Flood Frequency Explorer

An interactive application to explore the concept of uncertainty in flood estimation.

Enter Station Name (ALL CAPS)

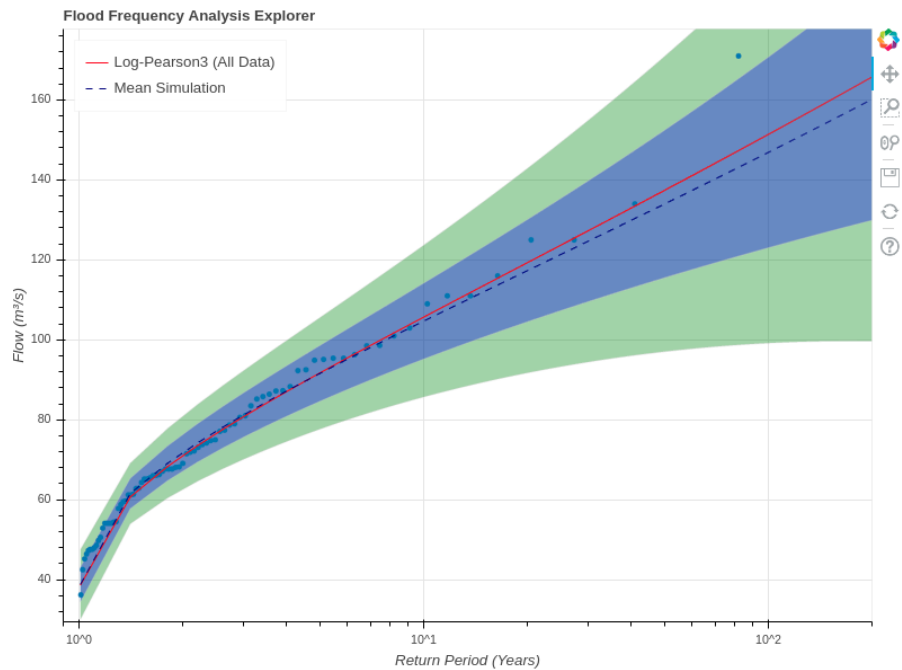| 08MH016: CHILLIWACK RIVER AT OUTLET OF CHILLIWACK LAKE |
| --- |

Sample Size for Simulations

20.0

Number of Simulations

250.0

Mean of 250 simulations of a sample size 20 out of a total 81 years of record. Bands indicate 1 and 2 standard deviations from the mean, respectively.

Flood Frequency Analysis Explorer

Description

Flood Frequency Analysis (FFA) is common practice for hydrologists to estimate flood magnitude. A typical method of estimating a return period flood uses the annual peak instantaneous flow series as an input, fitting a statistical distribution to the data (typically GEV, LPIII, etc.) and then extrapolating beyond the data, where there is considerable uncertainty.

A dataset of sufficient length to capture the true distribution of annual floods does not exist. As a result, practitioners as a rule use limited sample sizes to

1

work with. The samples are then sensitive to sampling variability. Further, the fragile assumption of non-stationarity founds the basic FFA methodology on soggy ground (nice).

Fitting a Log-Pearson type 3 distribution to the full set of annual maximum peak instantaneous flow events, we get some continuous estimate of the underlying distribution (solid red line).

Assuming the set of annual instantaneous floods is comprised of statistically independent events, we can select a subset of size N (without replacement) of events from the total record at random. Fitting a Log-Pearson type 3 distribution to the subset, we get some other continuous estimate of the distribution. The smaller the subset, the poorer we expect the subset to reflect reality. Again, because the individual events are assumed to be statistically independent, we then run the simulation M times.

Taking the results of each of the M simulations of N samples of the population, the mean and standard deviation of all of the distribution fits are calculated for a 'continuous' range of return periods from 1.01 to 200 years. The shaded blue region represents 1 standard deviation from the mean, while the green region represents 2 standard deviations from the mean.

The interpretation of the blue and green regions is something like, *If I had a limited record of N years upon which to base a flood estimate, I can expect the mean of the distribution for some longer period to be within the illustrated range of uncertainty, given the data.*

Thoughts:

- practically speaking, in designing a structure, we choose some design life
- can we work backwards and find an expected design life or uncertainty given the length of dataset upon which a distribution is built?
- how do distributions change over time as data are added?
- can we make some comparison between the length of record we have to work with and either some desired design life or some observation about how the appropriateness of the estimate changes over time?
- Can we make some generalization about the ratio of the subset length to the total data length, where the subset length is related to the record length we have, while the design life is related to the total data length?
- **I could also make it so that the selection of a subset can be done interactively with a box select or lasso tool?**

Getting Started

Dependencies

This application is developed on Ubuntu 18.

Package requirements include: * numpy * pandas * scipy * Bokeh

Installing

- Clone the repo from github

- install the required packages (listed above)

Executing program

From the root directory, execute:

```
>bokeh serve .
```

Note the address that the local Bokeh server launches, indicated in the terminal. Typically, the address is:

```
http://localhost:5006/flood_freq
```

Help

Any advise for common problems or issues.

```
command to run if program contains helper info
```

Authors

Contributors names and contact info

ex. Dan KOvacek ex. [@postnostills](https://twitter.com/postnostills)

Version History

- 0.1
  - Initial Release

License

This project is licensed under the MIT License - see the LICENSE.md file for details

Acknowledgments

Inspiration, code snippets, etc. * awesome-readme