CE462A

HMWK - 2

Given the data Plot the Intensity-Duration-Frequency (IDF) curves by first estimating annual maximum intensity for some speciific duration strom of some return periods

Submitted By:
Prashant Chouhan
170492
IIT Kanpur

Instructor:
Shivam Tripathi

Introduction:

Intensity-Duration-Frequency (IDF) curves describe the relationship between rainfall intensity, rainfall duration, and return period. IDF curves are commonly used in the design of hydrologic, hydraulic, and water resource systems. IDF curves are obtained through frequency analysis of rainfall observations.

Methodology:

First and foremost, task is to find the annual maximum rainfall intensity for specific durations (or the annual maximum rainfall depth over the specific durations). Durations used for the design applications are: 1-hr, 2-hr,4-hr, 8-hr, 12-hr, and 24-hr.

The development of IDF curves requires that a frequency analysis be performed for each set of annual maxima, one each associated with each rain duration. The basic objective of each frequency analysis is to determine the exceedance probability distribution function of rain intensity for each duration. For the frequency analysis fit a theoretical Extreme Value (EV) distribution (e.g., Gumbel Type I) to the observations and then use the theoretical distribution to estimate the rainfall events associated with given exceedance probabilities.

Theoretical Extreme Value (EV) Distribution Approach:

The Gumbel Type I distribution is,

$$G(x;\mu,\beta) = \frac{1}{\beta} e^{\frac{x-\mu}{\beta}} e^{-e^{\frac{x-\mu}{\beta}}}$$

where μ is the location parameter and b are the scale parameter. It can be shown that the value of the random variable XT associated with a given return period, T, may be obtained from the following expression,

$$X_T = \bar{X} + K_T S$$

Where,

$$\bar{X} = \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{and} \qquad S = \frac{1}{m-1}\sum_{i=1}^{m}(x_i - \bar{X})^2$$

where $\bar{X}$ is the mean of the observations, and S is the standard deviation of the observations. The frequency factor associated with return period T, K_T, is given by

$$K_T = -\frac{\sqrt{6}}{\pi}[0.5772 + \ln(\ln(\frac{T}{T-1}))]$$

X_T is the depth of the rainfall. We get the Intensity by dividing X_T by the time Duration D.

Tables:

Duration (Hrs)

Years	1	2	4	8	12	24
1	14.87	22	32.07	43.87	46.27	46.27
2	12.18	16.04	23.35	33.05	33.52	41.93
3	17.95	24.28	34.98	39.26	54.18	63.4
4	14.42	22.92	29.68	33.81	34.53	46.6
…	…	…	…	…	…	…
97	18.31	24.08	32.82	39.32	44.86	63.19
98	13.72	17.96	27.9	32.39	41.27	42.57
99	16.31	22.99	33.35	34.81	42.93	44.85
Mean	16.69	27.03	36.22	42.87	47.530	53.58
Std	3.96	7.60	11.37	13.55	15.45	16.17

Table 1: Annual maximum rainfall (mm)for some years of different durations and their mean and Std

Return Period (Years)

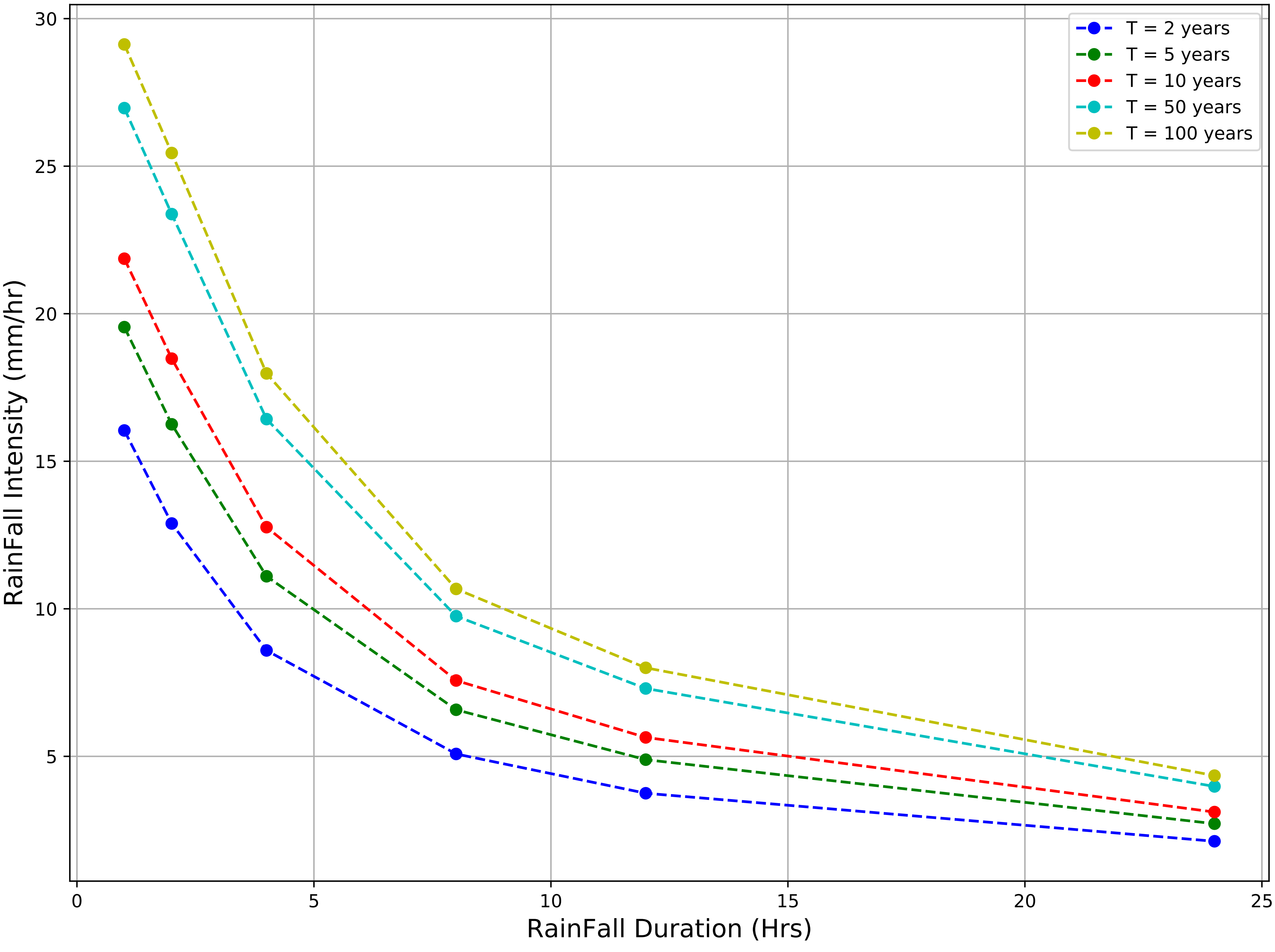
	2	5	10	50	100
K_T value	-0.164272	0.719457	1.304563	2.592288	3.136680

Table 2: Calculation of K_T value by the formula mentioned above

Return Period (Years)

Duration (Hrs)	2	5	10	50	100
1	16.0432	19.5455	21.8642	26.9675	29.1249
2	12.8904	16.2518	18.4774	23.3755	25.4462
4	8.5887	11.102	12.766	16.4283	17.9765
8	5.0814	6.5786	7.5699	9.7515	10.6738
12	3.7493	4.8875	5.6411	7.2996	8.0008
24	2.1219	2.7175	3.1119	3.9799	4.3469

Table 3: Rainfall intensity (mm/hr) associated with each return period.



T = 2 years
T = 5 years
T = 10 years
T = 50 years
T = 100 years
RainFall Intensity (mm/hr)
RainFall Duration (Hrs)
0
5
10
15
20
25
30

Fitting empirical equation:

To Fit an empirical equation of the form,

$$i = \frac{KT^x}{(D+a)^n}$$

We have to estimate parameters **K, x, a, n.** For this purpose we set a value for **a** = 1.2 because we know from the experimental data of these coefficients that generally **a** is in the range (0,2) and slight variation in the value of **a** does not make a big difference to the other coefficients.

To find the other coefficients we apply linear regression approach.

$$i = \frac{kT^x}{(D+a)^n}$$
$$log(i) = log(k) + x * log(T) - n * log(D+a)$$
$$Y = A + B * X_1 + C * X_2$$

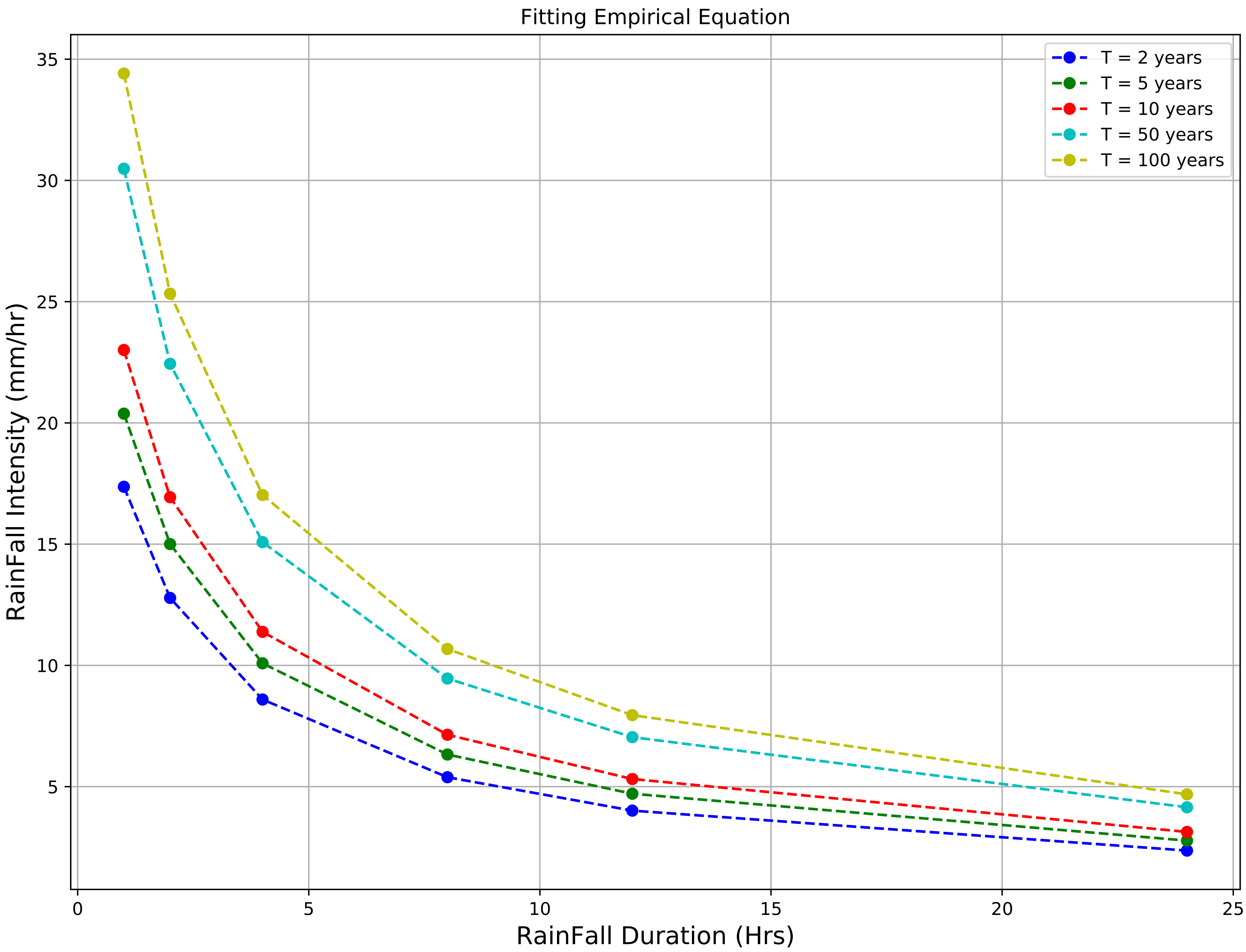
Where,

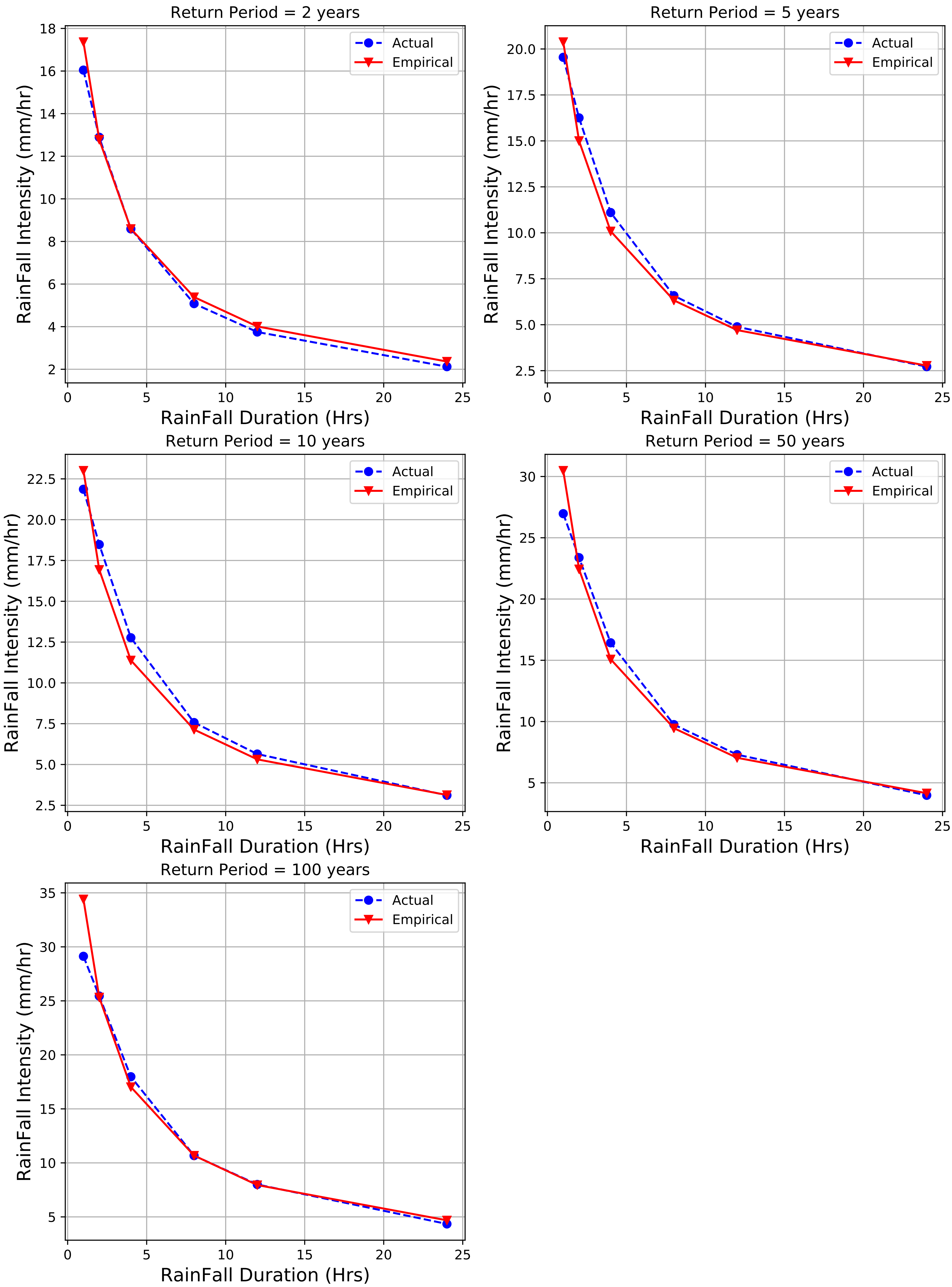
$$Y = log(i) \quad \text{i in (cm/hr)}$$
$$A = log(k)$$
$$B = x$$
$$C = -n$$
$$X_1 = log(T) \quad \text{T in (years)}$$
$$X_2 = log(D+a) \quad \text{D in (Hrs)}$$

$$X = [\,1 \quad X_1 \quad X_2\,]$$
$$b = \begin{bmatrix} A \\ B \\ C \end{bmatrix}$$
$$b = (X^T X)^{-1} X^T Y$$
$$K = exp(A); \quad x = B; \quad n = -C; \quad a = \text{(earlier specified)}$$

K	x	a	n
2.93	0.17	1.2	0.82

Table 4: Values of Coefficients in the empirical Equation


Fitting Empirical Equation
T = 2 years
T = 5 years
T = 10 years
T = 50 years
T = 100 years
RainFall Intensity (mm/hr)
RainFall Duration (Hrs)
0
5
10
15
20
25
30
35


Return Period = 2 years
Return Period = 5 years
Return Period = 10 years
Return Period = 50 years
Return Period = 100 years
Actual
Empirical
RainFall Intensity (mm/hr)
RainFall Duration (Hrs)

References:

- https://www.engr.colostate.edu/~ramirez/ce_old/classes/cive322-Ramirez/CIVE322-Basic-Hydrology.htm
- https://pandas.pydata.org/pandas-docs/stable/getting_started/tutorials.html
- https://www.wikiwand.com/en/Gumbel_distribution
- Class handouts.
- https://www.google.com/

In [127]:

```
import pandas as pd
import numpy as np
import math
import matplotlib.pyplot as plt
from sklearn import linear_model
```

In [128]:

```
df = pd.read_excel("Rainfall_Data.xls")
data = np.zeros((99, 24))
FinalData = np.zeros((6, 5))
```

In [129]:

```
for i in range(99):
    df1 = (((df.where(df['Year']==i+1)).dropna()).drop(['Year','Day'],axis=1)).T
    for j in range(24):
        data[i][j] = max((df1.rolling(j+1).sum()).max())
data1 = pd.DataFrame(data=data[:,[0,1,3,7,11,23]],index=range(1,100),columns=[1,2,4
data1
```

Out[129]:

	1	2	4	8	12	24
1	14.87	22.00	32.07	43.87	46.27	46.27
2	12.18	16.04	23.35	33.05	33.52	41.93
3	17.95	24.28	34.98	39.26	54.18	63.40
4	14.42	22.92	29.68	33.81	34.53	46.60
5	20.27	37.49	41.88	46.11	50.88	54.67
...	...	...	...	...	...	...
95	22.08	41.74	53.44	56.14	56.14	56.14
96	16.02	30.37	37.57	55.29	60.80	60.80
97	18.31	24.08	32.82	39.32	44.86	63.19
98	13.72	17.96	27.90	32.39	41.27	42.57
99	16.31	22.99	33.35	34.81	42.93	44.85

99 rows × 6 columns

In [130]:

```
MeanStd = pd.DataFrame([data1.mean(axis = 0),data1.std(ddof = 1,axis = 0)],index =
MeanStd
```

Out[130]:

	1	2	4	8	12	24
Mean	16.694242	27.030505	36.223434	42.877475	47.530606	53.582020
Std	3.963013	7.607383	11.375960	13.553562	15.455382	16.177189

In [131]:

```
Kt= []
for i in [2,5,10,50,100]:
    Kt.append(-1*math.sqrt(6)/math.pi*(0.5772+math.log(math.log(i/(i-1)))))
Kt = pd.DataFrame(np.transpose(Kt),index=['2','5','10','50','100'],columns=['K_T'])
Kt
```

Out[131]:

	2	5	10	50	100
K_T	-0.164272	0.719457	1.304563	2.592288	3.136681

In [328]:

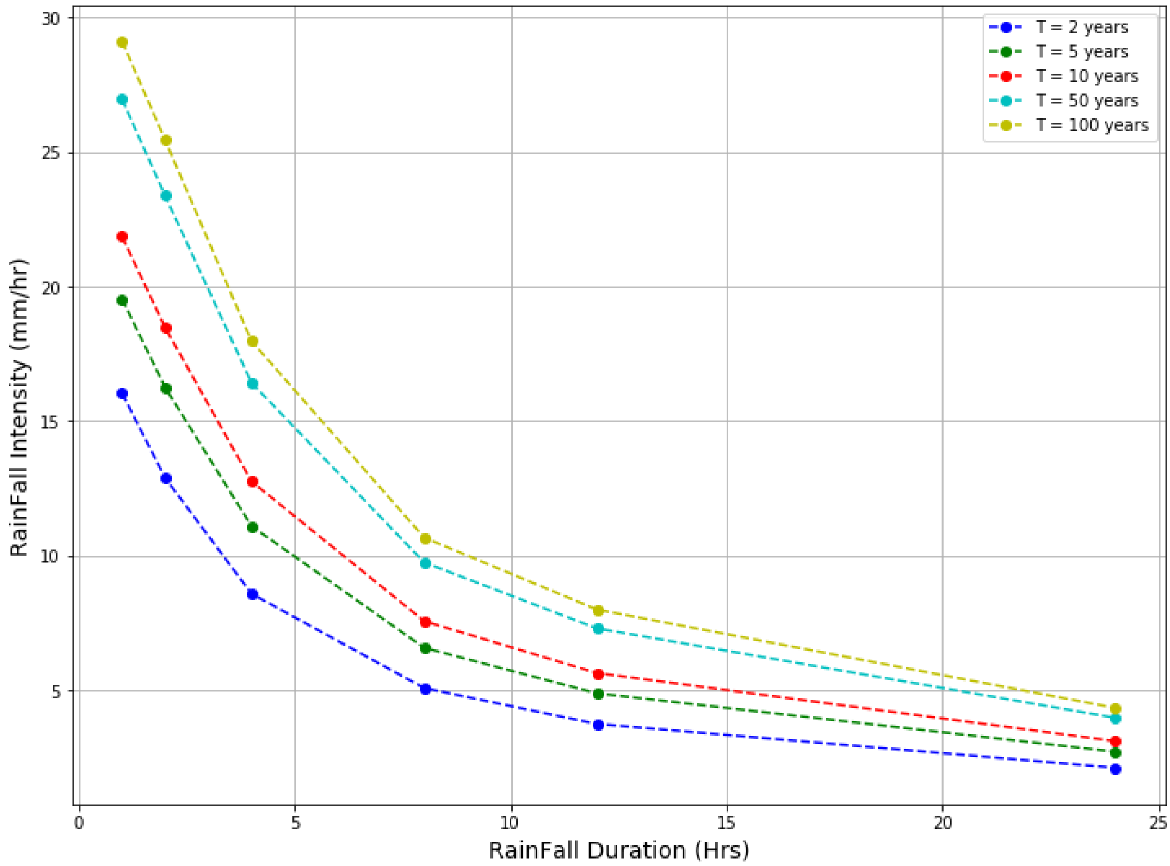
```
for i in range(6):
    for j in range(5):
        FinalData[i][j] =(
            round(((MeanStd.iloc[0]).iloc[i]+(Kt.iloc[0]).iloc[j]*(MeanStd.iloc[1])
FinalData_df = pd.DataFrame(FinalData,index = ['1', '2','4','8','12','24'],columns
FinalData_df.to_excel("output.xlsx")
```

In [329]:

```
RetPer = []
for i in range(6):
    RetPer.append(2)
    RetPer.append(5)
    RetPer.append(10)
    RetPer.append(50)
    RetPer.append(100)
Dur = []
for i in [1,2,4,8,12,24]:
    for j in range(5):
        Dur.append(i)
Y = np.array(np.log(0.1*FinalData.ravel()))
a = 1.2
X = np.transpose([np.ones(30),np.array(np.log((RetPer))), np.log(np.array(Dur) + a)
coef = np.dot(np.dot(np.array(np.linalg.inv((np.dot( np.transpose(X),X)))),np.trans
k = math.exp(coef[0])
x = (coef[1])
n = -(coef[2])
def f(Dur,ret):
    return (k*(ret**x)/((Dur+a)**n))
```

In [330]:

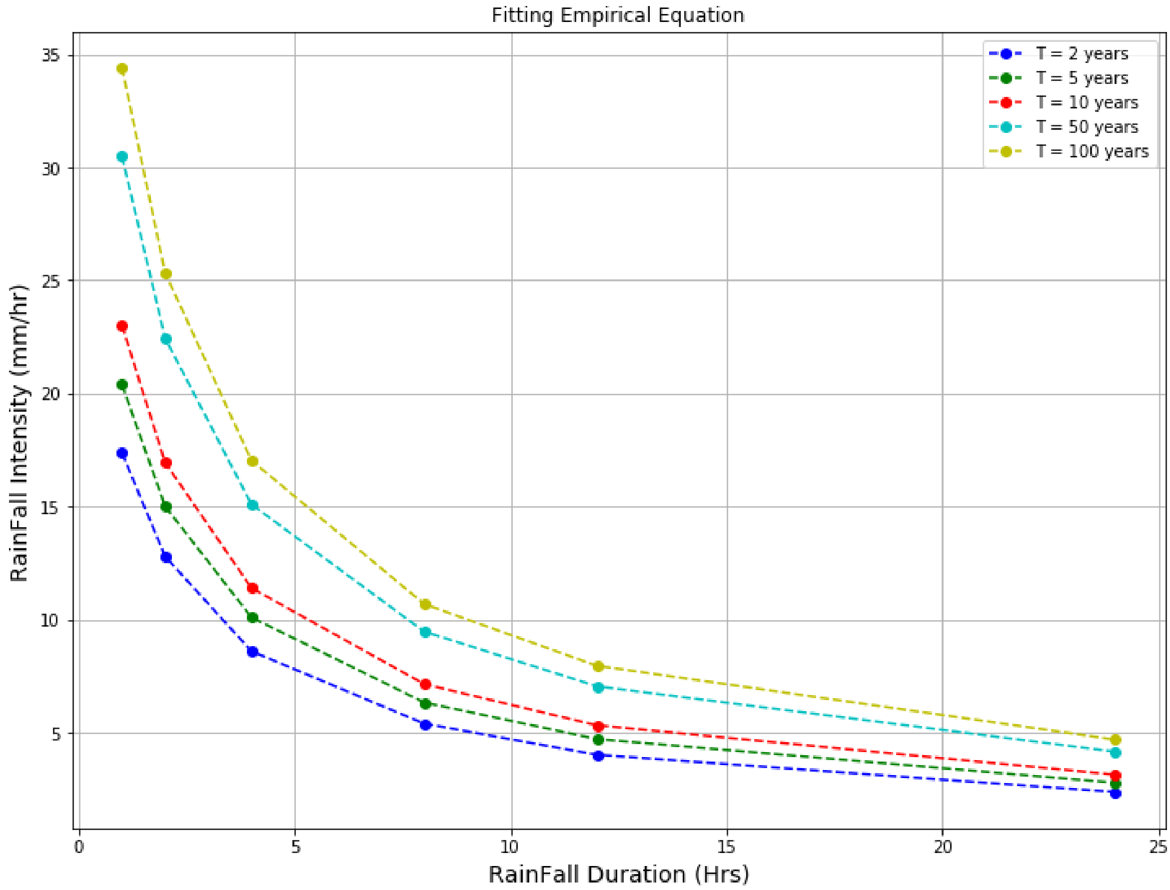
```
fig = plt.figure(figsize=(12, 9))
ax = plt.subplot(111)
Duration = np.array([1,2,4,8,12,24])
ax.plot(Duration,FinalData[:,0],linestyle='--', marker='o', color='b',label = 'T =
ax.plot(Duration,FinalData[:,1],linestyle='--', marker='o', color='g',label = 'T =
ax.plot(Duration,FinalData[:,2],linestyle='--', marker='o', color='r',label = 'T =
ax.plot(Duration,FinalData[:,3],linestyle='--', marker='o', color='c',label = 'T =
ax.plot(Duration,FinalData[:,4],linestyle='--', marker='o', color='y',label = 'T =
# plt.plot(np.arange(1,25,1), 10*f(np.arange(1,25,1)), 'o-', 'k')

plt.ylabel('RainFall Intensity (mm/hr)', fontsize=14)
plt.xlabel('RainFall Duration (Hrs)', fontsize=14)
ax.legend()
plt.grid()
plt.savefig('line_plot.pdf')
plt.show()
```



USing Empirical Formula

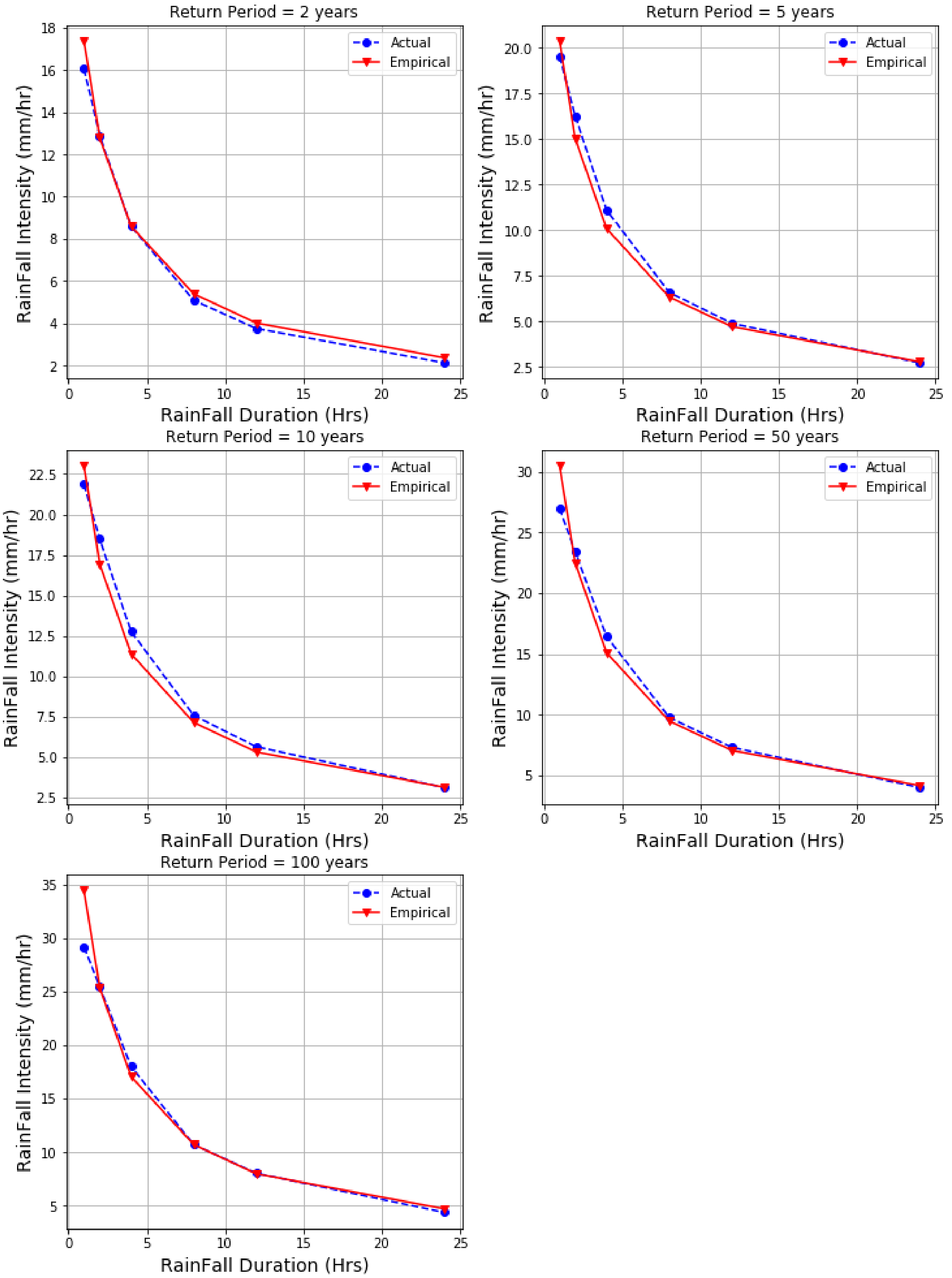
In [319]:

```
fig = plt.figure(figsize=(12, 9))
ax = plt.subplot(111)
Duration = np.array([1,2,4,8,12,24])
ax.plot(Duration,10*f(Duration,2),linestyle='--', marker='o', color='b',label = 'T
ax.plot(Duration,10*f(Duration,5),linestyle='--', marker='o', color='g',label = 'T
ax.plot(Duration,10*f(Duration,10),linestyle='--', marker='o', color='r',label = 'T
ax.plot(Duration,10*f(Duration,50),linestyle='--', marker='o', color='c',label = 'T
ax.plot(Duration,10*f(Duration,100),linestyle='--', marker='o', color='y',label = '
plt.ylabel('RainFall Intensity (mm/hr)', fontsize=14)
plt.xlabel('RainFall Duration (Hrs)', fontsize=14)
ax.legend()
ax.set_title('Fitting Empirical Equation')
plt.grid()
plt.savefig('Empirical.pdf')
plt.show()
```



In [331]:

```
Duration = np.array([1,2,4,8,12,24])
plt.figure(figsize=(12,17))
ax = plt.subplot(3,2,1)
ax.plot(Duration,FinalData[:,0],linestyle='--', marker='o', color='b',label = 'Actu
ax.plot(Duration,10*f(Duration,2),linestyle='-', marker='v', color='r',label = 'Emp
plt.ylabel('RainFall Intensity (mm/hr)', fontsize=14)
plt.xlabel('RainFall Duration (Hrs)', fontsize=14)
ax.legend()
plt.grid()
ax.set_title("Return Period = 2 years")
ax = plt.subplot(3,2,2)
ax.plot(Duration,FinalData[:,1],linestyle='--', marker='o', color='b',label = 'Actu
ax.plot(Duration,10*f(Duration,5),linestyle='-', marker='v', color='r',label = 'Emp
plt.ylabel('RainFall Intensity (mm/hr)', fontsize=14)
plt.xlabel('RainFall Duration (Hrs)', fontsize=14)
ax.legend()
plt.grid()
ax.set_title("Return Period = 5 years")
ax = plt.subplot(3,2,3)
ax.plot(Duration,FinalData[:,2],linestyle='--', marker='o', color='b',label = 'Actu
ax.plot(Duration,10*f(Duration,10),linestyle='-', marker='v', color='r',label = 'Em
plt.ylabel('RainFall Intensity (mm/hr)', fontsize=14)
plt.xlabel('RainFall Duration (Hrs)', fontsize=14)
ax.legend()
ax.set_title("Return Period = 10 years")
plt.grid()
ax = plt.subplot(3,2,4)
ax.plot(Duration,FinalData[:,3],linestyle='--', marker='o', color='b',label = 'Actu
ax.plot(Duration,10*f(Duration,50),linestyle='-', marker='v', color='r',label = 'Em
plt.ylabel('RainFall Intensity (mm/hr)', fontsize=14)
plt.xlabel('RainFall Duration (Hrs)', fontsize=14)
ax.legend()
ax.set_title("Return Period = 50 years")
plt.grid()
ax = plt.subplot(3,2,5)
ax.plot(Duration,FinalData[:,4],linestyle='--', marker='o', color='b',label = 'Actu
ax.plot(Duration,10*f(Duration,100),linestyle='-', marker='v', color='r',label = 'E
plt.ylabel('RainFall Intensity (mm/hr)', fontsize=14)
plt.xlabel('RainFall Duration (Hrs)', fontsize=14)
ax.legend()
ax.set_title("Return Period = 100 years")
plt.grid()
plt.savefig('compare.pdf')
```


Return Period = 2 years
RainFall Intensity (mm/hr)
RainFall Duration (Hrs)
Actual
Empirical
Return Period = 5 years
RainFall Intensity (mm/hr)
RainFall Duration (Hrs)
Actual
Empirical
Return Period = 10 years
RainFall Intensity (mm/hr)
RainFall Duration (Hrs)
Actual
Empirical
Return Period = 50 years
RainFall Intensity (mm/hr)
RainFall Duration (Hrs)
Actual
Empirical
Return Period = 100 years
RainFall Intensity (mm/hr)
RainFall Duration (Hrs)
Actual
Empirical