Università degli Studi di Trento

DIPARTIMENTO DI INGEGNERIA CIVILE AMBIENTALE E MECCANICA
Corso di Laurea magistrale in Ingegneria Civile

Relazione di costruzioni idrauliche

Determinazione delle curve di possibilità pluviometrica: stazione S.Massenza

Studenti:
Gianordoli Sebastiano - 190956
Pederzolli Massimo - 110396
Mosquera Diego - 117307

Docente:
Rigon Riccardo

Anno Accademico 2016– 2017

Indice

Elenco delle figure

Elenco delle tabelle

Elenco dei codici

Introduzione

L'obiettivo di questo elaborato è confrontare le altezze di massima precipitazione annuale per diverse durate tracciando le curve di possibilità pluviometrica LSPP relative a diversi tempi di ritorno. I dati così ottenuti saranno utili per il dimensionamento delle fognature pluviali.

1.1 Descrizione della stazione

La stazione meteorologica di riferimento denominata'Santa Massenza Centrale' (T189) è situata presso l'omonimo paese in Valle dei Laghi. I dati relativi ai massimi di precipitazione cumulata sono stati tratti dal sito internet *meteotrentino* seguendo il percorso 'dati meteo' - 'stazioni meteo', ora è possibile scegliere due vie tramite 'elenco stazioni' oppure tramite mappa 'Mappa stazioni con google map'. Selezionata la stazione di interesse, nella sezione 'output predefiniti' è scaricabile il file 'Massimi di precipitazione' in formato XLS. I rilevamenti partono dal 1968 con sufficiente continuità. Nel corso degli anni la stazione è stata spostata fino a raggiungere la posizione attuale nel 2012. Lo spostamento legato a motivi strettamente organizzativi, risulta limitato e tale da non compromettere la validità dei dati raccolti. Per la scelta della stazione sono stati seguiti alcuni criteri fondamentali:

- sviluppo temporale di almeno 25-30 anni, ossia almeno il triplo del tempo di ritorno per la pregettazione di una fognatura pluviale
- buona continuità nell'intervallo temporale considerato
- recente intervallo di misurazione al fine di ottenere dati attendibili riguardo l'evoluzione degli eventi meteorologici

I dati della stazione sono riassunti nella tabella seguente:

Figura 1.1: Stazione T189

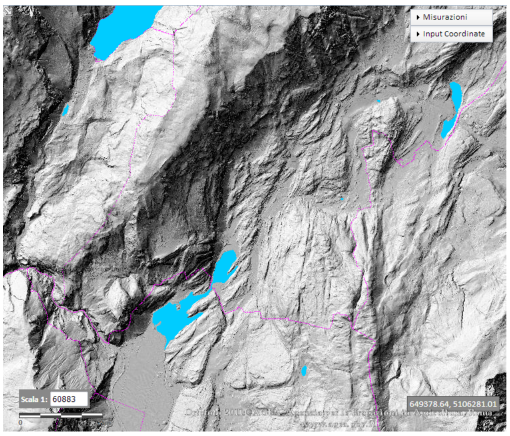
Codice	T 189
Tavoletta n.	32 059120
Cordinata N	5103205
Cordinata E	653156
Latitudine	$46°03'54''.9$ N
Longitudine	$10°58'49''.0$ E
Quota	252 mslm
Regione	Trentino
Note	Attiva

Tabella 1.1: Dati T189

(a) *Posizione della stazione.*

(b) *Valle dei Laghi con posizione della stazione.*

(c) *Valle dei Laghi vista nord.*



(d) *Esposizione solare mappa lidar.*

(e) *S.Massenza zona stazione.*

Figura 1.2: Alcune immagini illustrative sulla stazione S.Massenza

1.2 Descrizione morfologica e climatica

La Valle dei laghi si sviluppa parallelamente alla valle dell'Adige ed è la naturale prosecuzione della zona nord del lago di Garda. E' caratterizzata dalla presenza di ben 7 laghi che ne influenzano il clima che varia da prealpino a mediterraneo con sensibili differenze tra fondo valle e zone montane. La temperatura media annua è di 15 C, mentre le precipitazioni si assestano su 800-900 millimetri all'anno. Tutta la zona risente grandemente degli effetti del lago di Garda sia per quanto riguarda la temperatura che la piovosità, in particolare si ricorda la presenza del ora ossia un vento caldo con direttrice sud-nord, costantemente presente durante i lunghi pomeriggi estivi.

1.3 Strumenti di lettura ed elaborazione dati

Per l'analisi e l'elaborazione dei dati è stato utilizzato il codice di programmazione *Python* in modalità integrata ad un'interfaccia grafica denominata *notebook jupiter*. Nel corso della trattazione sono stati inseriti alcuni frammenti di codice significativi per illustrare più chiaramente la procedura di calcolo.

1.4 Sistemazione dei dati

La tabella dei dati ottenuta dal sito è stata ordinata e inserita in un dataframe in modo da poter essere visualizzata ed elaborata. Di seguito si riportano alcune operazioni preliminari:

```
import pandas as pd
xl_file = pd.ExcelFile('massimi1.xls')
dfs = pd.read_excel('massimi1.xls', sheetname='DATI')
xl_file
dfs
```

Codice 1.1: Esempio di caricamento dataframe

Successivamente i dati sono stati ordinati e puliti mentre gli spazi vuoti sono stati sostituiti dal valore numero nan.

```
df1 = pd.DataFrame(df.iloc[:,[0,7,9,11,13,15]])
#crea variabile con colonne da eliminare
df1.dropna(how='all') #elimina colonne
df22 = pd.DataFrame(df1.replace(r'\␣s+', np.nan, regex=True))
df22
#sostituisco valori  mancanti con nan
```

Codice 1.2: Riordino del dataframe

In conclusione il dataframe ordinato é stato esportato in un file di testo per essere elaborato con lo script successivo.

```
df312.to_csv('C:\\Users\\Administrator\\Desktop\\np3.txt', header=None,
 index=None, mode='w', sep=',')
```

Codice 1.3: Export file .csv

A questo punto il dataframe viene indicizzato in modo tale che ogni riga sia identificata con l'anno del rilevamento grazie al comando presente nella libreria datetime. In questo modo si rende più chiara la rappresentazione dei dati.

```
from pandas import *
import pandas as pd
data = pd.read_csv('np3.txt')
import datetime as datetime
map(datetime,data['anno'])
dataa=data.set_index('anno')
```

Codice 1.4: Indicizzazione del dataframe

Nella tabella sottostante sono riportate le massime altezze di precipitazione, estratte dagli annali idrologici, espresse in funzione di diverse durate di evento meteorologico. Sono determinate tramite rilevamento con cadenza di 10 minuti, utilizzando una finestra temporale mobile di ampiezza corrispondente alla durata considerata.

Anno	1 ora	3 ore	6 ore	12 ore	24 ore
1968	28.1	34.8	39.9	47.8	65
1969	21.8	33.4	39.8	54.8	69.2
1970	33.6	36.2	51	89	106.6
1971	16	27.2	42.4	70.2	98.6
1972	23	40.6	44.6	46.2	80.6
1973	21	25.2	26.4	42.6	48.8
1974	21.6	23.4	39.6	52.4	56.2
1975	20.6	27	41.2	51	89
1976	22	36	49	82.4	90.6
1977	20	43	61	81	110.4
1978	20	22.4	32	43.2	67.4
1979	39.6	84	101.4	113.6	145
1980	18	27.6	43	53.6	54
1981	43.2	45.6	45.6	45.8	60
1982	17.4	23.6	42.8	64.8	84.8
1983	17.4	29.6	39	57.4	60.8
1984	24	39.4	45.2	77.2	119.4
1985	20.8	23	37.6	46.2	96
1986	10	14	17.6	30	40.6
1987	19	35.4	48	48.6	60.4
1988	15.4	31.8	46.2	47.6	60.8
1989	13.8	32.6	37	42.8	64.2
1990	13.8	27.4	37.8	64.8	92
1991	22.8	40.4	60.4	84.6	126.8
1992	15	28.8	40.2	62	76.6
1993	21	23.4	36	58.6	79.4
1994	28.4	31	57.8	80	138.6
1995	19.2	42	82.4	116.6	146.2
1996	16.2	37.8	50.6	80	118.8
1997	32.8	36.2	40.8	67.2	106.2
1998	17	29	42.4	60	64.2
1999	18.8	21.4	40	47.4	51.2
2000	20.2	41.4	48	51.6	72.8
2003	22.6	24	38.4	62.8	97
2004	30.2	43.8	48	58.6	78.8
2005	25.6	39	66	88	102.6
2006	21	42.6	43	50.6	67.4
2007	17.4	23.2	42.2	52.6	58
2008	22.8	40.8	47.8	76.8	95.8
2009	48.2	71.4	73.8	74	84.2
2010	17.4	28.4	50.6	85.6	124
2011	25.8	34.2	35.8	54	71.8
2012	16.2	32.6	52.4	62.4	97.2

Tabella 1.2: Precipitazioni massime [mm]

1.5 Rappresentazione e verifica della coerenza

Per iniziare la libreria matplotlib permette la visualizzazione grafica dell'intero dataframe o di una sua parte specifica.

```
import matplotlib.pyplot as plt
\%matplotlib inline
#ho caricato la libreria
ax=dataa.plot(title="Pioggie␣stazione␣S.Massenza",style="o-")
ax.set_ylabel("Precipitazione␣[mm]")
#stampa di tutto l'intervallo temporale dei dati
ax=dataa.plot(title="Pioggie␣a␣Paperopoli",style="o-")
#'o' (solo dei pallini in corrispondenza dei dati)
ax.set_ylabel("Precipitazione␣[mm]")
plt.axis([1980, 1985, 0, 100])
plt.show()
plt.savefig('plot1')# salva l'immagine stampata
#stampa intervallo dei dati da 1980 a 1985
```

Codice 1.5: Stampa dei dati tramite matplotlib

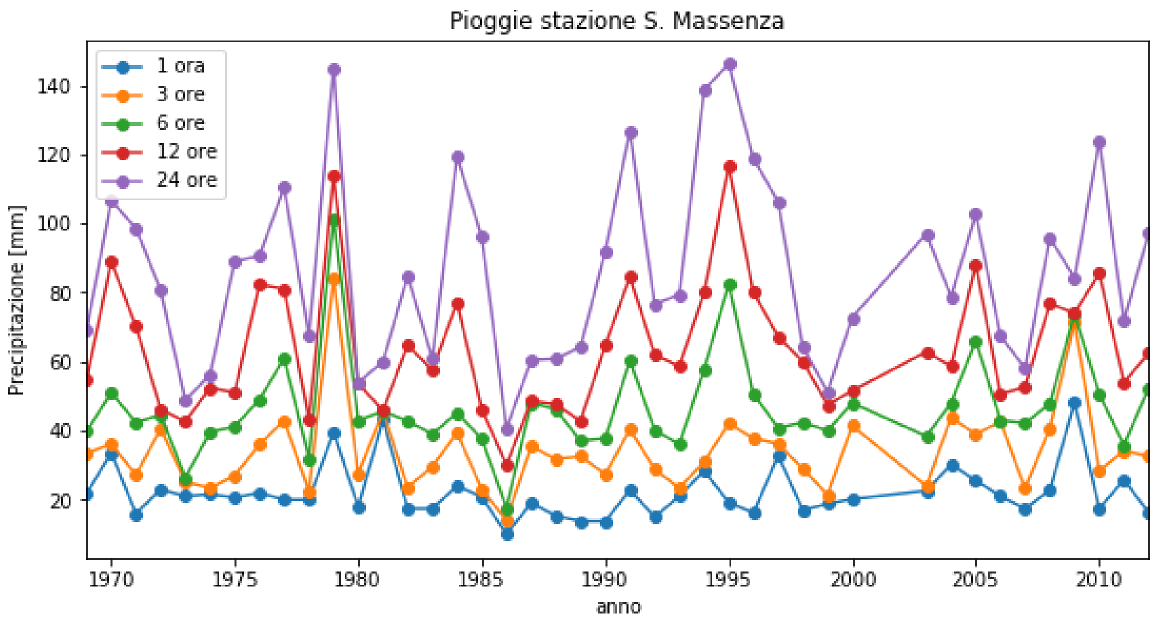


Figura 1.3: Dati stazione S.Massenza

Prima di proseguire nell'elaborazione è necessario verificare la coerenza dei dati tramite l'osservazione dellle medie relative alle varie durate di precipitazione che devono risultare crescenti; inoltre per durate maggiori devono verificarsi altezze di precipitazione maggiori così come risulta dalla stampa seguente:

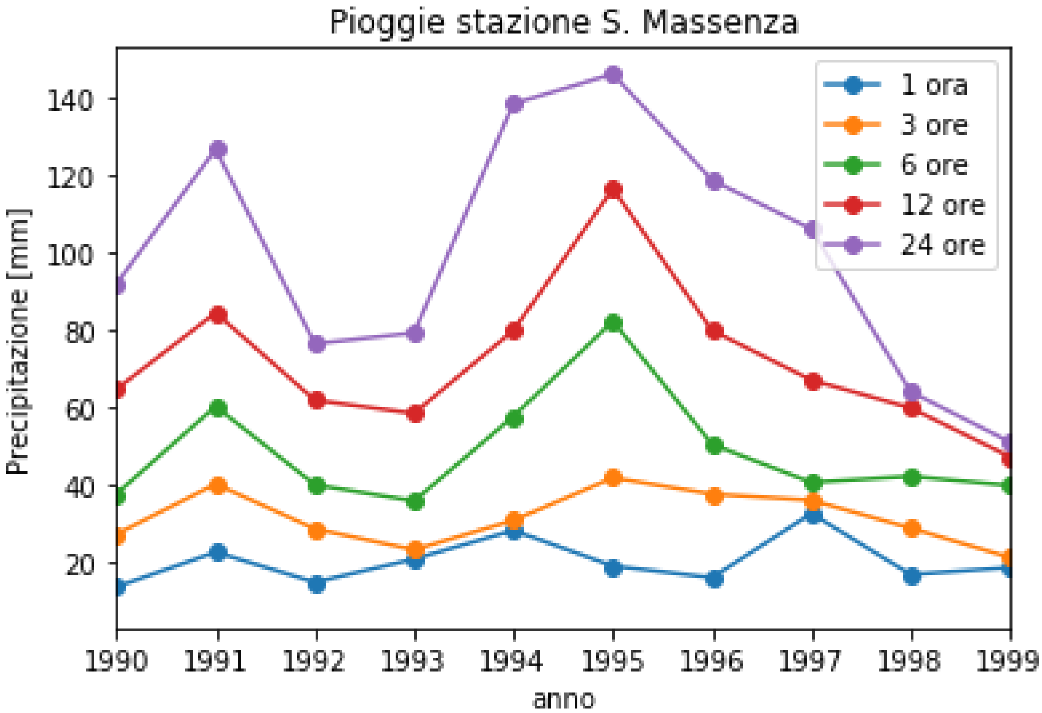


Figura 1.4: Dati stazione da 1990 a 1999

La visualizzazione tramite boxplot permette un ulteriore controllo visivo.

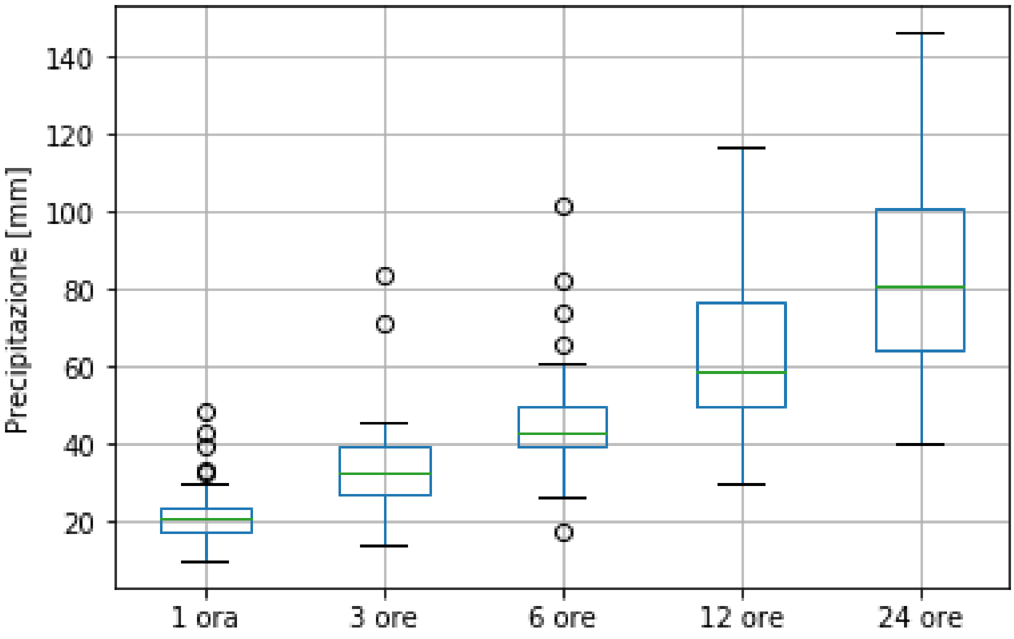


Figura 1.5: Visualizzazione tramite boxplot

Il boxplot rappresenta per ogni durata 5 valori numerici rappresentativi ovvero: minimo, primo quartile, media, terzo quartile e massimo. Questa rappresentazione, a differenza del'istogramma, non dipende dalla classe di estremi scelti ed è per questo univoca. Nel caso in esame i dati risultano quindi attendibili, infatti le medie crescono con le durate. Le misurazioni esterne al range sono dette outliers e derivano probabilmente da errori di misura. In aggiunta tramite il comando *nome.describe()* si ricavano i parametri del campione riportati nella tabella seguente:

Durata	N.misure	Media	Dev.std	Min	1 quartile	Mediana	2 quartile	Max
1 ora	43.0	22.29	7.742	10.00	17.40	20.80	23.50	48.20
3 ore	43.0	34.29	12.30	14.00	27.10	32.60	39.90	84.00
6 ore	43.0	46.62	14.21	17.6	39.7	43.00	49.80	101.40
12 ore	43.0	63.41	18.66	30.01	49.62	58,12	77.00	116,61
24 ore	43.0	85.53	26.92	40.63	64.21	80.60	100.66	146.20

Un'altra rappresentazione interessante è quella del dynamite plot ossia un grafico a barre che rappresenta la media per ogni durata a cui è sovrapposto lo scarto quadratico medio.

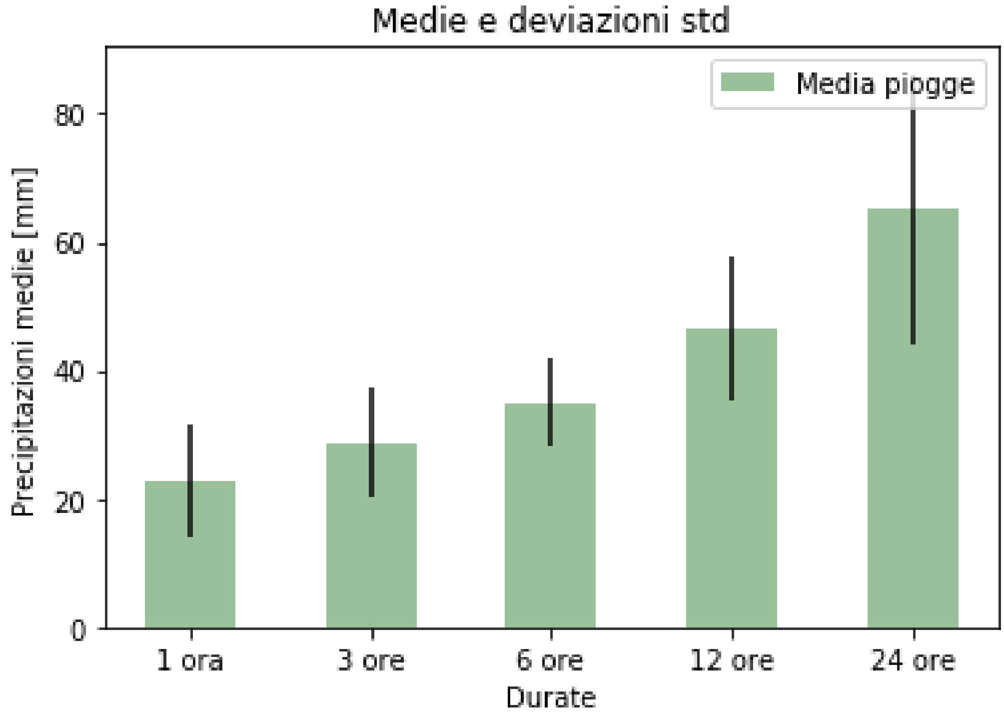
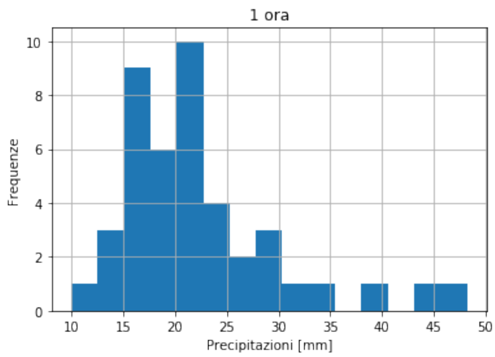


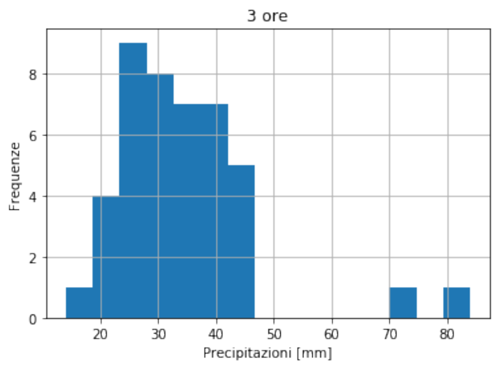
Figura 1.6: Visualizzazione tramite dynamite plot

1.6 Distribuzioni

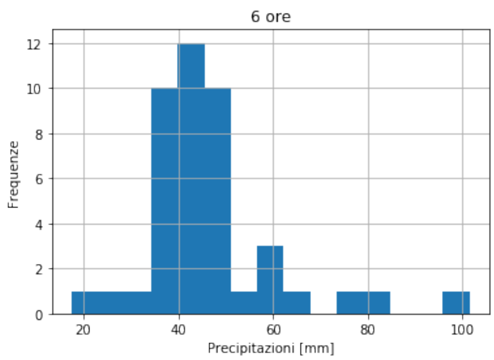
Di seguito si riporta una rappresentazione qualitativa delle frequenze relative alle altezze di precipitazione per durate fissate. La rappresentazione tramite istogrammi prevede lungo le ascisse le altezze di precipitazione e lungo le ordinate le rispettive frequenze.



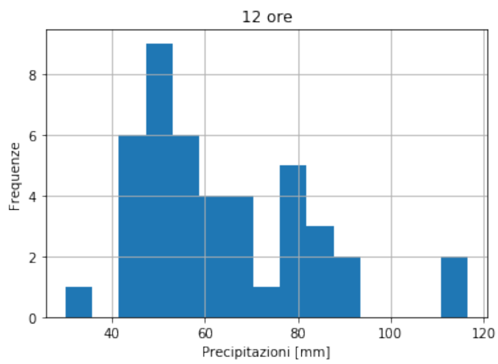
(a) *Durata 1 ora.*



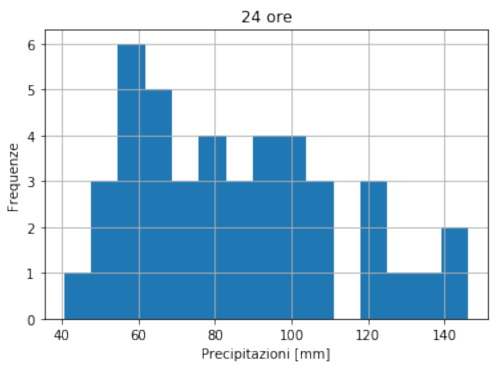
(b) *Durata 3 ore.*



(c) *Durata 6 ore.*



(d) *Durata 12 ore.*



(e) *Durata 24 ore.*

Figura 1.7: Istogrammi delle frequenze

Cenni teorici con frammenti di codice

2.1 Curve di possibilità pluviometrica

Le curve di possibilità pluviometrica forniscono una stima dell'altezza di precipitazione per un'assegnata durata dell'evento e un tempo di ritorno, generalmente sono rappresentate da leggi di potenza:

$$h(t_{\mathrm{p}}, T_{\mathrm{r}}) = a(T_{\mathrm{r}}) * t_{\mathrm{p}}^{n}$$

dove:

- T_{r} indica il tempo di ritorno, ossia il periodo medio in cui una fissata precipitazione è raggiunta o superata
- h indica l'altezza di precipitazione cumulata
- a è un coefficiente legato a T_{r}
- t_{p} è la durata di precipitazione
- n è un numero reale compreso tra zero e uno: n>0 visto che l'altezza è non decrescente all'aumentare della durata, mentre n<1 dato che l'intensità media decresce all'aumentare della durata:

$$J(t_{\mathrm{p}}, T_{\mathrm{r}}) = \frac{h(t_{\mathrm{p}}, T_{\mathrm{r}})}{t_{\mathrm{p}}} = a(T_{\mathrm{r}}) * t_{\mathrm{p}}^{(n-1)} \text{—>}(n-1) < 0 \text{ e quindi } n < 1$$

Per il dimensionamento di una fognatura pluviale è necessaria la determinazione di un'altezza di precipitazione correlata ad un tempo di ritorno. Ad esso deve essere associata una probabilità di non superamento sufficientemente alta in base al tipo di opera e alla vulnerabilità dell'area. In questo caso si considera zona densamente popolata quindi: $T_r = 10$ anni. Il tempo di ritorno è univocamente legato ad una probabilità di non superamento infatti:

$$T_r = \frac{T_t}{L} = \frac{m * n}{L} = \frac{m}{P[H > h_*]} = \frac{m}{1 - P[H < h_*]}$$

dove:

- T_t periodo complessivo delle misurazioni
- L numero di eventi positivi (precipitazioni estreme)
- n numero misurazioni
- m durata di ogni misurazione (1 anno)

Alla luce di quanto appena illustrato si ottiene:

$$P[H < h*] = 1 - \frac{1}{T_r} \text{ quindi per } T_r = 10 \text{ anni ottengo } P = 0.9$$

Per la determinazione delle altezze si sostituisce alla distribuzione di probabilità empirica una funzione nota, continua e in grado di approssimare l'andamento della empirical cumulative distribution function. Per questa trattazione è stata scelta la curva dei valori estremi di tipo 1 detta curva di Gumbel:

$$P[H < h^*; a, b] = e^{-e^{\frac{h-a}{b}}}$$

che ad ogni valore del dominio associa la probabilità di non superamento. La curva di Gumbel è descritta da due parametri: b è il parametro di forma prossimo alla *varianza*, a è invece il parametro di posizione e corrisponde alla *moda*. Per questa distribuzione sono conosciute anche le espressioni dei principali parametri:

- Media o valore atteso: $E[h; a, b] = b * \gamma + a$ dove $\gamma = 0.57721$ è la costante di Eulero-Mascheroni
- Moda = a
- Mediana $Med[h; a, b] = a - b * log(log(2))$
- Varianza Var[h;b]=$b^2 * \frac{\pi^2}{6}$

I parametri a e b devono essere calibrati affinchè la curva di Gumbel descriva più precismente possibile la empirical cumulative distribution function. Tra i molti metodi di adattamento ne sono stati applicati tre:

- Metodo dei momenti
- Metodo dei minimi quadrati
- Metodo della massima verosimiglianza

2.2 Metodo dei momenti

In generale il momento i-esimo di una variabile aleatoria è scrivibile come:

$$E_i[X] = \int_{-\infty}^{+\infty} x^i * f(x)\, dx$$

Il primo e il secondo momento, corrispondenti rispettivamente a $i = 1$ e $i = 2$ coincidono con media e varianza. Il metodo dei momenti consiste nell'eguagiare i primi n momenti del campione con quelli della distribuzione, dove n indica il numero di parametri da calcolare. La curva di Gumbel è biparametrica quindi si ottiene un sistema di due equazioni in due incognite:

$$\begin{cases} \mu_h = b * \gamma + a \\ Var_h = \dfrac{b^2 * \pi^2}{6} \end{cases}$$

In Pyton il sistema è implementabile in questo modo:

```
math.sqrt(6)/math.pi
0.77969680
EulerGamma=0.57721566490153286060651209O
0.779696801233676*EulerGamma
0.450005
def g(m,s):
    return pd.DataFrame([m-0.4500532075456946*s,0.779696801233676*s],
    index=["a","b"])
abs_m=g(means,stds)
abs_m
#I parametri sono inseriti nella variabile abs_m
```

Codice 2.1: Definizione del sistema e calcolo dei parametri

La stampa dei risultati si ottiene tramite il comando plt. In questo caso è necessaria particolare attenzione, infatti per ogni durata devono essere riprodotte non solo le distribuzioni di Gumbel ma anche le rappresentazioni delle funzioni empiriche di probablità, ognuna riferita alla durata corrispondente. Tramite i comandi 'label' è possibile inserire nome degli assi e dimensione dei caratteri. Un esempio di stampa è riportato nella pagina seguente.

```
from scipy.stats import genextreme,gumbel_r
from numpy import linspace
def gumbel(x,abs_d,col):
    return np.exp(-np.exp(-(x-abs_d[col]["a"])/abs_d[col]["b"]))
    from statsmodels.distributions.empirical_distribution import ECDF
data.min()[0]
data.max()[4]
t_rain=np.linspace(data.min()[0],data.max()[4],100)
# ho definito il dominio di stampa
gb=pd.DataFrame([gumbel(t_rain,abs_m,"1␣ora"),
                 gumbel(t_rain,abs_m,"3␣ore"),
                 gumbel(t_rain,abs_m,"6␣ore"),
                 gumbel(t_rain,abs_m,"12␣ore"),
                 gumbel(t_rain,abs_m,"24␣ore")]).T
gb.index=t_rain
gb.columns=["1h","3h","6h","12h","24"]
# ho definito dataframe con tutte le curve di Gumbel
#stampo il dataframe con sovrapposti i dati empirici
plt.rc('xtick', labelsize=10)
plt.rc('ytick', labelsize=10) i
ax=gb.plot(color=["red","blue","cyan","green","black"])
ecdf1h = ECDF(data["1␣ora"])
ax.plot(data["1␣ora"],ecdf1h(data["1␣ora"]),'o',c="red",label="1h")
data3h=data["3␣ore"].dropna()
ecdf3h = ECDF(data3h)
ax.plot(data3h,ecdf3h(data3h),"o",c="blue",label="3h")
data6h=data["6␣ore"].dropna()
ecdf6h = ECDF(data6h)
ax.plot(data6h,ecdf6h(data6h),"o",c="cyan")
data12h=data["12␣ore"].dropna()
ecdf12h = ECDF(data12h)
ax.plot(data12h,ecdf12h(data12h),"o",c="green")
data24h=data["24␣ore"].dropna()
ecdf24h = ECDF(data24h)
ax.plot(data24h,ecdf24h(data24h),"o",c="black")
ax.set_title('Curve␣di␣gumbel␣calcolate␣con␣m.␣momenti')
ax.set_xlabel('Pioggia␣(mm)')
ax.set_ylabel('P[H<h]')
```

Codice 2.2: Definizione e stampa delle curve di Gumbel

2.3 Metodo della massima verosimiglianza

Il metodo della massima verosimiglianza consiste nel massimizzare la funzione di verosimiglianza definita in base alla probabilità di osservare una precisa realizzazione campionaria. Così si ricavano i parametri che approssimano meglio (ossia con la massima probabilità) il campione. Per ipotesi le misurazioni si assumono indipendenti, così la loro probabilità congiunta è pari alla produttoria delle singole funzioni di probabilità calcolate in corrispondenza del singolo dato. La funzione di verosimiglianza sara dunque:

$$P[h_1, h_2, ..., h_n; a, b] = \prod_{i=1}^{n} P[h_i; a, b]$$

Essa rappresenta la distribuzione dei parametri condizionati alle misure e fornisce risultati migliori al crescere della numerosità del campione. Per ricavare quindi la **massima** verosimiglianza è sufficiente imporre la stazionarietà della funzione a cui è applicato il logaritmo per semplificare il calcolo:

$$\begin{cases} \dfrac{d(log(P[h_1,...,h_n];a,b))}{da} = 0 \\ \dfrac{d(log(P[h_1,...,h_n];a,b))}{db} = 0 \end{cases}$$

Il sistema è non lineare, quindi viene risolto tramite metodi numerici come Eulero o Newton. Generalmente i valori iniziali di a e b necessari al calcolo della funzione di verosimiglianza sono ricavati con altri metodi come momenti o minimi qudrati. In pyton il calcolo dei parametri avviene in modo automatico grazie ad uno speciale algoritmo implementato nella libreria scipy (downhill simplex algorithm):

```
samp = data["N␣ore"]
paramNh = gumbel_r.fit(samp) # adattamento della distribuzione
paramNh
#da eseguire per ogni durata considerata
```

Codice 2.3: Calcolo dei parametri a e b

2.4 Metodo dei minimi quadrati

Il metodo consiste nella definizione dello scarto quadratico medio tra le misure di probabilità della distribuzione empirica con la corrispondente probabilità di non superamento calcolata tramite la distribuzione di Gumbel. Questa differenza o scarto è funzione dei parametri della distribuzione e quindi rispetto ad essi può essere minimizzato:

$\delta^2(a,b) = \sum_{i=1}^{n}(F_i - P[H < h_i; a, b])^2$ per cui $$\begin{cases} \dfrac{d(\delta^2(a,b))}{da} = 0 \\ \dfrac{d(\delta^2(a,b))}{db} = 0 \end{cases}$$

Il sistema appena ricavato permette di determinare tutti i parametri della distribuzione, in questo caso 2. In Pyton per prima cosa di definisce la distribuzione di Gumbel, quindi si calcola prima la distribuzione empirica dei dati, poi la funzione di probabilità appena definita. La loro differenza elevata al quadrato è minimizzata tramite l'algoritmo *least squares* della libreria scipy:

```
from scipy.optimize import least_squares
def fun(x, t, y):
    return np.exp(-np.exp(-(t-x[0])/x[1]))-y
ecdf1h = ECDF(data["n␣ora"])
t_train=sorted(data["n␣ora"])
y_train=ecdf1h(t_train)
x0=[30.,9.]
res_lsq_1h = least_squares(fun, x0, args=(t_train, y_train))
#applico l'algoritmo ai dati sperimentali
# e attesi ordinati dal comando sorted
```

Codice 2.4: Calcolo dello scarto quadratico

2.5 Test di Pearson o del χ^2

Questo test permette di provare la validità dell'ipotesi secondo cui una data variabile casuale (dati pluviometrici) sia distribuita secondo un'assegnata funzione di probabilità (Gumbel), comunemente si parla di test di adattamento. Il test valuta una specifica funzione χ^2 per ogni coppia di parametri della distribuzione. In generale si ha che:

$$\chi^2 = \sum_i \frac{(osservato - atteso)_i^2}{atteso_i}$$

in cui "osservato" rappresenta il numero di dati empirici contenuti in un intervallo di probabiltà fissato i, mentre "atteso" rappresenta l'equivalente calcolato tramite la distribuzione nota. In termini più formali si ha:

$$\chi^2 = \sum_{i=1}^{k} \frac{n_i - N * (P(H < h_{i+1}) - P(H < h_i))^2}{N * (P(H < h_{i+1}) - p(H < h_i))}$$

dove:

- k rappresenta il numero di suddivisioni del campo di probabilità
- N indica la numerosità del campione
- n_i numero di dati osservati per intervallo.
- $N * p =$ numero atteso di dati nell'intervallo i-esimo

Per l'implementazione in pyton si parte del definire gli 'oggetti' che compongono la funzione χ^2:

```
q=[0.2,0.4,0.6,0.8,1]
h1h=rv.ppf(q)
ecdfnh = ECDF(data["n␣ora"])
r=ecdf1h(h1h)
```

Codice 2.5: Calcolo numero dati ottenuti per ogni intervallo di misura

Ci si aspetta che, in ogni intervallo, con limite destro h1h, ci siano $e = (l1h * q)$ elementi mentre ce ne sono in realtà $o = (l1h * r)$. La formula dell'X^2 è allora:

$$X^2 = \frac{1}{e} \sum_{i=1}^{k} (e - o)^2 \tag{2.1}$$

```
o0=lnh*r
o=o0-np.append([0],np.delete(o0,-1))
e=[0.2*lnh for i in range(len(r))]
((o-e)**2/e).sum()
def X2(data,abs_t,h,delta):
    dt=data[h].dropna()
    lh=len(dt)
    q=[delta*(i+1) for i in range(int(1/delta))]
    rv=gumbel_r(loc=abs_t.at["a",h],scale=abs_t.at["b",h])
    ecdf = ECDF(dt)
    r=ecdf(rv.ppf(q))
    o0=lh*r
    o=o0-np.append([0],np.delete(o0,-1))
    e=[delta*lh for i in range(len(r))]
    return ((o-e)**2/e).sum()
```

Codice 2.6: Calcolo della test statistic e χ^2

La migliore inferenza statistica secondo questo test sarà quella a cui corrisponde il χ^2 minore.

2.6 Stima delle curve di possibilità pluviometrica

Dopo aver applicato l'inferenza statistica e verificato la sua accettabilità con il test di affidabilità è possibile tracciare le curve di possibilità climatica. Interpolando le cinque migliori curve di Gumbel, ognuna definita per una durata diversa si ottengono altrettanti punti a cui corrisponde univocamente una coordinata nel piano altezza di precipitazione-durata evento:

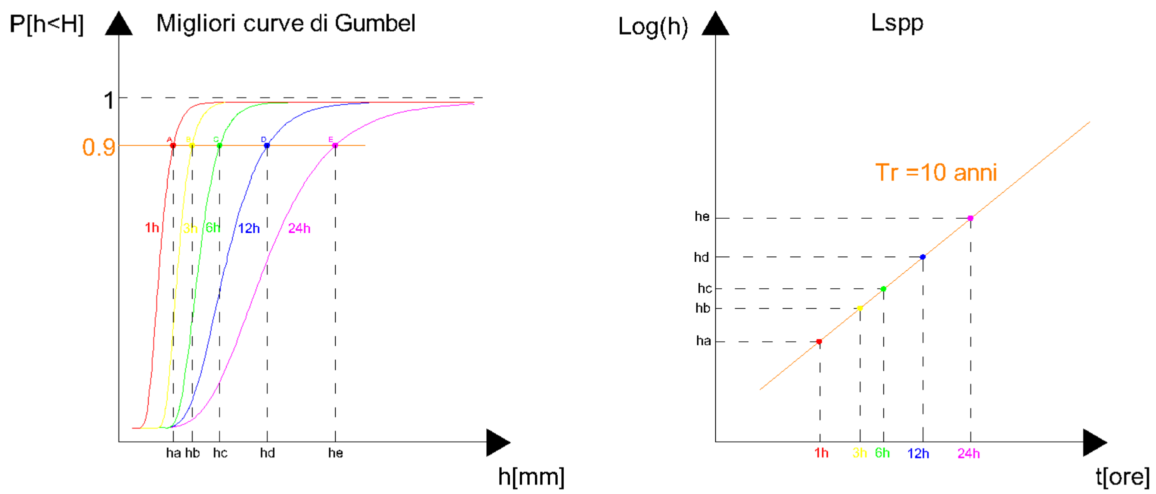


Figura 2.1: Esempio di determinazione Lspp

Risultati

3.1 Curve di Gumbel calcolate tramite metodo dei momenti

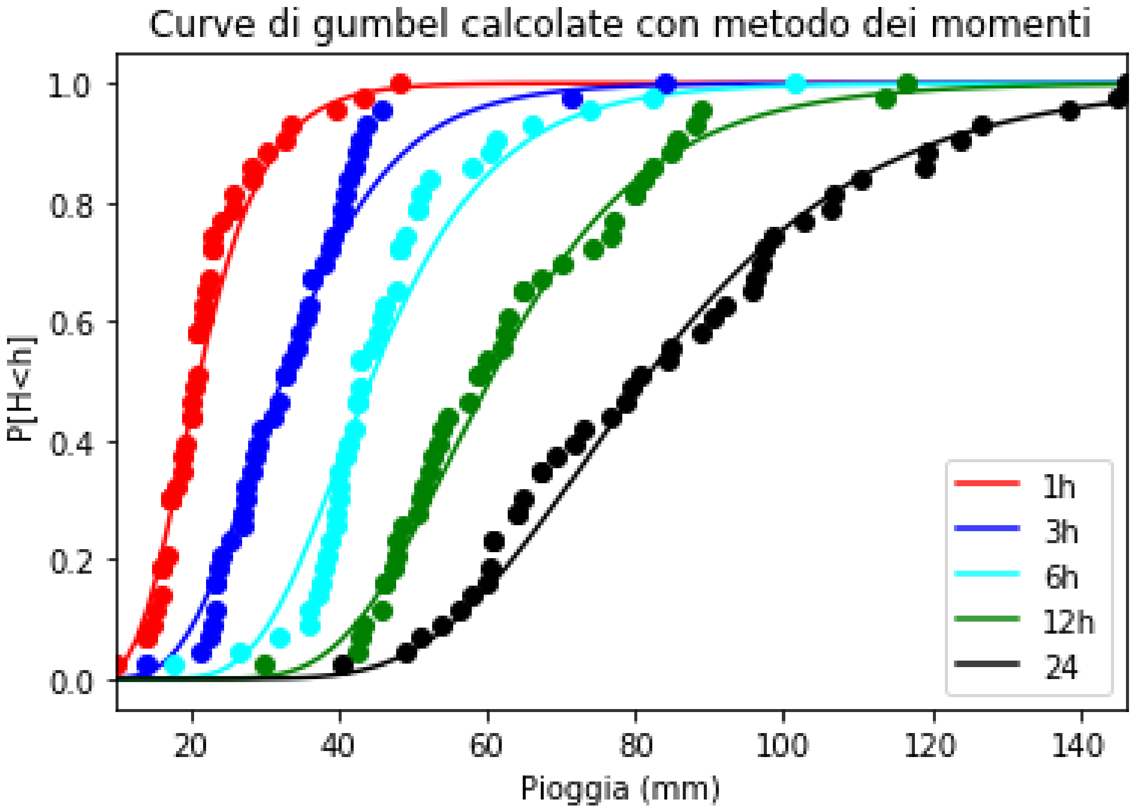


Figura 3.1: ECDF e curve di Gumbel calcolate con metodo dei momenti

Parametro	1 ora	3 ore	6 ore	12 ore	24 ore
a	18.811037	28.756146	40.226527	55.007352	73.417647
b	6.036412	9.592389	11.078015	14.547940	20.992565

Tabella 3.1: Parametri adattati delle curve di Gumbel tramite metodo dei momenti

3.2 Curve di Gumbel calcolate tramite massima verosimiglianza

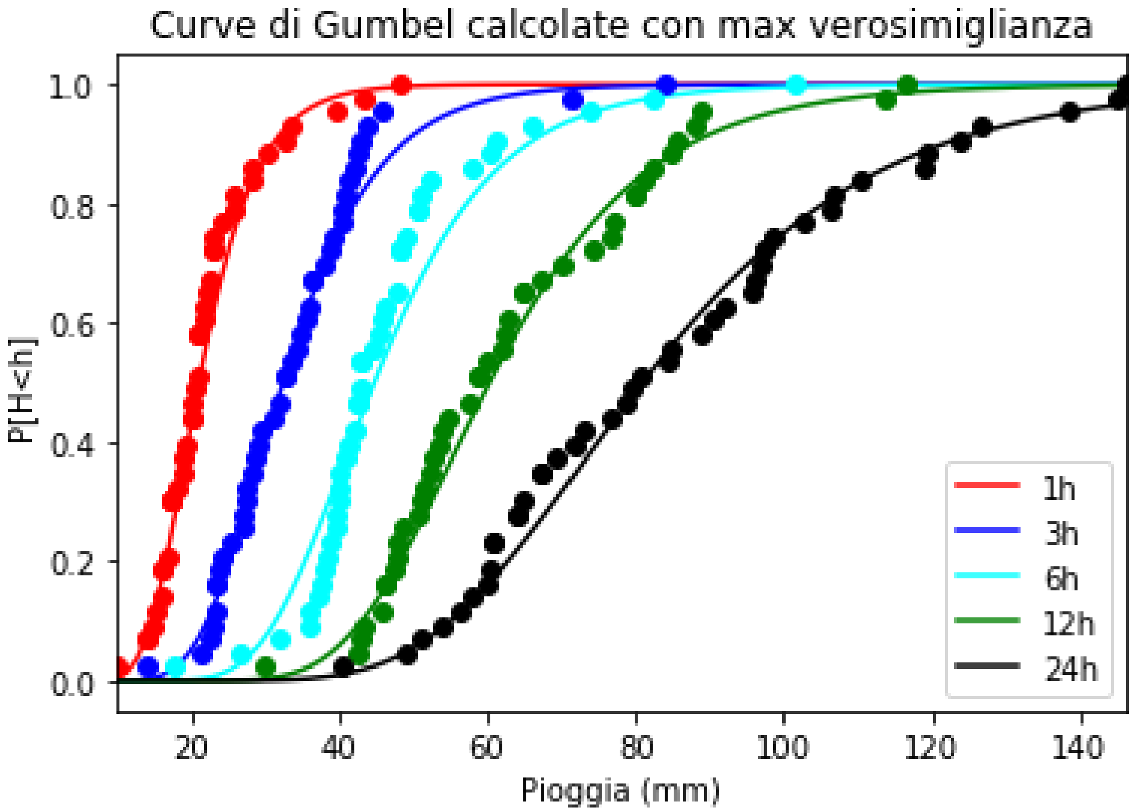


Figura 3.2: ECDF e curve di Gumbel calcolate con massima verosimiglianza

Parametro	1 ora	3 ore	6 ore	12 ore	24 ore
a	19.075870	29.292245	40.637668	55.042028	73.032842
b	5.306906	8.535853	11.050835	14.376372	21.594481

Tabella 3.2: Parametri adattati delle curve di Gumbel tramite massima verosimiglianza

3.3 Curve di Gumbel calcolate tramite minimi quadrati

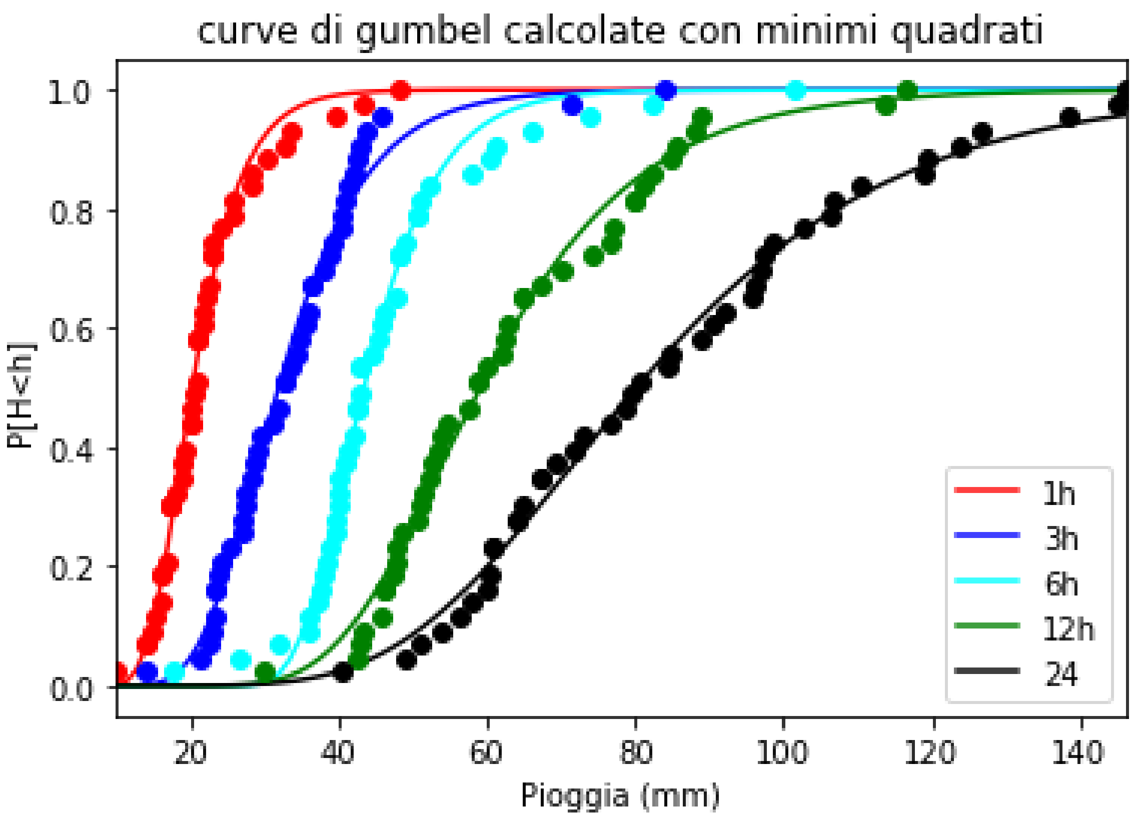


Figura 3.3: ECDF e curve di Gumbel calcolate con minimi quadrati

Parametro	1 ora	3 ore	6 ore	12 ore	24 ore
a	18.590688	28.954455	40.973747	53.809066	71.4188
b	4.519153	7.974466	6.643395	14.475850	23.8540

Tabella 3.3: Parametri adattati delle curve di Gumbel tramite minimi quadrati

3.4 Curve di Gumbel ottimali

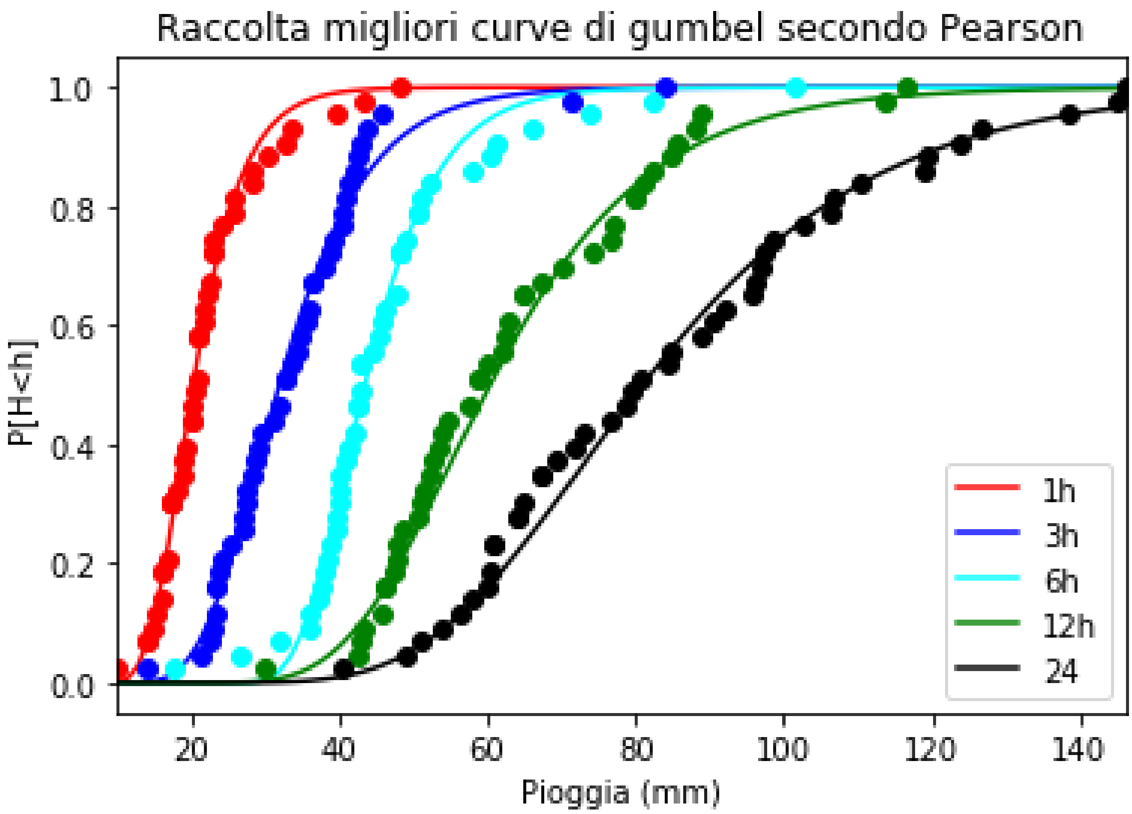


Figura 3.4: Curve di Gumbel ottimali per tutte le durate

Parametro	1 ora	3 ore	6 ore	12 ore	24 ore
χ^2 m. momenti	8.744186	5.953488	6.883721	2.000000	1.302326
χ^2 min quadrati	1.534884	1.534884	0.139535	4.790698	2.465116
χ^2 max verosimiglianza	2.232558	2.697674	12.232558	3.395349	1.302326
a	18.590688	28.954455	40.973747	55.007352	73.032842
b	4.519153	7.974466	6.643395	14.547940	21.594481

Tabella 3.4: Parametri ottimali delle curve di Gumbel per ogni durata scelti con χ^2minore

3.5 Lspp: piano cartesiano

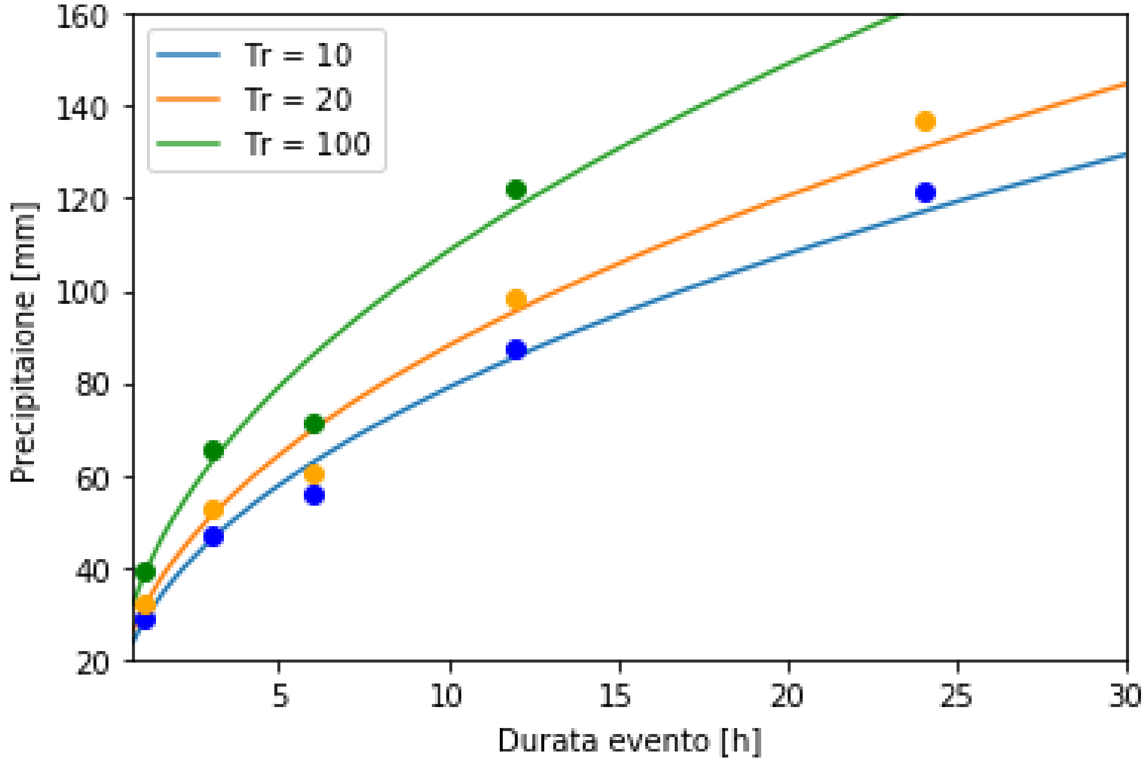


Figura 3.5: Curve di possibilità pluviometrica nel piano cartesiano

Tempo di ritorno	Altezza di precipitazione [mm]				
	1h	**3h**	**6h**	**12h**	**24h**
10 anni	28.76	46.90	55.92	87.75	121.63
20 anni	32.01	52.64	60.71	98.22	137.17
100 anni	39.38	65.64	71.53	121.93	172.37

Tabella 3.5: Altezze di precipitazione per dati tempi di ritorno per ogni durata

3.6 Lspp: piano logaritmico

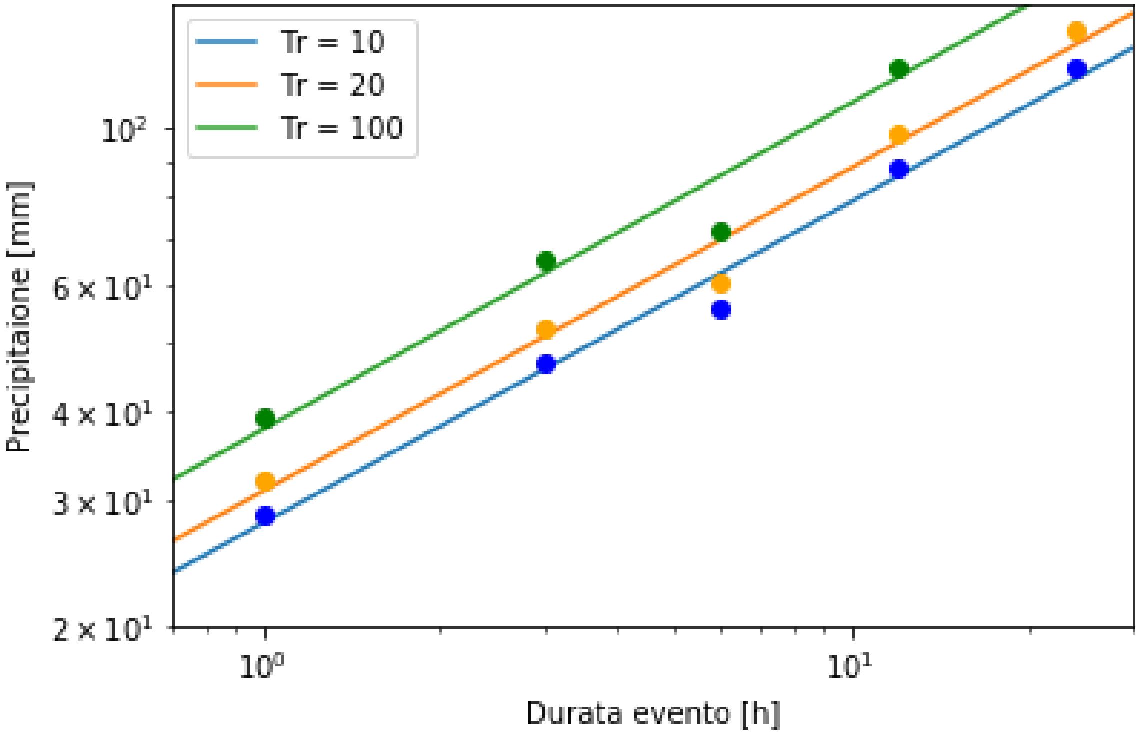


Figura 3.6: Curve di possibilità pluviometrica nel piano logaritmico

Bibliografia

[1] Diapositive del corso di Costruzioni Idrauliche, *Prof. Riccardo Rigon 2017*

[2] Pyton notebook: *version 3*
http://www.pyton.org

[3] Abouthydrology:*extreme rainfall distribution estimation with Pyton*
http://www.abouthydrology.blogspot.it

[4] TeX live 2016 - texworks compilator
http://www.tex.org