geom_vline(xintercept = v_par,
    col = "red", linetype = "dashed") + geom_hline(yintercept = v_val,
    col = "grey39", linetype = "dashed") + geom_hline(yintercept = h1,
    col = "grey39", linetype = "dashed") + geom_segment(aes(x = x1,
    y = y1, xend = x2, yend = y2), data = data.frame(x1 = b1,
    x2 = b2, y1 = h1, y2 = h1), col = "blue") + geom_segment(aes(x = x1,
    y = y1, xend = x2, yend = y2), data = data.frame(x1 = b1,
    x2 = b2, y1 = h1, y2 = h1), col = "blue") + geom_segment(aes(x = x1,
    y = -Inf, xend = x1, yend = y2), data = data.frame(x1 = b1,
    x2 = b2, y1 = h1, y2 = h1), col = "blue", linetype = "dashed") +
    geom_segment(aes(x = x2, y = -Inf, xend = x2, yend = y2),
        data = data.frame(x1 = b1, x2 = b2, y1 = h1,
            y2 = h1), col = "blue", linetype = "dashed") +
    annotate("text", x = 0.8, y = -265, label = "0.84",
        col = "blue", hjust = 1, size = 3) + geom_segment(aes(x = b1,
    y = -265, xend = b2, yend = -265), arrow = arrow(length = unit(0.3,
    "cm"), ends = "both"), col = "blue") + annotate("text",
    x = 3.1, y = -265, label = "3.04", col = "blue",
    hjust = 0, size = 3) + annotate("label", x = 1.94,
    y = -265, label = "95$\\%$ CI", col = "blue", size = 3,
    label.size = NA, fill = "white") + scale_y_continuous(limits = c(-270,
    -240)) + ggtitle("(a)") + theme(plot.title = element_text(size = 10),
    plot.title.position = "plot")



### Parametric Bootstrapping ###

set.seed(1)
nsim = 10000
sim1 <- matrix(NA, nsim, 2)

for (i in 1:nsim) {
    rg <- evir::rgpd(25, xi = xi, mu = 200, beta = beta)
    gp <- optim(fn = llfun, par = c(1000, 1), threshold = 200,
        indata = rg, hessian = T)

    sim1[i, 1] <- gp$par[1]
    sim1[i, 2] <- gp$par[2]

    # print(gp$estimate[2])
}

df.sim1 <- data.frame(sim1)

p16 <- ggplot(df.sim1, aes(x = X2)) + geom_histogram(aes(y = ..ndensity..),
    colour = "black", fill = "#DBD7D2", bins = 20) +
    geom_vline(xintercept = xi, col = "red", linetype = "dashed",
        size = 0.7) + theme_bw() + theme(plot.background = element_blank(),
    panel.grid.major = element_blank(), aspect.ratio = 0.7,
    panel.grid.minor = element_blank()) + xlab("$\\xi$") +
    ylab("Probability") + scale_x_continuous(limits = c(0,
    5)) + ggtitle("(b) Parametric boostraping") + theme(plot.title = element_text(size = 10),
    plot.title.position = "plot") + geom_vline(xintercept = 1,
    linetype = "dashed", col = "blue")

plot_grid(p15, p16, nrow = 1, rel_widths = c(1, 1),
    align = "vh")
```

10.9 Replicate Figure 11

```
load("params.rda")

bayes.parms <- data.frame(xi = params$k)

p17 <- ggplot(bayes.parms, aes(x = xi)) + geom_histogram(aes(y = ..ndensity..),
    colour = "black", fill = "#D0F0C0", bins = 20) +
    geom_vline(xintercept = xi, col = "red", linetype = "dashed") +
    theme_bw() + theme(plot.background = element_blank(),
    panel.grid.major = element_blank(), aspect.ratio = 0.7,
    panel.grid.minor = element_blank()) + xlab("$\\xi$") +
    ylab("Probability") + scale_x_continuous(limits = c(0,
    5)) + ggtitle("(b) Bayesian posterier density") +
    theme(plot.title = element_text(size = 10), plot.title.position = "plot")


mvnorm_freq <- MASS::mvrnorm(n = 10000, mu = opt.par,
    Sigma = vcovx)

df_mvnorm <- data.frame(beta = mvnorm_freq[, 1], xi = mvnorm_freq[,
    2])

p18 <- ggplot(df_mvnorm, aes(x = xi)) + geom_histogram(aes(y = ..ndensity..),
    colour = "black", fill = "#F4C2C2", bins = 20) +
    geom_vline(xintercept = xi, col = "red", linetype = "dashed") +
    theme_bw() + theme(plot.background = element_blank(),
    panel.grid.major = element_blank(), aspect.ratio = 0.7,
    panel.grid.minor = element_blank()) + xlab("$\\xi$") +
    ylab("Probability") + scale_x_continuous(limits = c(0,
    5)) + ggtitle("(a) Simulated Parameters") + theme(plot.title = element_text(size = 10),
    plot.title.position = "plot")



plot_grid(p18, p17, nrow = 1, rel_widths = c(1, 1),
    align = "vh")
```

```
library(gridExtra)
library(rstan)
library(bayesplot)
library(loo)
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores() - 4)
options(scipen = 100, digits = 4)

d <- data.frame(sort(dual))
colnames(d) <- "dual"


n <- dim(d)[1]
yt <- d$dst
ds <- list(ymin = 200, N = n, Nu = 25, Nall = 72, H = 7700000,
    L = 200, y = d$dual, Nt = length(yt), yt = yt)
fit_gpd <- stan(file = "r_stan.stan", data = ds, refresh = 0,
    chains = 4, seed = 100)
params <- rstan::extract(fit_gpd)
```

```
//functions for generalized pareto distribution and inverse gamma
//https://mc-stan.org/users/documentation/case-studies/gpareto_functions.html

functions {
  real gpareto_lpdf(vector y, real ymin, real k, real sigma) {
    // generalised Pareto log pdf
    int N = rows(y);
    real inv_k = inv(k);
    if (k<0 && max(y-ymin)/sigma > -inv_k)
      reject("k<0 and max(y-ymin)/sigma > -1/k; found k, sigma =", k, sigma);
    if (sigma<=0)
      reject("sigma<=0; found sigma =", sigma);
    if (fabs(k) > 1e-15)
      return -(1+inv_k)*sum(log1p((y-ymin) * (k/sigma))) -N*log(sigma);
    else
      return -sum(y-ymin)/sigma -N*log(sigma); // limit k->0
  }
  real gpareto_cdf(vector y, real ymin, real k, real sigma) {
    // generalised Pareto cdf
    real inv_k = inv(k);
    if (k<0 && max(y-ymin)/sigma > -inv_k)
      reject("k<0 and max(y-ymin)/sigma > -1/k; found k, sigma =", k, sigma);
    if (sigma<=0)
      reject("sigma<=0; found sigma =", sigma);
    if (fabs(k) > 1e-15)
      return exp(sum(log1m_exp((-inv_k)*(log1p((y-ymin) * (k/sigma))))));
    else
      return exp(sum(log1m_exp(-(y-ymin)/sigma))); // limit k->0
  }
  real gpareto_lcdf(vector y, real ymin, real k, real sigma) {
    // generalised Pareto log cdf
    real inv_k = inv(k);
    if (k<0 && max(y-ymin)/sigma > -inv_k)
      reject("k<0 and max(y-ymin)/sigma > -1/k; found k, sigma =", k, sigma);
    if (sigma<=0)
      reject("sigma<=0; found sigma =", sigma);
    if (fabs(k) > 1e-15)
      return sum(log1m_exp((-inv_k)*(log1p((y-ymin) * (k/sigma)))));
    else
      return sum(log1m_exp(-(y-ymin)/sigma)); // limit k->0
  }
  real gpareto_lccdf(vector y, real ymin, real k, real sigma) {
    // generalised Pareto log ccdf
    real inv_k = inv(k);
    if (k<0 && max(y-ymin)/sigma > -inv_k)
      reject("k<0 and max(y-ymin)/sigma > -1/k; found k, sigma =", k, sigma);
    if (sigma<=0)
      reject("sigma<=0; found sigma =", sigma);
    if (fabs(k) > 1e-15)
      return (-inv_k)*sum(log1p((y-ymin) * (k/sigma)));
    else
      return -sum(y-ymin)/sigma; // limit k->0
  }
  real gpareto_rng(real ymin, real k, real sigma) {
    // generalised Pareto rng
    if (sigma<=0)
      reject("sigma<=0; found sigma =", sigma);
    if (fabs(k) > 1e-15)
      return ymin + (uniform_rng(0,1)^-k -1) * sigma / k;
    else
      return ymin - sigma*log(uniform_rng(0,1)); // limit k->0
  }
  real gammainc(real a, real x){
    return gamma_q(a,x) * tgamma(a);}
}

//data vector
```

```
data {
  real ymin;
  int<lower=0> N;
  vector<lower=ymin>[N] y;
  int<lower=0> Nt;
  vector<lower=ymin>[Nt] yt;
}
transformed data {
  real ymax = max(y);
}
parameters {
  real<lower=0> sigma;
  real<lower=-sigma/(ymax-ymin)> k;
}
model {
  y ~ gpareto(ymin, k, sigma);
}
generated quantities {
  vector[N] log_lik;
  vector[N] yrep;
  vector[Nt] predccdf;


  for (n in 1:N) {
    log_lik[n] = gpareto_lpdf(rep_vector(y[n],1) | ymin, k, sigma);
    yrep[n] = gpareto_rng(ymin, k, sigma);
  }

  for (nt in 1:Nt)
    predccdf[nt] = exp(gpareto_lccdf(rep_vector(yt[nt],1) | ymin, k, sigma));
}
```

Table 6: Table used in the data analysis. Please consult Cirillo and Taleb (2020) for references

Name	Start Year	End Year	Lower Est $\times 10^3$	Avg Est $\times 10^3$	Upper Est $\times 10^3$	Rescaled Avg Est $\times 10^3$	Population $\times 10^6$
Plague of Athens	-429	-426	75	88.0	100	13376	50
Antonine Plague	165	180	5000	7500.0	10000	283355	202
Plague of Cyprian	250	266	1000	1000.0	1000	37227	205
Plague of Justinian	541	542	25000	62500.0	100000	2246550	213
Plague of Amida	562	562	30	30.0	30	1078	213
Roman Plague of 590	590	590	10	20.0	30	719	213
Plague of Sheroe	627	628	100	100.0	100	3594	213
Plague of the British Isles	664	689	150	175.0	200	6290	213
Plague of Basra	688	689	200	200.0	200	7189	213
Japanese smallpox epidemic	735	737	2000	2000.0	2000	67690	226
Black Death	1331	1353	75000	137500.0	200000	2678283	392
Sweating sickness	1485	1551	10	10.0	10	166	461
Smallpox Epidemic in Mexico	1520	1520	5000	6500.0	8000	107684	461
Cocoliztli Epidemic of 1545–1548	1545	1548	5000	10000.0	15000	165668	461
1563 London plague	1562	1564	20	20.0	20	277	554
Cocoliztli epidemic of 1576	1576	1580	2000	2250.0	2500	31045	554
1592–93 London plague	1592	1593	20	20.0	20	275	554
Malta plague epidemic	1592	1593	3	3.0	3	41	554
Plague in Spain	1596	1602	600	650.0	700	8969	554
New England epidemic	1616	1620	7	7.0	7	97	554
Italian plague of 1629–1631	1629	1631	280	280.0	280	3863	554
Great Plague of Sevilla	1647	1652	150	150.0	150	2070	554
Plague in Kingdom of Naples	1656	1658	1250	1250.0	1250	15840	603
Plague in the Netherlands	1663	1664	24	24.0	24	306	603
Great Plague of London	1665	1666	100	100.0	100	1267	603
Plague in France	1668	1668	40	40.0	40	507	603
Malta plague epidemic	1675	1676	11	11.0	11	143	603
Great Plague of Vienna	1679	1679	76	76.0	76	963	603
Great Northern War plague outbreak	1700	1721	176	192.0	208	2427	603
Great Smallpox Epidemic in Iceland	1707	1709	18	18.0	18	228	603
Great Plague of Marseille	1720	1722	100	100.0	100	1267	603
Great Plague of 1738	1738	1738	50	50.0	50	470	814
Russian plague of 1770–1772	1770	1772	50	50.0	50	470	814
Persian Plague	1772	1772	2000	2000.0	2000	15444	990
Ottoman Plague Epidemic	1812	1819	300	300.0	300	2317	990
Caragea's plague	1813	1813	60	60.0	60	463	990
Malta plague epidemic	1813	1814	5	5.0	5	35	990
First cholera pandemic	1816	1826	100	100.0	100	772	990
Second cholera pandemic	1829	1851	100	100.0	100	772	990
Typhus epidemic in Canada	1847	1848	20	20.0	20	154	990
Third cholera pandemic	1852	1860	1000	1000.0	1000	6053	1263
Cholera epidemic of Copenhagen	1853	1853	5	5.0	5	29	1263
Third plague pandemic	1855	1960	15000	18500.0	22000	111986	1263
Smallpox in British Columbia	1862	1863	3	3.0	3	18	1263
Fourth cholera pandemic	1863	1875	600	600.0	600	3632	1263
Fiji Measles outbreak	1875	1875	40	40.0	40	242	1263
Yellow Fever	1880	1900	100	125.0	150	757	1263
Fifth cholera pandemic	1881	1896	9	9.0	9	42	1654
Smallpox in Montreal	1885	1885	3	3.0	3	14	1654
Russian flu	1889	1890	1000	1000.0	1000	4620	1654
Sixth cholera pandemic	1899	1923	800	800.0	800	3696	1654
China plague	1910	1912	40	40.0	40	185	1654
Encephalitis lethargica pandemic	1915	1926	1500	1500.0	1500	6930	1654
American polio epidemic	1916	1916	6	7.0	7	30	1654
Spanish flu	1918	1920	17000	58500.0	100000	193789	2307
HIV/AIDS pandemic	1920	2020	25000	30000.0	35000	61768	3712
Poliomyelitis in USA	1946	1946	2	2.0	2	5	2948
Asian flu	1957	1958	2000	2000.0	2000	5186	2948
Hong Kong flu	1968	1969	1000	1000.0	1000	2102	3637
London flu	1972	1973	1	1.0	1	2	3866
Smallpox epidemic of India	1974	1974	15	15.0	15	29	4016
Zimbabwean cholera outbreak	2008	2009	4	4.0	4	5	6788
Swine Flu	2009	2009	152	364.0	575	409	6788
Haiti cholera outbreak	2010	2020	10	10.0	10	11	7253
Measles in D.R. Congo	2011	1018	5	5.0	5	5	7253
Ebola in West Africa	2013	2016	11	11.0	11	12	7176
Indian swine flu outbreak	2015	2015	2	2.0	2	2	7253
Yemen cholera outbreak	2016	2020	4	4.0	4	4	7643
2018–19 Kivu Ebola epidemic	2018	2020	2	2.0	3	2	7643
2019-20 COVID-19	2019	2020	117	133.5	150	50	7643
Measles in D.R. Congo	2019	2020	5	5.0	5	5	7643
Dengue fever	2019	2020	2	2.0	2	2	7643