UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

R – ASSIGNMENT FLASH FLOOD

Flood Risk Management

Report By: Aashish Bhardwaj [Student ID: 47659]

Contents

Introduction

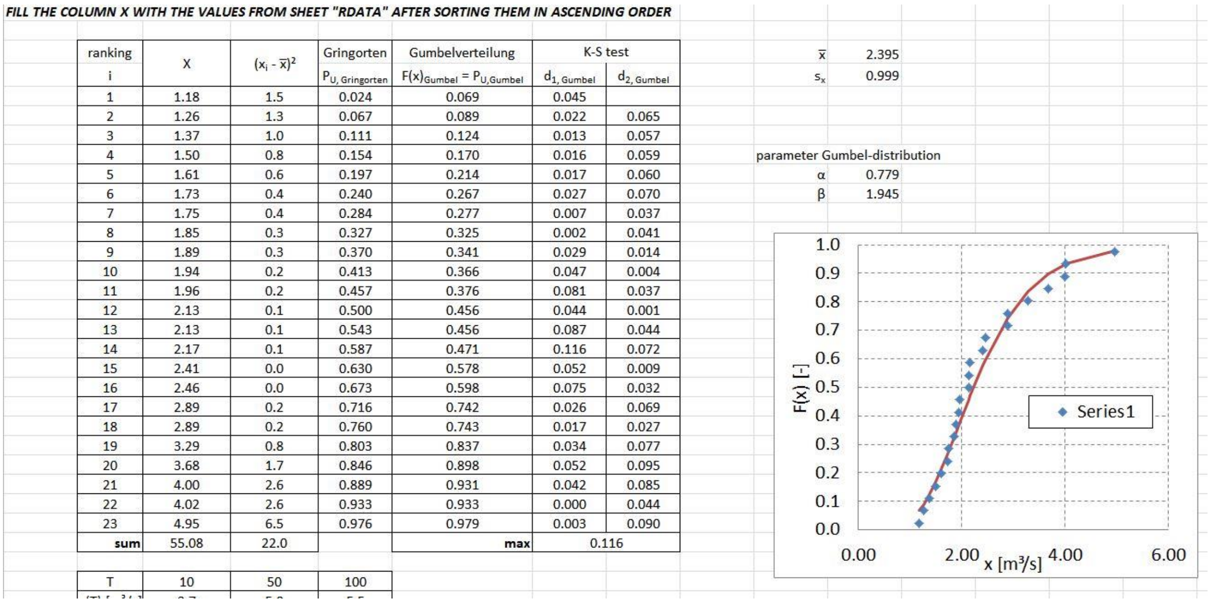
The raw time series from the rain gauge tipping bucket was the input data given to calculate the IDF curve. The coding is done using R Studio. There are several predefined functions in library which were imported to ease the task in coding. The code is complete to the point that it plots the IDF curve and also passes Kolmogorov-Smirnov Goodness-of-Fit Test. There is work in progress regarding the synthetic time series and plotting all the goodness of fit graph together.

Methodology

line#	Explanation
1	removing all the variables and data frame created
2	setting up the directory to work
3	calling dplyr which is used for left join (merging the uniform time series with the non uniform series given in the beginning
4	forecast library to run ma function
5	calling zoo library
6	ggplot2 to plot the IDF curve
7	Read the data from the text file and store it as data frame
8	create a combined date time
9	remove the date column
10	remove the time column from df
11	allocated valye of 0.1 mm to all the time series
12	agregate the time series with values close to seconds
13	rename the column 2 of aggregated ts to "DP"
14	rename the column 1 of aggregated ts to "DT"
15	create ts2 - new time series with uniformaly distributed values
16	rename the ts2 column 1 as DT
17	assign depth value of 0 into the new uniform time series (ts2)
18	create final time series by merging aggregated and uniform time series using left join
19	remove the extra column dp2 which had initially as zero values for depth
20	all the "NA" values existing in "ts3", as replaced by "0"
21-27	Calculating the moving average for [15,30,60,120,360,720,1440] minutes
28	replacing the NA values with 0
29	creating an additional column with values of year only
30	complex nested function in order to deal with the "coerced double" error
31	creating new data frame for the years of the time series
32	renaming the column of IDF with "year"
33-39	aggregating the maximum values of of ts3 year wise
40	creating a new dataframe with 1st column having hour value
41	renaming the column 1 of t2 with "hr"
42	taking mean values of all the columns of IDF except 1st one
43	taking the standard deviation of all the columns of IDF
44-48	calculating rhe intensities for different return period using gumbel
49-69	plotting the IDF
72	importing the IDF data to new dataframe "gf"
73	creating a dataframe with hourly values
74	renaming the column of com with "duration"
75-109	Nested FOR to find the max gumbel values of all the durations and compare it with critical value "0.247", true for n=23
112	creating new data frame from IDF
113	putting data in ascending order of max intensity value for each year
114-115	renaming column of dataframe
116	calculating mean for max intensity values
117	calculating standard deviation for max internsity value
119-130	calculating the data required for K-S test . Values of Gringorten and Gumbelverteilung.
131	find maximum out of D1 and D2
132	plotting the goodness of fit

For the code submitted in Github “R_AashishBhardwaj_v24” , above is the explanation for every line mentioned in the code. Better code could have been developed to optimize the memory allocated and keeping the code short, but instead long and explicit code was preferred for the understanding.

The methodology which was for Goodness of fit is based on excel attached with the report. Following is the excel we used in TU-Dresden, Germany.



FILL THE COLUMN X WITH THE VALUES FROM SHEET "RDATA" AFTER SORTING THEM IN ASCENDING ORDER

ranking	X	$(x_i - \bar{x})^2$	Gringorten	Gumbelverteilung	K-S test	
i			$P_{U, Gringorten}$	$F(x)_{Gumbel} = P_{U,Gumbel}$	$d_{1, Gumbel}$	$d_{2, Gumbel}$
1	1.18	1.5	0.024	0.069	0.045	
2	1.26	1.3	0.067	0.089	0.022	0.065
3	1.37	1.0	0.111	0.124	0.013	0.057
4	1.50	0.8	0.154	0.170	0.016	0.059
5	1.61	0.6	0.197	0.214	0.017	0.060
6	1.73	0.4	0.240	0.267	0.027	0.070
7	1.75	0.4	0.284	0.277	0.007	0.037
8	1.85	0.3	0.327	0.325	0.002	0.041
9	1.89	0.3	0.370	0.341	0.029	0.014
10	1.94	0.2	0.413	0.366	0.047	0.004
11	1.96	0.2	0.457	0.376	0.081	0.037
12	2.13	0.1	0.500	0.456	0.044	0.001
13	2.13	0.1	0.543	0.456	0.087	0.044
14	2.17	0.1	0.587	0.471	0.116	0.072
15	2.41	0.0	0.630	0.578	0.052	0.009
16	2.46	0.0	0.673	0.598	0.075	0.032
17	2.89	0.2	0.716	0.742	0.026	0.069
18	2.89	0.2	0.760	0.743	0.017	0.027
19	3.29	0.8	0.803	0.837	0.034	0.077
20	3.68	1.7	0.846	0.898	0.052	0.095
21	4.00	2.6	0.889	0.931	0.042	0.085
22	4.02	2.6	0.933	0.933	0.000	0.044
23	4.95	6.5	0.976	0.979	0.003	0.090
sum	55.08	22.0		**max**	0.116	

T	10	50	100

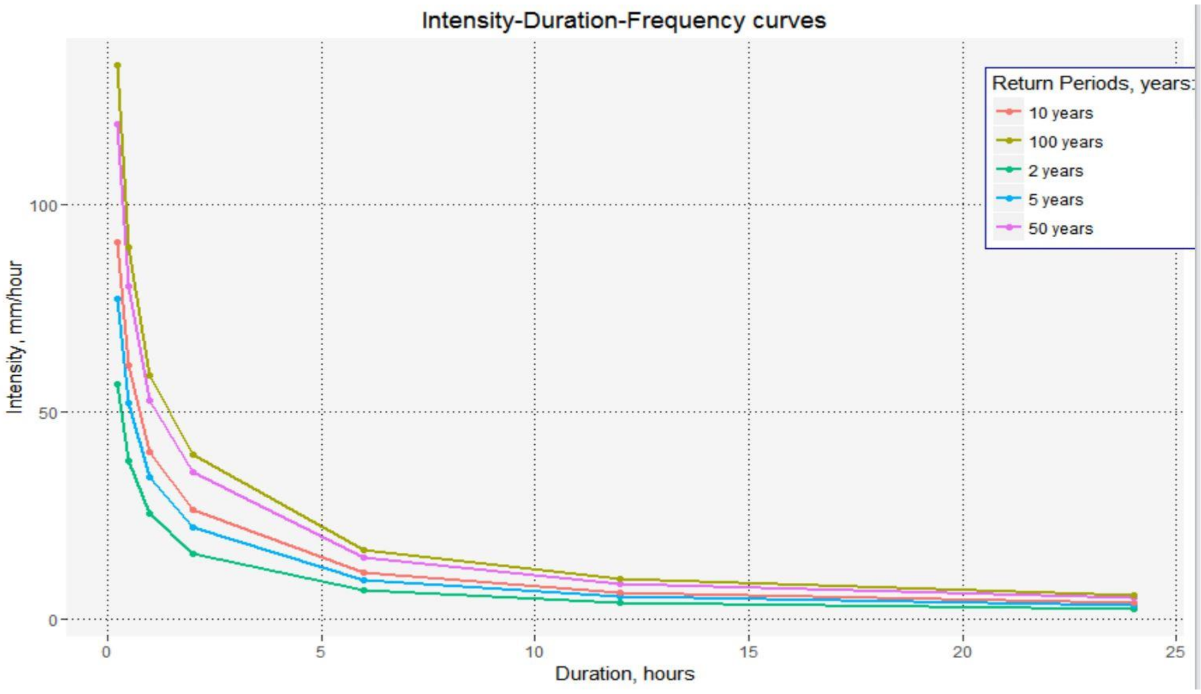
$\bar{x}$	2.395
s_x	0.999

parameter Gumbel-distribution

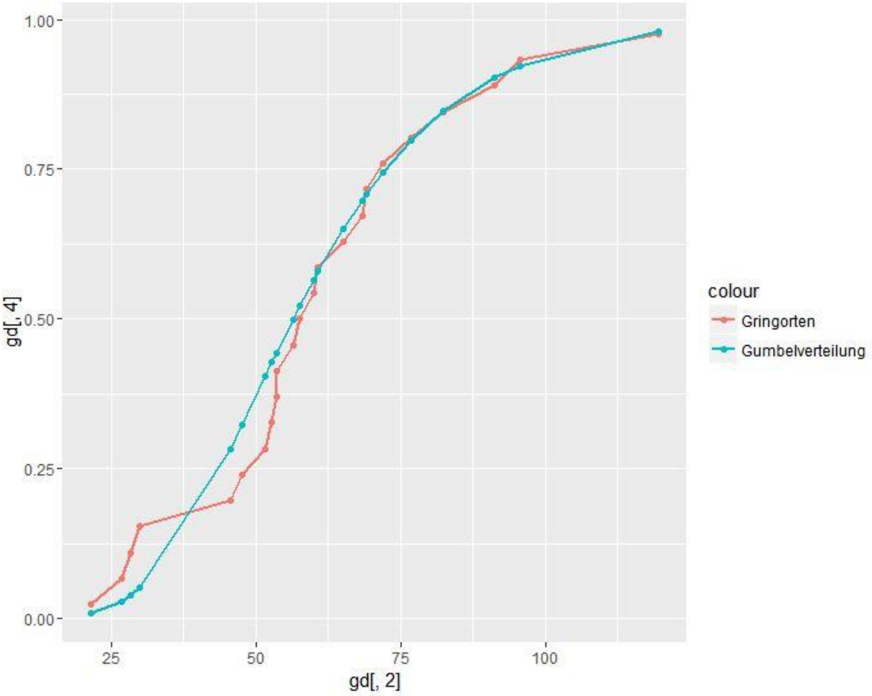
α	0.779
β	1.945

Results

Below is the final IDF curve produced from the data.



Following the output for goodness of fit for K-S test.



Following is the output from the code which reads that K-S test is passed for all time duration.

```
[1] "The curve fits good for hour value :  "
[1] 0.25
[1] "The curve fits good for hour value :  "
[1] 0.5
[1] "The curve fits good for hour value :  "
[1] 1
[1] "The curve fits good for hour value :  "
[1] 2
[1] "The curve fits good for hour value :  "
[1] 6
[1] "The curve fits good for hour value :  "
[1] 12
[1] "The curve fits good for hour value :  "
[1] 24
```

This reveals that when we calculate maximum out of d1 and d2 for a given duration. The maximum value is lower than the critical value of “0.247” and hence passes the test.