

# frequency\_factor\_example

September 30, 2020

## 1 Frequency Analysis using Frequency Factor

**Question:** For the given data, find the annual maximum daily flow having a return period of 10 Years? Assume that the annual maximum daily flow follow a) Lognormal b) Gumbel distribution.

**Note:** Answer in slide may differ due to rounding off error.

```
[1]: # Initialization: Import required libraries
import os

import numpy as np
from scipy import stats
import pandas as pd
import matplotlib.pyplot as plt

# Data reading
T=10
P=1/T

DATA_FILE=os.path.join('data','Gauge-Discharge Data Jondhra,1980-2019.csv')
_data = pd.read_csv(DATA_FILE, index_col='Dates')
data = _data['01-06-1980':'31-05-2020']['Flow in cumecs']
data.index = pd.to_datetime(data.index, format='%d-%m-%Y')
```

### 1.1 Functions for finding maxima in time series

The following python functions can extract the maximum values from a hydrological time series. The description of all functions is provided in the first line of function body (enclosed by """...""").

```
[2]: # Functions for extracting maxima values from the hydrological time series
def check_1D_timeseries(data):
    """Check if the input is a valid time series"""
    if not isinstance(data, pd.Series):
        raise ValueError('This function only works with pandas Series.')
    if not isinstance(data.index, pd.DatetimeIndex):
        raise ValueError('This function only works with time series.')
```

```

def get_water_year(date):
    """Given a timeseries return water year for each value."""
    date = pd.to_datetime(date)
    # Water year start in June and ends in May next year
    water_yr = np.where(date.month >= 6, date.year, date.year-1)
    return water_yr

def find_year_maxima(data):
    """Find maximum value for different water years."""
    # Check if data is time series
    check_1D_timeseries(data)

    year_maxima = data.groupby(get_water_year(data.index)).max()
    max_flow_dates = data.groupby(get_water_year(data.index)).idxmax()

    return year_maxima, max_flow_dates

def find_exceedance(data, quantile=0.95):
    """Find values in time series above some quantile."""
    # Check if data is time series
    check_1D_timeseries(data)

    return data[data >= data.quantile(q=quantile)]

```

## 1.2 Statistics for annual maximum flood

```

[3]: year_maxima, max_flow_dates = find_year_maxima(data)

mean_x = np.mean(year_maxima.values)
std_x = np.std(year_maxima.values)
C_v = std_x/mean_x
mean_x, std_x, C_v

```

[3]: (4998.527175625, 2482.4226540946233, 0.49663082081457904)

### 1.3 a) Lognormal distribution

Assuming  $Y = \log(X)$ , then

$$\bar{y} = \frac{1}{2} \log \left[ \frac{\bar{x}^2}{C_v^2 + 1} \right]$$

$$S_y = \sqrt{\log(C_v^2 + 1)}$$

```
[4]: y_bar = 0.5*np.log(mean_x**2/(C_v**2+1))
S_y = np.sqrt(np.log(C_v**2+1))
F_y = 1 - P
Z = stats.norm.ppf(F_y)
K_T_ln = Z
y_T_ln = y_bar + K_T_ln * S_y
x_T_ln = np.exp(y_T_ln)
print('Annual maximum flood for return period {} is {:.2f} cumeec,'.format(T,
↪x_T_ln),
      'if the annual maximum flood series follows lognormal distribution.')
```

Annual maximum flood for return period 10 is 8171.37 cumeec, if the annual maximum flood series follows lognormal distribution.

#### 1.4 b) Gumbel Distribution

Assuming  $Y = (X - \beta)/\alpha$ , where

$$\alpha = S_x/1.2825$$

$$\beta = \bar{x} - 0.4501S_x$$

For a value of  $X$  (say  $x_T$ ; having corresponding value of  $y_T$  as per above transformation), using the values of  $\alpha$  and  $\beta$ , the frequency factor is given by:

$$K_T = (y_T - 0.5772)/1.2825$$

where,  $y_T$  is estimated by equation  $P(X \geq x_T) = 1 - e^{-e^{-y_T}}$

```
[5]: y_T_g = -np.log(-np.log(1-P))
K_T_g = (y_T_g-0.5772)/1.2825
x_T_g = mean_x + K_T_g*std_x
print('Annual maximum flood for return period {} is {:.2f} cumeec,'.format(T,
↪x_T_g),
      'if the annual maximum flood series follows Gumbel distribution.')
```

Annual maximum flood for return period 10 is 8237.13 cumeec, if the annual maximum flood series follows Gumbel distribution.