



UNIVERSITA' DEGLI STUDI DI TRENTO

Corso di Laurea in Ingegneria per l'Ambiente ed il Territorio

# Analisi delle Curve di Possibilità Pluviometrica

---

Corso di Idrologia

**Docente: Riccardo Rigon**

**Autori: Mara Ottoboni, Alice Resconi**

**Anno Accademico 2014-2015**

## INDICE

<b>1. Introduzione</b>	<b>5</b>
<b>2. Analisi dei dati orari</b>	<b>7</b>
2.1 Precipitazioni massime annuali .....	7
2.2 Analisi per 15 minuti .....	11
2.2.1 Metodo dei momenti .....	12
2.2.2 Metodo dei minimi quadrati .....	13
2.2.3 Metodo della massima verosimiglianza .....	14
2.2.4 Test di Pearson .....	14
2.3 Analisi per 1 ora .....	16
2.3.1 Metodo dei momenti .....	17
2.3.2 Metodo dei minimi quadrati .....	17
2.3.3 Metodo della massima verosimiglianza .....	18
2.3.4 Test di Pearson .....	18
2.4 Analisi per 3 ore .....	20
2.4.1 Metodo dei momenti .....	21
2.4.2 Metodo dei minimi quadrati .....	21
2.4.3 Metodo della massima verosimiglianza .....	22
2.4.4 Test di Pearson .....	22
2.5 Analisi per 6 ore .....	24
2.5.1 Metodo dei momenti .....	25
2.5.2 Metodo dei minimi quadrati .....	25
2.5.3 Metodo della massima verosimiglianza .....	26
2.5.4 Test di Pearson .....	26
2.6 Analisi per 12 ore .....	28
2.6.1 Metodo dei momenti .....	29
2.6.2 Metodo dei minimi quadrati .....	29
2.6.3 Metodo della massima verosimiglianza .....	30
2.6.4 Test di Pearson .....	30
2.7 Analisi per 24 ore .....	32
2.7.1 Metodo dei momenti .....	33
2.7.2 Metodo dei minimi quadrati .....	33
2.7.3 Metodo della massima verosimiglianza .....	34
2.7.4 Test di Pearson .....	34

<b>3. Costruzione delle curve di possibilità pluviometrica</b>	<b>36</b>
3.1 Curve di Gumbel .....	36
3.2 Curve di densità di probabilità legate alle curve di Gumbel .....	37
3.3 Interpolazione con tempi di ritorno .....	39
3.3.1 Tempo di ritorno di 5 anni .....	39
3.3.2 Tempo di ritorno di 10 anni .....	39
3.3.3 Tempo di ritorno di 20 anni .....	39
3.4 LSPP .....	41
3.5 LSPP linearizzate .....	42

<b>A. Programmi utilizzati</b>	<b>43</b>
--------------------------------	-----------

<b>B. Riferimenti bibliografici</b>	<b>43</b>
-------------------------------------	-----------

### Elenco delle figure

1. Posizione stazione meteorologica .....	5
2. Plot .....	8
3. Boxplot .....	8
4. Istogrammi relativi ai massimi di precipitazione per le varie durate .....	10
5. Ecdf per 15 minuti .....	11
6. Q-Q Plot per 15 minuti .....	12
7. Ecdf e curva di Gumbel per 15 minuti con parametri ottimali .....	13
8. Test di Pearson per 15 minuti con il metodo della massima verosimiglianza ....	15
9. Ecdf per 1 ora .....	16
10. Q-Q Plot per 1 ora .....	16
11. Ecdf e curva di Gumbel con parametri ottimali .....	17
12. Test di Pearson per 1 ora con il metodo dei minimi quadrati .....	19
13. Ecdf per 3 ore .....	20
14. Q-Q Plot per 3 ore .....	20
15. Ecdf e curva di Gumbel con parametri ottimali .....	21
16. Test di Pearson per 3 ore con il metodo dei minimi quadrati .....	23
17. Ecdf per 6 ore .....	24
18. Q-Q Plot per 6 ore .....	24

19. Ecdf e curva di Gumbel con parametri ottimali .....	25
20. Test di Pearson per 6 ore con il metodo dei momenti .....	27
21. Ecdf per 12 ore .....	28
22. Q-Q Plot per 12 ore .....	28
23. Ecdf e curva di Gumbel con parametri ottimali .....	29
24. Test di Pearson per 12 ore con il metodo dei momenti .....	31
25. Ecdf per 24 ore .....	32
26. Q-Q Plot per 24 ore .....	32
27. Ecdf e curva di Gumbel con parametri ottimali .....	33
28. Test di Pearson per 24 ore con il metodo dei momenti .....	35
29. Curve di Gumbel .....	37
30. Curve di densità di probabilità .....	38
31. Linee Segnalatrici di Possibilità Pluviometrica .....	41
32. Linee Segnalatrici di Possibilità Pluviometrica, grafico bi-logaritmico .....	42

## 1. Introduzione

I dati esaminati in questo elaborato sono stati presi dal portale di Meteotrentino. La zona di studio è quella che interessa la stazione meteorologica T0129 di Trento Laste situata a 312 metri s.l.m.; ad oggi è attiva (palo del vento a 5 metri). La “Strada delle Laste”, termine corrotto di “lastre”, con riferimento ai lastroni di pietra e alle relative cave che caratterizzano la località, attraversa la collina di Trento. In questa zona si trova il noto Santuario della Madonna delle Laste.

### Caratteristiche stazione:

Tavoletta n.: 32 060100

Coordinata Est: 665153

Coordinata Nord: 5104248

Latitudine: 46°04'18.7" N

Longitudine: 11°08'08.4" E



Figura 1: Posizione stazione meteorologica

Poiché i dati forniti sono incompleti si è scelto di considerare un arco temporale di quarant'anni, dall'anno 1951 al 1990. Questi sono stati elaborati mediante l'utilizzo del free-software *RStudio*, al fine di ottenere le curve di possibilità pluviometrica per i seguenti intervalli temporali: quindici minuti, un'ora, tre ore, sei ore, dodici ore e ventiquattro ore.

Di seguito si riportano le serie temporali dei dati per la stazione di Trento (Laste) relativi agli Annali Idrologici e rappresentanti i massimi di precipitazione per le durate prefissate. Con NA si identificano dati non registrati.

anno	15 minuti	1 ora	3 ore	6 ore	12 ore	24 ore
1951	NA	40.0	62.0	79.0	99.0	116.4
1952	NA	26.6	30.0	43.0	58.6	95.6
1953	NA	16.8	32.8	40.4	67.0	104.6
1954	NA	13.4	29.0	29.8	41.0	49.0
1955	NA	12.8	16.0	25.0	38.4	47.6
1956	NA	11.0	20.6	26.0	43.0	56.8
1957	13.8	20.8	23.6	27.2	35.8	64.6
1958	NA	16.8	21.0	33.2	38.0	55.6
1959	11.6	20.4	26.0	47.2	68.4	109.6
1960	NA	22.8	46.2	59.8	80.8	96.8
1961	NA	23.0	30.2	38.6	52.4	69.6
1962	NA	29.0	43.0	54.8	56.2	59.2
1963	NA	20.0	26.0	28.6	43.6	55.4
1964	NA	12.6	18.6	31.6	50.0	74.6
1965	11.6	14.2	22.0	42.6	61.6	113.0
1966	NA	19.0	31.0	51.0	81.4	125.6
1967	NA	12.6	14.0	26.0	37.4	44.4
1968	8.4	22.2	31.2	44.2	44.6	67.4
1969	9.8	19.4	38.2	38.8	65.6	77.2
1970	14.0	22.4	34.6	46.4	51.6	69.8
1971	14.0	17.0	21.8	27.8	40.0	73.6
1972	8.8	16.8	24.4	30.8	51.4	71.6
1973	18.0	32.8	33.0	34.8	42.0	56.0
1974	12.8	17.0	25.6	44.2	50.0	NA
1975	10.0	14.0	18.2	27.0	30.0	51.0
1976	NA	21.0	44.2	74.0	112.0	147.6
1977	10.8	13.8	20.2	33.6	46.4	54.0
1978	18.8	21.6	22.6	26.0	40.0	66.0
1979	7.2	21.0	28.0	39.0	73.8	102.0
1980	16.4	26.0	41.4	57.8	86.8	106.8
1981	10.2	13.4	23.2	28.8	47.4	91.0
1982	13.6	26.2	27.4	36.0	51.0	60.4
1983	12.6	24.2	38.6	55.8	70.2	71.2
1984	5.2	13.8	18.6	37.6	39.2	41.4
1985	13.8	37.4	52.8	52.8	62.4	67.8
1986	7.0	12.8	22.8	44.0	80.2	120.8
1987	15.6	19.8	27.2	39.4	63.2	68.6
1988	7.0	14.4	22.4	38.0	48.6	56.8
1989	16.4	34.4	35.4	36.0	44.0	57.6
1990	11.6	18.4	46.2	75.2	117.2	121.2

## 2. Analisi dei dati orari

### 2.1 Precipitazioni massime annuali

Vengono importati i dati contenuti nel file .txt all'interno della HomeFolder e creata una entità che li contenga nella workspace:

```
list.files()
read.table("massimiprova.txt", header = TRUE) -> data
data
```

Sono inoltre inizializzate due variabili che contengono i dati in forma vettoriale: la variabile hh rappresenta i mm di pioggia, mentre la variabile h rappresenta le durate rispetto alle quali si riferiscono i dati di hh.

```
rep(c(0.25,1,3,6,12,24), each=dim(data)[[1]]) -> h
h
dim(data)[[1]]
c(data[[2]],data[[3]],data[[4]],data[[5]],data[[6]],data[[7]]) -> hh
hh
```

Si riportano i plot dei massimi di precipitazione in funzione della durata e il relativo diagramma a scatole:

```
plot(hh~h,xlab="Durata[h]",ylab="Precipitazione[mm]",
      main="Precipitazioni Massime a Trento Laste")
boxplot(hh~h,xlab="Durata[h]",ylab="Precipitazione[mm]",
        main="Precipitazioni Massime a Trento Laste")
```

Il boxplot è una tipologia grafica che permette di rappresentare dati mediante scatole, nelle quali si distinguono:

- la mediana (0.5 quantile), individuata mediante la linea in grassetto;
- la scatola, delimitata dallo 0.25 e 0.75 quantile;
- i baffi, ovvero segmenti oltre i quali si trovano solamente dati eccezionali, rispetto alle ipotesi fatte sul tipo di distribuzione.

### Precipitazioni Massime a Trento Laste

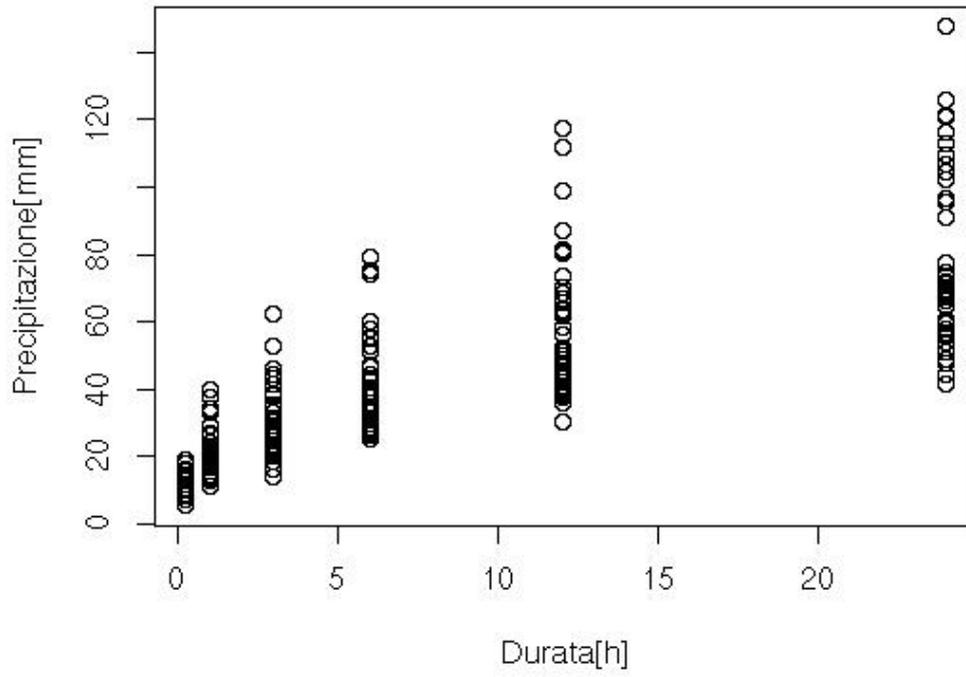


Figura 2: Plot

### Precipitazioni Massime a Trento Laste

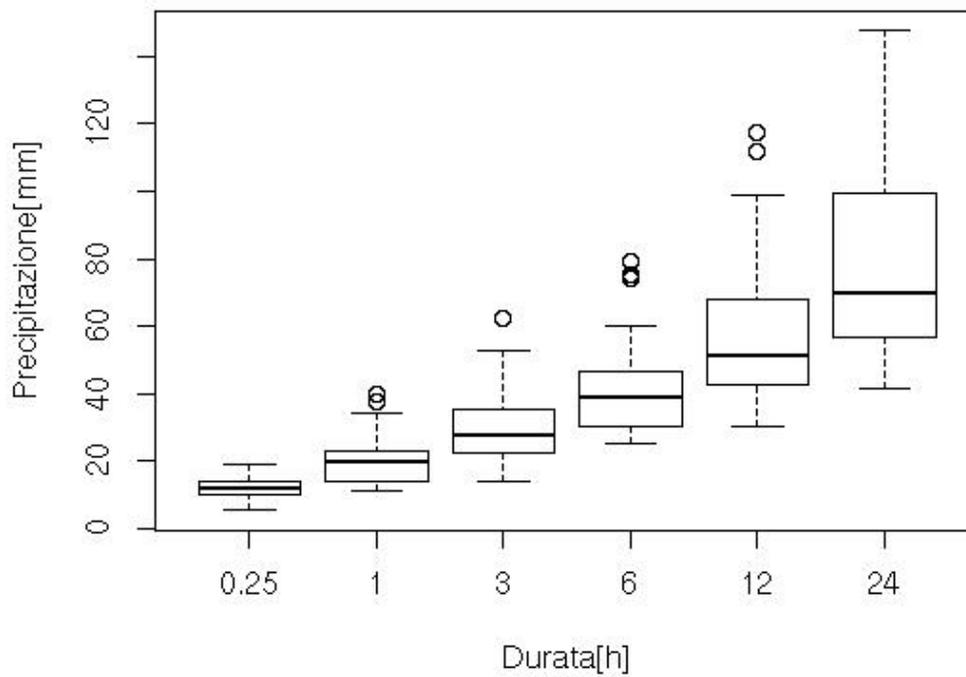


Figura 3: Boxplot

Di seguito si riporta il codice con i relativi istogrammi ottenuti per ogni intervallo di tempo considerato:

```
hist(data[[2]], breaks=8, xlab="Precipitazioni [mm]", ylab="Frequenza",  
      main= "Precipitazioni Massime a Trento Laste (15min)")
```

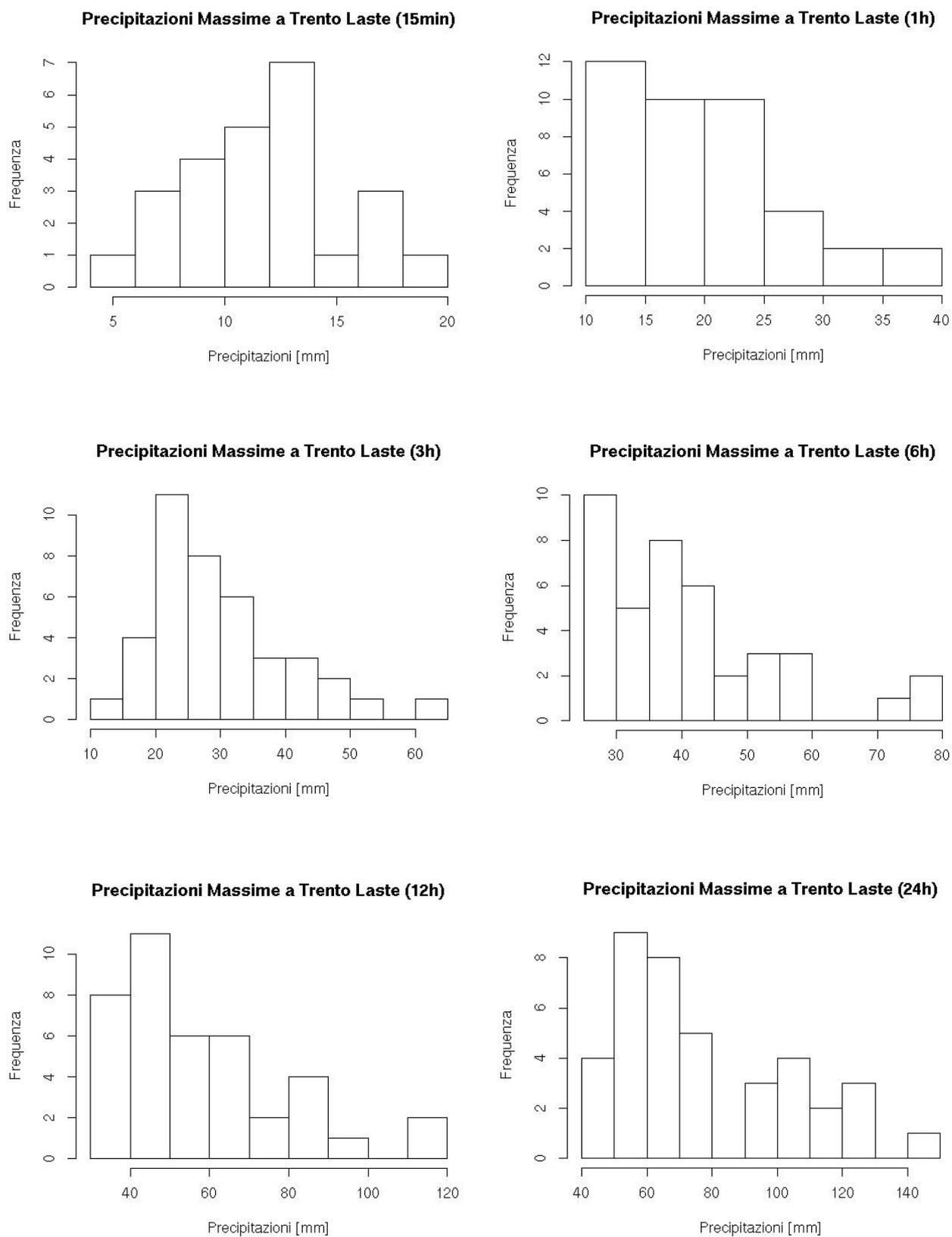
```
hist(data[[3]], breaks=8, xlab="Precipitazioni [mm]", ylab="Frequenza",  
      main= "Precipitazioni Massime a Trento Laste (1h)")
```

```
hist(data[[4]], breaks=8, xlab="Precipitazioni [mm]", ylab="Frequenza",  
      main= "Precipitazioni Massime a Trento Laste (3h)")
```

```
hist(data[[5]], breaks=8, xlab="Precipitazioni [mm]", ylab="Frequenza",  
      main= "Precipitazioni Massime a Trento Laste (6h)")
```

```
hist(data[[6]], breaks=8, xlab="Precipitazioni [mm]", ylab="Frequenza",  
      main= "Precipitazioni Massime a Trento Laste (12h)")
```

```
hist(data[[7]], breaks=8, xlab="Precipitazioni [mm]", ylab="Frequenza",  
      main= "Precipitazioni Massime a Trento Laste (24h)")
```



**Figura 4:** Istogrammi relativi ai massimi di precipitazione per le varie durate

Per ogni durata è stata eseguita una procedura al fine di interpolare i dati ad una distribuzione di probabilità. La famiglia di curve candidata per questo scopo è la curva dei valori estremi di tipo I, o curva di Gumbel, attraverso metodi di adattamento dei parametri, quali il metodo dei momenti, il metodo dei minimi quadrati e quello della massima verosimiglianza. Tramite il Test di Pearson si scelgono le coppie di valori (a, b) atte ad interpolare in maniera ottimale i dati.

## 2.2 Analisi per 15 minuti

Si creano due vettori contenenti la colonna anni e la colonna relativa alle precipitazioni di durata 15 minuti:

```
data15 <- data[[2]]
anni <- data[[1]]
```

I dati mancanti vengono eliminati per evitare problemi computazionali

```
anni <- anni[!is.na(data15)]
data15 <- data15[!is.na(data15)]
```

Si costruisce la frequenza empirica di non superamento (ecdf) con la relativa rappresentazione grafica

```
ecdf(data[[2]]) -> x
x
plot(x,xlab="h[mm]",ylab="P[H<h]",main="Frequenza di non superamento")
```

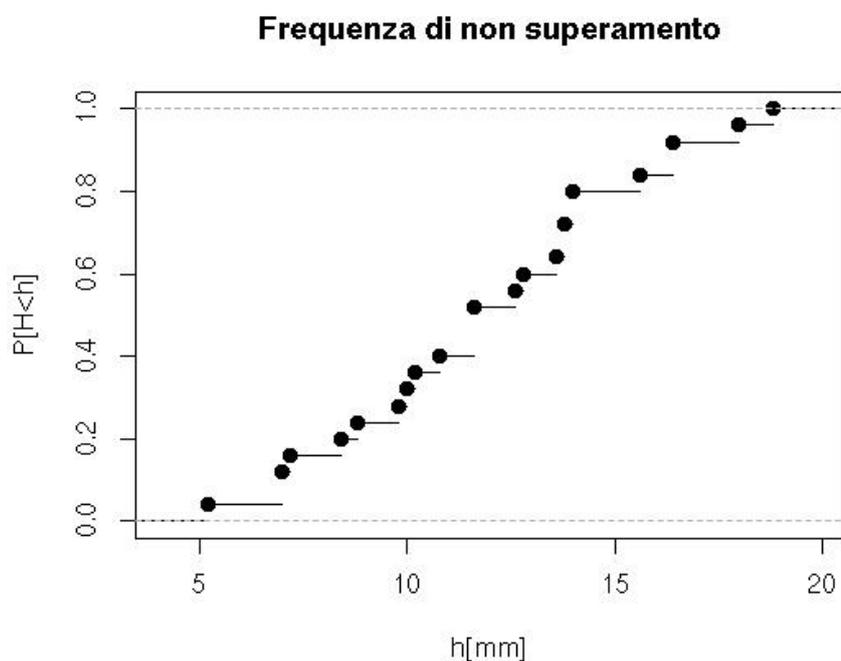


Figura 5: ecdf per 15 minuti

Si verifica che i dati orari abbiano una distribuzione normale. Essi vengono standardizzati, ovvero ridotti a media nulla e deviazione standard unitaria:

```
mean(data15)
var(data15)
sd(data15)
h.norm = (data15 - mean(data15))/sd(data15)
qqnorm(h.norm)
abline(0,1)
mean(data15) -> m15
var(data15) -> v15
```

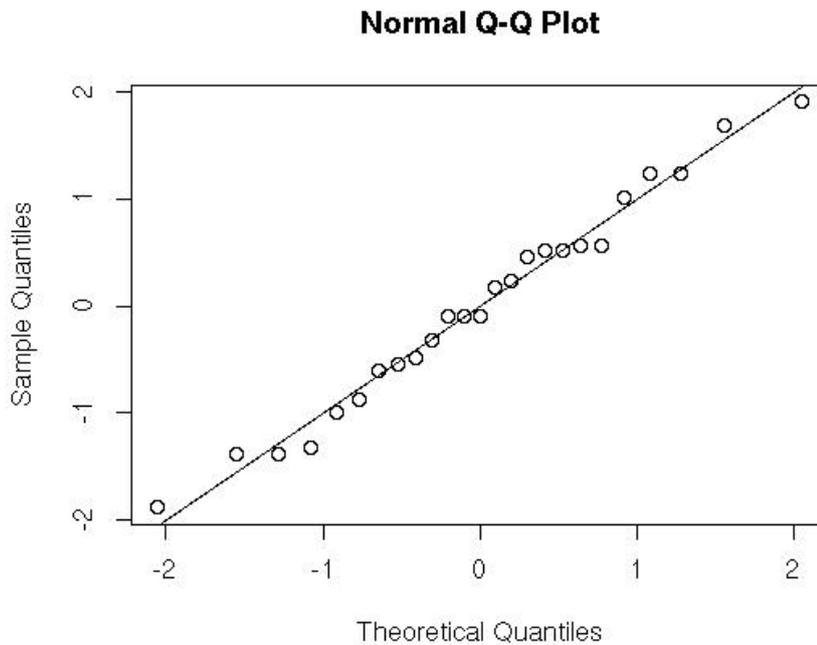


Figura 6: Q-Q Plot per 15 minuti

### 2.2.1 Metodo dei momenti

Il metodo stima i parametri della curva di Gumbel attraverso un'uguaglianza tra momenti della popolazione e momenti del campione empirico. Per disegnare le curve di Gumbel viene caricato il pacchetto evd.

```
load(evd)
pi
b1.gumbel = sqrt(6*v15)/pi
b1.gumbel
eulergamma = 0.577216
a1.gumbel = (m15 - b1.gumbel * eulergamma)
a1.gumbel
xx <- sort(data15)
xx
```

```
plot(xx,pgumbel(xx,loc=a1.gumbel,scale=b1.gumbel),xlab="h[mm]",
      ylab="P[H<h]",main="Frequenza di non superamento e curva di Gumbel
      con parametri ottimali",col="blue",type="l")
plot(ecdf(data15),xlab="h[mm]",main="Frequenza di non superamento",add=T)
```

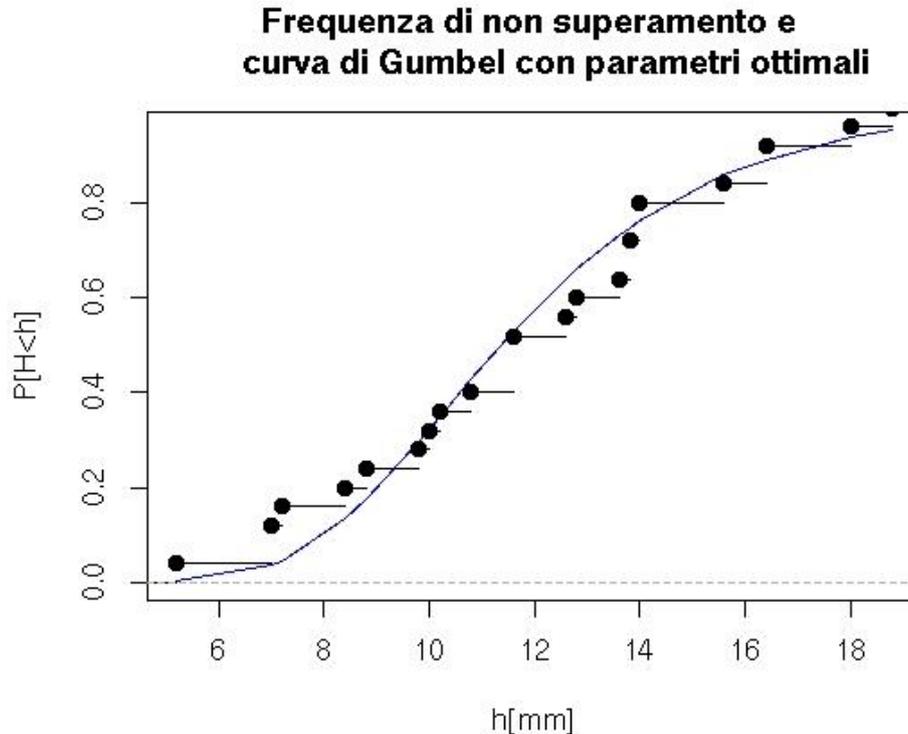


Figura 7: ecdf e curva di Gumbel per 15 minuti con parametri ottimali

### 2.2.2 Metodo dei minimi quadrati

Il metodo consiste nel valutare la frequenza empirica di superamento associata a ciascuno dei dati.

```
ec = ecdf(data15)
length(data15)
Fi = ec(sort(data15))
Fi
Y = -(log(-log(Fi[-25])))
Y
```

L'ultimo dato è infinito quindi viene eliminato:

```
X = sort(data15)[-25]
X
lsfit(X,Y) -> fts
fts$coefficients
b2.gumbel = fts$coefficients[[2]]^-1
b2.gumbel
a2.gumbel = -fts$coefficients[[1]]*b2.gumbel
a2.gumbel
b1.gumbel
a1.gumbel
```

### 2.2.3 Metodo della massima verosimiglianza

Per eseguire il metodo della massima verosimiglianza si carica il pacchetto MASS, che contiene la funzione fitdistr().

```
load(MASS)
fitdistr(data15, densfun = dgumbel, start=list(loc=a1.gumbel,
      scale=b1.gumbel)) -> mlab
mlab
a3.gumbel <- 10.2120423
b3.gumbel <- 3.2731753
```

### 2.2.4 Test di Pearson

Il Test di Pearson enuclea il metodo che ha stimato i parametri nella maniera migliore. Ciò è possibile attraverso il calcolo del  $\chi^2$  per ogni metodo e la scelta di quello che restituisce il valore minore.

```
q = c(0.2, 0.4, 0.6, 0.8)
q
vect = c(a1.gumbel, b1.gumbel, a2.gumbel, b2.gumbel, a3.gumbel, b3.gumbel)
vect
```

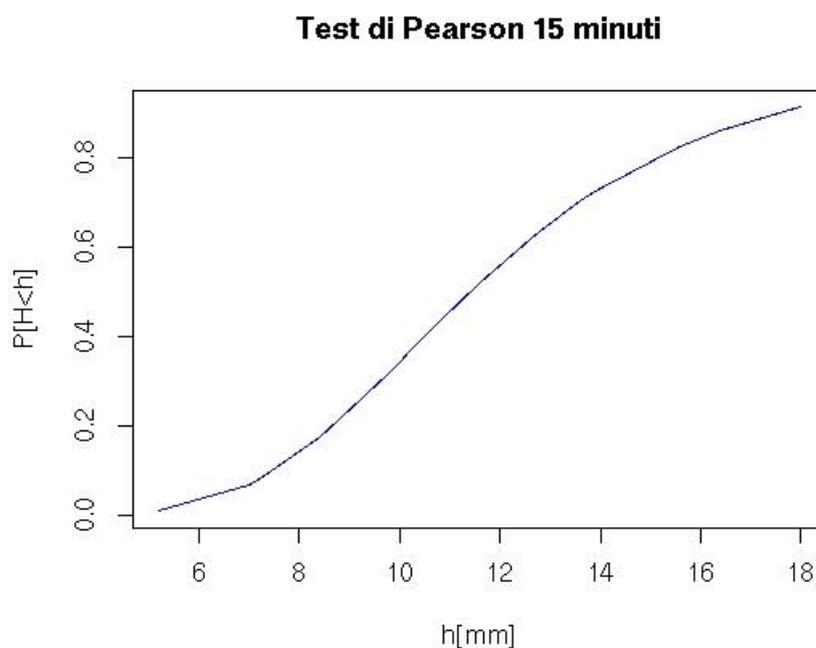
```
#a1 b1 (metodo dei momenti)
qgumbel (q, loc=a1.gumbel, scale=b1.gumbel) -> qi
qi
ec(qi)*25
c(0, ec(qi)*25) -> no1
no1
c(ec(qi)*25, 25) -> no2
no2
no2 - no1 -> no
no
0.2*length(data15) -> deltapi
deltapi
X1 = sum((no - deltapi)^2/deltapi)
X1
```

```
#a2 b2 (metodo dei minimi quadrati)
qgumbel (q, loc=a2.gumbel, scale=b2.gumbel) -> qi
qi
ec(qi)*25
c(0, ec(qi)*25) -> no1
no1
c(ec(qi)*25, 25) -> no2
no2
no2 - no1 -> no
no
0.2*length(data15) -> deltapi
deltapi
X2 = sum((no - deltapi)^2/deltapi)
X2
```

```
#a3 b3 (metodo della massima verosimiglianza)
qgumbel (q, loc=a3.gumbel, scale=b3.gumbel) -> qi
qi
ec(qi)*25
c(0, ec(qi)*25) -> no1
no1
c(ec(qi)*25, 25) -> no2
no2
no2 - no1 -> no
no
0.2*length(data15) -> deltapi
deltapi
X3 = sum((no - deltapi)^2/deltapi)
X3
```

	M. momenti	M. minimi quadrati	M. massima verosimiglianza
a.gumbel	10.3467	9.900547	10.2120423
b.gumbel	2.794971	3.036988	3.2731753
T.Pearson	X1 = 2	X2 = 1.6	X3 = 1.2

```
#plot della soluzione migliore 15min (metodo max verosimiglianza)
plot(X,pgumbel(X,loc=a3.gumbel,scale=b3.gumbel),xlab="h[mm]",
      ylab="P[H<h]",col="blue",type="l",main="Test di Pearson 15 minuti")
```



**Figura 8:** Test di Pearson per 15 minuti con il metodo della massima verosimiglianza

Si ripetono le operazioni precedenti per tutti gli intervalli di tempo al fine di determinare i parametri delle curve di Gumbel relativi ai vari intervalli.

### 2.3 Analisi per 1 ora

Per questa analisi oraria non si hanno valori mancanti da eliminare.

```
ecdf(data[[3]]) -> y
y
plot(y,xlab="h[mm] ",ylab="P[h<h] ",main="Frequenza di non superamento")
```

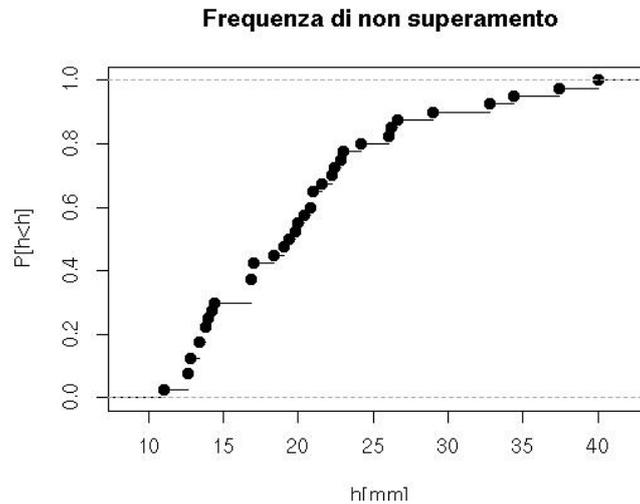


Figura 9: ecdf per 1 ora

```
mean(data[[3]])
var(data[[3]])
sd(data[[3]])
h.norm = (data[[3]] - mean(data[[3]]))/sd(data[[3]])
qqnorm(h.norm)
abline(0,1)
mean(data[[3]]) -> mlh
var(data[[3]]) -> vlh
```

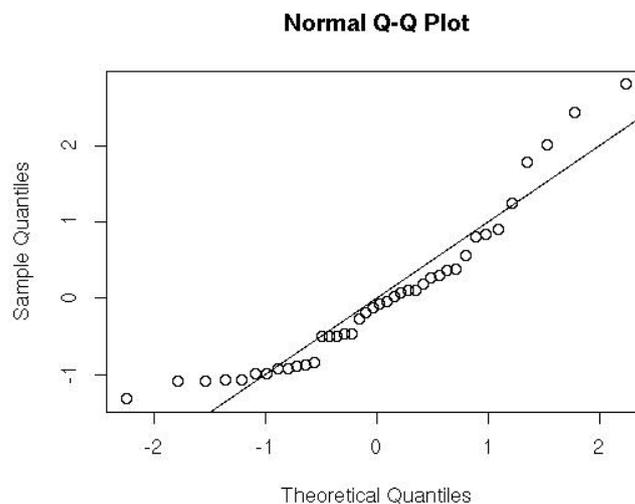


Figura 10: Q-Q Plot per 1 ora

### 2.3.1 Metodo dei momenti

```

pi
b1.gumbel = sqrt(6*v1h)/pi
b1.gumbel
eulergamma = 0.577216
a1.gumbel = (m1h - b1.gumbel * eulergamma)
a1.gumbel
yy <- sort(data[[3]])
YY
plot(yy,pgumbel(yy,loc=a1.gumbel,scale=b1.gumbel),xlab="h[mm]",
      ylab="P[H<h]",main="Frequenza di non superamento e
      curva di Gumbel con parametri ottimali",col="red",type="l")
plot(ecdf(data[[3]]),xlab="h[mm]",main="Frequenza di non superamento",
      add=T)

```

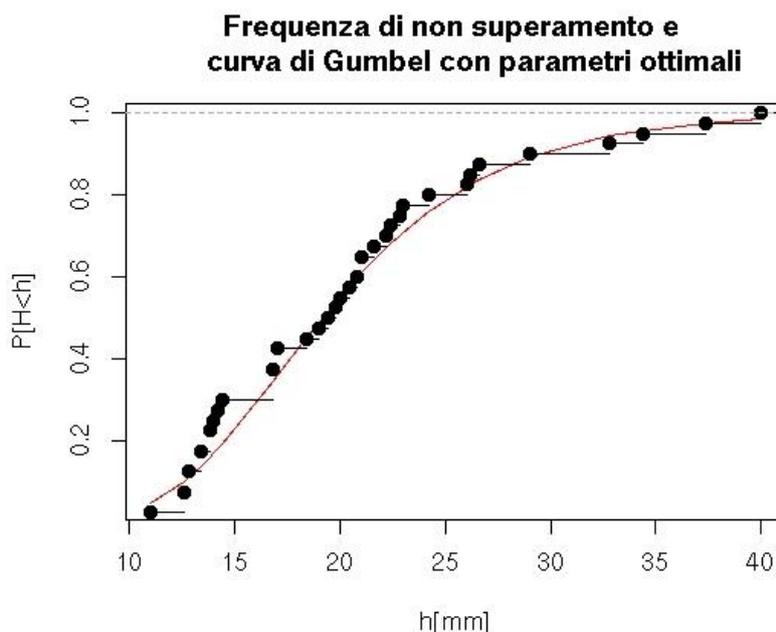


Figura 11: ecdf e curva di Gumbel per 15 minuti con parametri ottimali

### 2.3.2 Metodo dei minimi quadrati

```

ec = ecdf(data[[3]])
length(data[[3]])
Fi = ec(sort(data[[3]]))
Fi
Y = -(log(-log(Fi[-40])))
Y
X = sort(data[[3]])[-40]
X
lsfit(X,Y) -> fts
fts$coefficients
b2.gumbel = fts$coefficients[[2]]^-1
b2.gumbel
a2.gumbel = -fts$coefficients[[1]]*b2.gumbel
a2.gumbel
b1.gumbel
a1.gumbel

```

### 2.3.3 Metodo della massima verosimiglianza

```
load(MASS)
fitdistr(data[[3]],densfun=dgumbel,start=list(loc=a1.gumbel,
      scale=b1.gumbel)) -> mlab
mlab
a3.gumbel <- 17.1956785
b3.gumbel <- 5.0761437
```

### 2.3.4 Test di Pearson

```
q = c(0.2, 0.4, 0.6, 0.8)
q
vect=c(a1.gumbel,b1.gumbel,a2.gumbel,b2.gumbel,a3.gumbel,b3.gumbel)
vect
```

```
#a1 b1 (metodo dei momenti)
qgumbel (q, loc=a1.gumbel, scale=b1.gumbel) -> qi
qi
ec(qi)*40
c(0, ec(qi)*40) -> no1
no1
c(ec(qi)*40, 40) -> no2
no2
no2 - no1 -> no
no
0.2*length(data[[3]]) -> deltapi
deltapi
X1 = sum((no - deltapi)^2/deltapi)
X1
```

```
#a2 b2 (metodo dei minimi quadrati)
qgumbel (q, loc=a2.gumbel, scale=b2.gumbel) -> qi
qi
ec(qi)*40
c(0, ec(qi)*40) -> no1
no1
c(ec(qi)*40, 40) -> no2
no2
no2 - no1 -> no
no
0.2*length(data[[3]]) -> deltapi
deltapi
X2 = sum((no - deltapi)^2/deltapi)
X2
```

```
#a3 b3 (metodo della massima verosimiglianza)
qgumbel (q, loc=a3.gumbel, scale=b3.gumbel) -> qi
qi
ec(qi)*40
c(0, ec(qi)*40) -> no1
no1
c(ec(qi)*40, 40) -> no2
no2
```

```
no2 - no1 -> no
no
0.2*length(data[[3]]) -> deltapi
deltapi
X3 = sum((no - deltapi)^2/deltapi)
X3
```

	M. momenti	M. minimi quadrati	M.massima verosimiglianza
a.gumbel	17.12243	16.62057	17.1956785
b.gumbel	5.487665	5.622907	5.0761437
T.Person	X1 = 3.25	X2 = 1.75	X3 = 3.75

```
#plot della soluzione migliore 1h (metodo minimi quadrati)
plot(X,pgumbel(X,loc=a2.gumbel,scale=b2.gumbel),xlab="h[mm]",
      ylab="P[H<h]",col="red",type="l",main="Test di Pearson 1 ora")
```

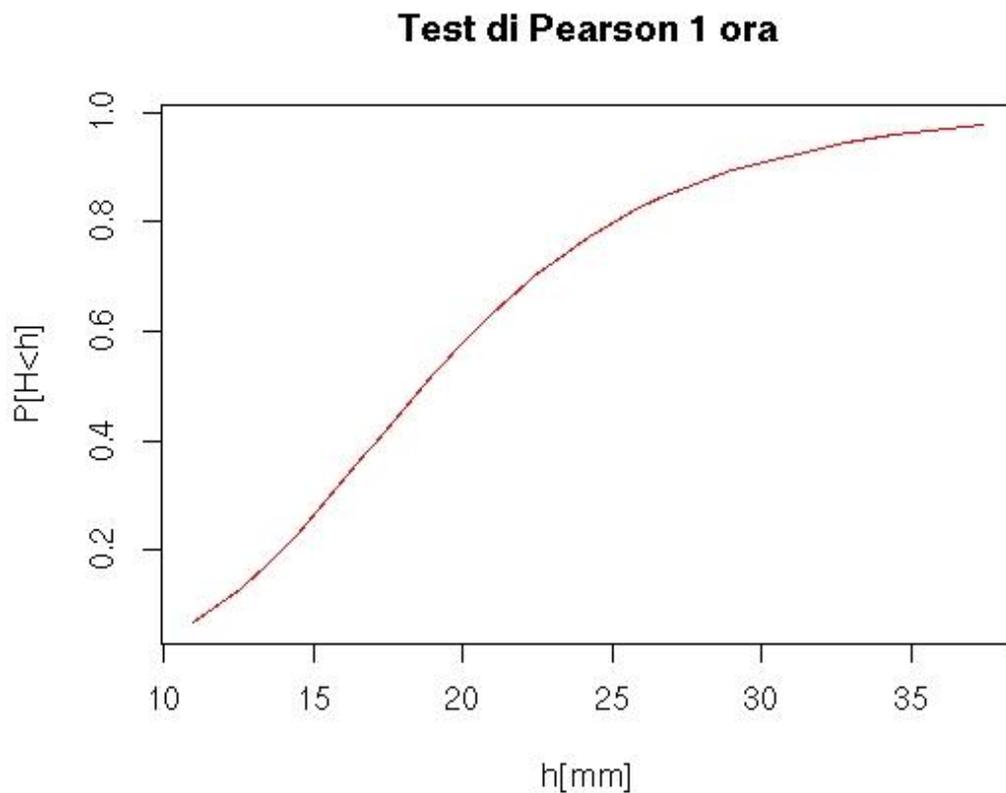


Figura 12: Test di Pearson per 1 ora con il metodo dei minimi quadrati

## 2.4 Analisi per 3 ore

```
ecdf(data[[4]]) -> z
```

```
z
```

```
plot(z,xlab="h[mm]",ylab="P[h<h]",main="Frequenza di non superamento")
```

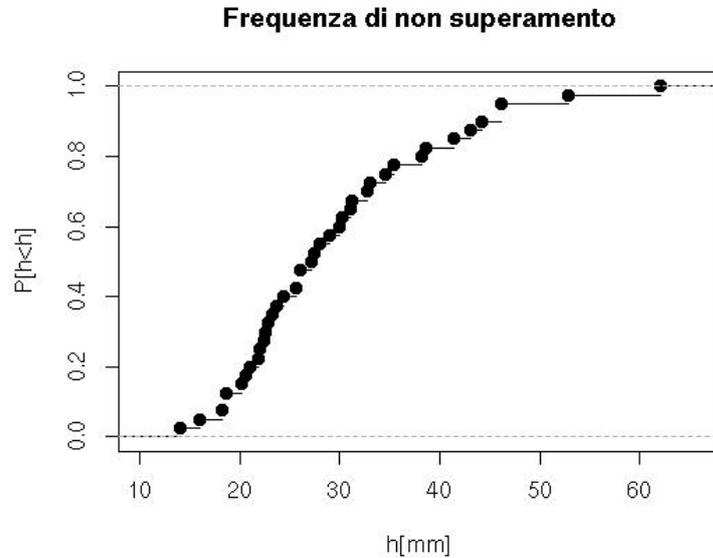


Figura 13: ecdf per 3 ore

```
mean(data[[4]])
```

```
var(data[[4]])
```

```
sd(data[[4]])
```

```
h.norm = (data[[4]] - mean(data[[4]]))/sd(data[[4]])
```

```
qqnorm(h.norm)
```

```
abline(0,1)
```

```
mean(data[[4]]) -> m3h
```

```
var(data[[4]]) -> v3h
```

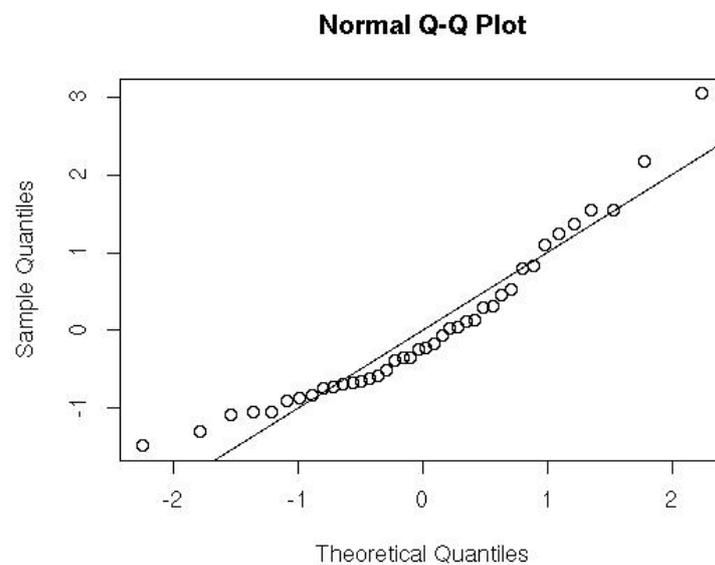


Figura 14: Q-Q Plot per 3 ore

### 2.4.1 Metodo dei momenti

```

pi
b1.gumbel = sqrt(6*v3h)/pi
b1.gumbel
eulergamma = 0.577216
a1.gumbel = (m3h - b1.gumbel * eulergamma)
a1.gumbel
zz <- sort(data[[4]])
zz
plot(zz, pgumbel(zz,loc=a1.gumbel,scale=b1.gumbel),xlab="h[mm]",
      ylab="P[H<h]",main="Frequenza di non superamento e
      curva di Gumbel con parametri ottimali", col="green", type="l")
plot(ecdf(data[[4]]), xlab="h[mm]", main="Frequenza di non superamento",
      add=T)

```

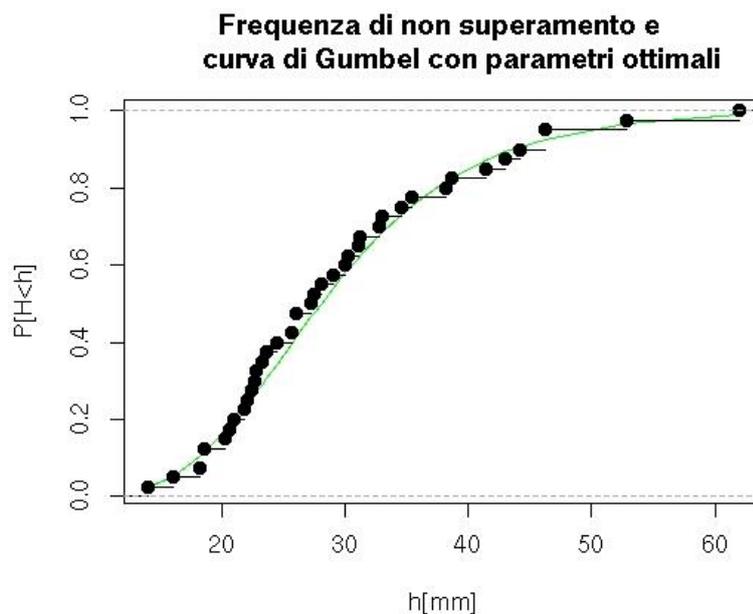


Figura 15: ecdf e curva di Gumbel per 3 ore con parametri ottimali

### 2.4.2 Metodo dei minimi quadrati

```

ec = ecdf(data[[4]])
length(data[[4]])
Fi = ec(sort(data[[4]]))
Fi
Y = -(log(-log(Fi[-40])))
Y
X = sort(data[[4]])[-40]
X
lsfit(X,Y) -> fts
fts$coefficients
b2.gumbel = fts$coefficients[[2]]^-1
b2.gumbel
a2.gumbel = -fts$coefficients[[1]]*b2.gumbel
a2.gumbel
b1.gumbel
a1.gumbel

```

### 2.4.3 Metodo della massima verosimiglianza

```
load(MASS)
fitdistr(data[[4]], densfun = dgumbel, start=list(loc=a1.gumbel,
        scale=b1.gumbel)) -> mlab
mlab
a3.gumbel <- 25.0690551
b3.gumbel <- 7.7768589
```

### 2.4.4 Test di Pearson

```
q = c(0.2, 0.4, 0.6, 0.8)
q
vect = c(a1.gumbel,b1.gumbel,a2.gumbel,b2.gumbel,a3.gumbel,b3.gumbel)
vect
```

```
#a1 b1 (metodo dei momenti)
qgumbel (q, loc=a1.gumbel, scale=b1.gumbel) -> qi
qi
ec(qi)*40
c(0, ec(qi)*40) -> no1
no1
c(ec(qi)*40, 40) -> no2
no2
no2 - no1 -> no
no
0.2*length(data[[4]]) -> deltapi
deltapi
X1 = sum((no - deltapi)^2/deltapi)
X1
```

```
#a2 b2 (metodo dei minimi quadrati)
qgumbel (q, loc=a2.gumbel, scale=b2.gumbel) -> qi
qi
ec(qi)*40
c(0, ec(qi)*40) -> no1
no1
c(ec(qi)*40, 40) -> no2
no2
no2 - no1 -> no
no
0.2*length(data[[4]]) -> deltapi
deltapi
X2 = sum((no - deltapi)^2/deltapi)
X2
```

```
#a3 b3 (metodo della massima verosimiglianza)
qgumbel (q, loc=a3.gumbel, scale=b3.gumbel) -> qi
qi
ec(qi)*40
c(0, ec(qi)*40) -> no1
no1
c(ec(qi)*40, 40) -> no2
no2
no2 - no1 -> no
```

```
no
0.2*length(data[[4]]) -> deltapi
deltapi
X3 = sum((no - deltapi)^2/deltapi)
X3
```

	M. momenti	M. minimi quadrati	M. massima verosimiglianza
a.gumbel	24.9809	24.44839	25.0690551
b.gumbel	8.26224	8.016804	7.7768589
T. Pearson	X1 = 0.75	X2 = 0.5	X3 = 0.75

```
#plot della soluzione migliore 3h (metodo minimi quadrati)
plot(X,pgumbel(X,loc=a2.gumbel,scale=b2.gumbel),xlab="h[mm]",
      ylab="P[H<h]",col="green",type="l",main="Test di Pearson 3 ore")
```

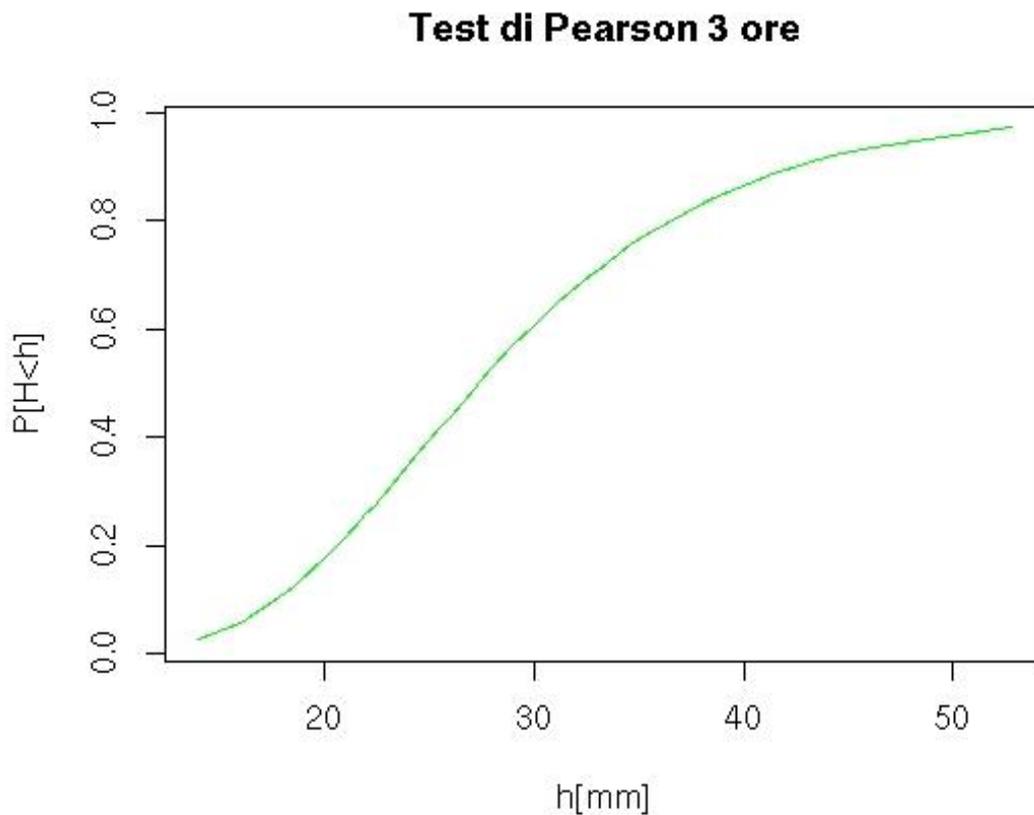


Figura 16: Test di Pearson per 3 ore con il metodo dei minimi quadrati

### 2.5 Analisi per 6 ore

```
ecdf(data[[5]]) -> u
```

```
u
```

```
plot(u,xlab="h[mm]",ylab="P[h<h]",main="Frequenza di non superamento")
```

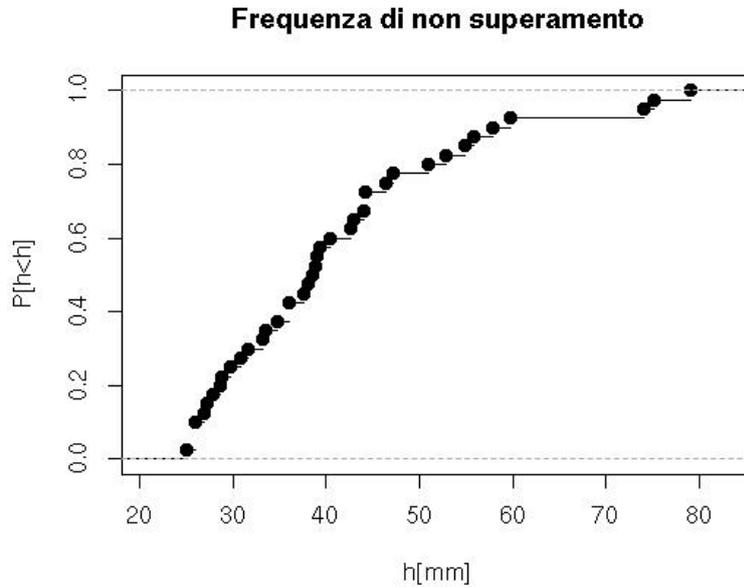


Figura 17: ecdf per 6 ore

```
mean(data[[5]])
```

```
var(data[[5]])
```

```
sd(data[[5]])
```

```
h.norm = (data[[5]] - mean(data[[5]])/sd(data[[5]])
```

```
qqnorm(h.norm)
```

```
abline(0,1)
```

```
mean(data[[5]]) -> m6h
```

```
var(data[[5]]) -> v6h
```

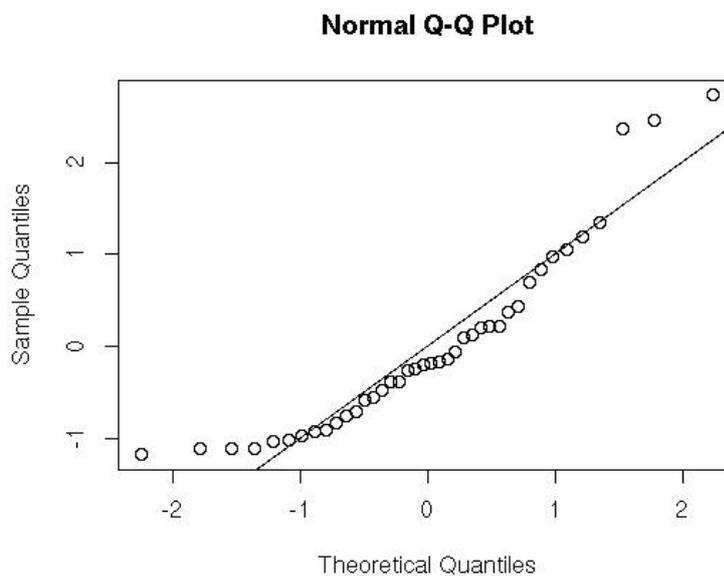


Figura 18: Q-Q Plot per 6 ore

### 2.5.1 Metodo dei momenti

```

pi
b1.gumbel = sqrt(6*v6h)/pi
b1.gumbel
eulergamma = 0.577216
a1.gumbel = (m6h - b1.gumbel * eulergamma)
a1.gumbel
uu <- sort(data[[5]])
uu
plot(uu,pgumbel(uu,loc=a1.gumbel,scale=b1.gumbel),xlab="h[mm]",
      ylab="P[H<h]",main="Frequenza di non superamento e
      curva di Gumbel con parametri ottimali",col="purple",type="l")
plot(ecdf(data[[5]]),xlab="h[mm]",main="Frequenza di non superamento",
      add=T)
    
```

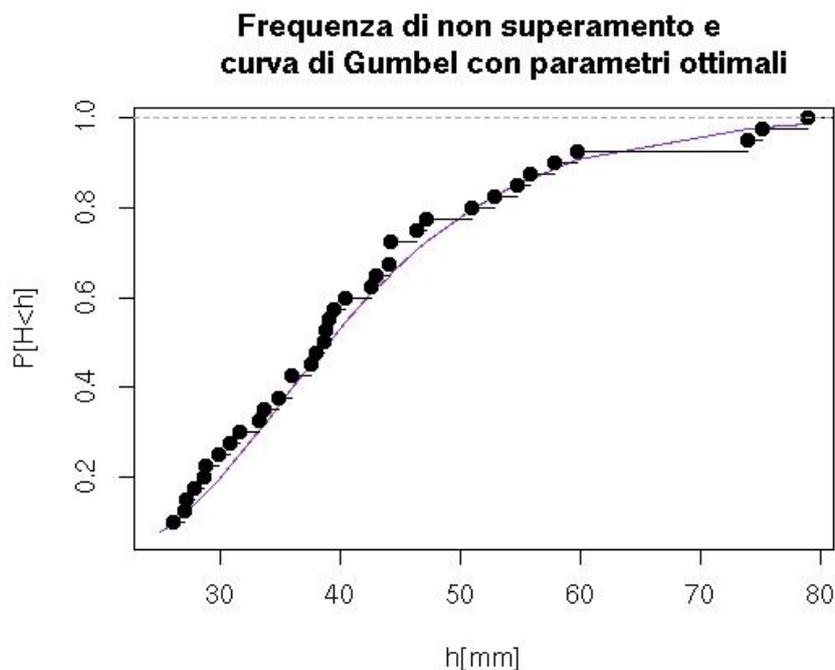


Figura 19: ecdf e curva di Gumbel per 6 ore con parametri ottimali

### 2.5.2 Metodo dei minimi quadrati

```

ec = ecdf(data[[5]])
length(data[[5]])
Fi = ec(sort(data[[5]]))
Fi
Y = -(log(-log(Fi[-40])))
Y
X = sort(data[[5]])[-40]
X
lsfit(X,Y) -> fts
fts$coefficients
b2.gumbel = fts$coefficients[[2]]^-1
b2.gumbel
a2.gumbel = -fts$coefficients[[1]]*b2.gumbel
a2.gumbel
b1.gumbel
a1.gumbel
    
```

### 2.5.3 Metodo della massima verosimiglianza

```
load(MASS)
fitdistr(data[[5]], densfun = dgumbel, start=list(loc=a1.gumbel,
        scale=b1.gumbel)) -> mlab
mlab
a3.gumbel <- 35.283372
b3.gumbel <- 9.733107
```

### 2.5.4 Test di Pearson

```
q = c(0.2, 0.4, 0.6, 0.8)
q
vect = c(a1.gumbel,b1.gumbel,a2.gumbel,b2.gumbel,a3.gumbel,b3.gumbel)
vect
```

```
#a1 b1 (metodo dei momenti)
qgumbel (q, loc=a1.gumbel, scale=b1.gumbel) -> qi
qi
ec(qi)*40
c(0, ec(qi)*40) -> no1
no1
c(ec(qi)*40, 40) -> no2
no2
no2 - no1 -> no
no
0.2*length(data[[5]]) -> deltapi
deltapi
X1 = sum((no - deltapi)^2/deltapi)
X1
```

```
#a2 b2 (metodo dei minimi quadrati)
qgumbel (q, loc=a2.gumbel, scale=b2.gumbel) -> qi
qi
ec(qi)*40
c(0, ec(qi)*40) -> no1
no1
c(ec(qi)*40, 40) -> no2
no2
no2 - no1 -> no
no
0.2*length(data[[5]]) -> deltapi
deltapi
X2 = sum((no - deltapi)^2/deltapi)
X2
```

```
#a3 b3 (metodo della massima verosimiglianza)
qgumbel (q, loc=a3.gumbel, scale=b3.gumbel) -> qi
qi
ec(qi)*40
c(0, ec(qi)*40) -> no1
no1
c(ec(qi)*40, 40) -> no2
no2
no2 - no1 -> no
no
```

```
0.2*length(data[[5]]) -> deltapi
deltapi
X3 = sum((no - deltapi)^2/deltapi)
X3
```

	M. momenti	M. minimi quadrati	M. massima verosimiglianza
a.gumbel	35.07829	34.14221	35.283372
b.gumbel	10.77015	11.10204	9.733107
T.Pearson	X1 = 0.75	X2 = 1	X3 = 1

```
#plot della soluzione migliore 6h (metodo momenti)
plot(X,pgumbel(X,loc=a1.gumbel,scale=b1.gumbel),xlab="h[mm]",
      ylab="P[H<h]",col="purple",type="l",main="Test di Pearson 6 ore")
```

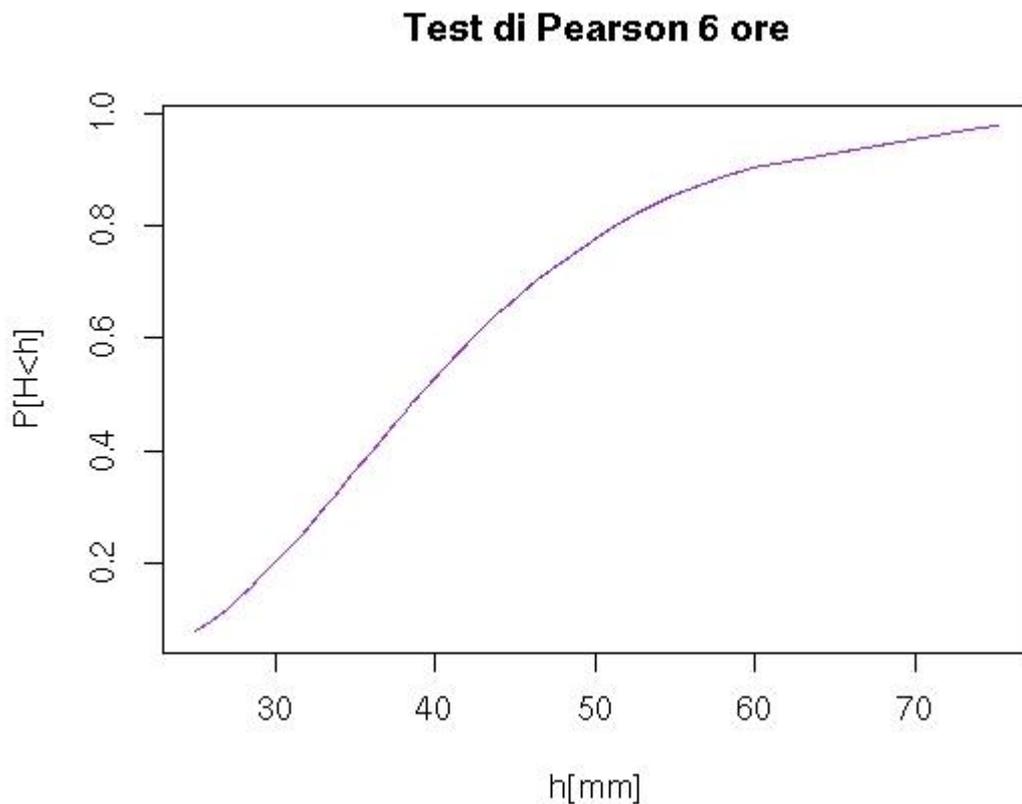


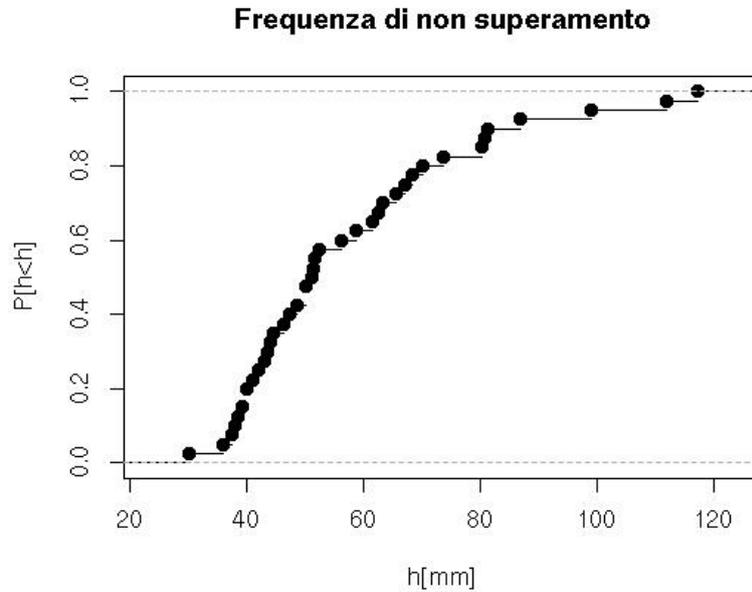
Figura 20: Test di Pearson per 6 ore con il metodo dei momenti

## 2.6 Analisi per 12 ore

```
ecdf(data[[6]]) -> v
```

```
v
```

```
plot(v,xlab="h[mm]",ylab="P[h<h]",main="Frequenza di non superamento")
```



**Figura 21:** ecdf per 12 ore

```
mean(data[[6]])
```

```
var(data[[6]])
```

```
sd(data[[6]])
```

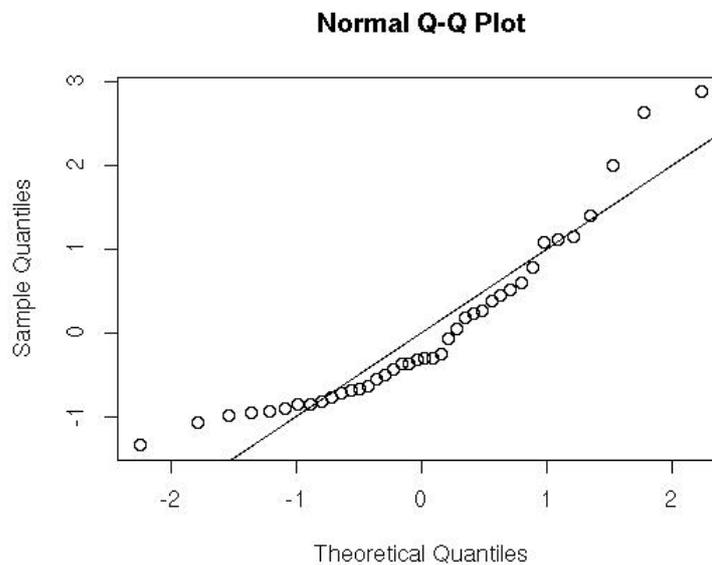
```
h.norm = (data[[6]] - mean(data[[6]]))/sd(data[[6]])
```

```
qqnorm(h.norm)
```

```
abline(0,1)
```

```
mean(data[[6]]) -> m12h
```

```
var(data[[6]]) -> v12h
```



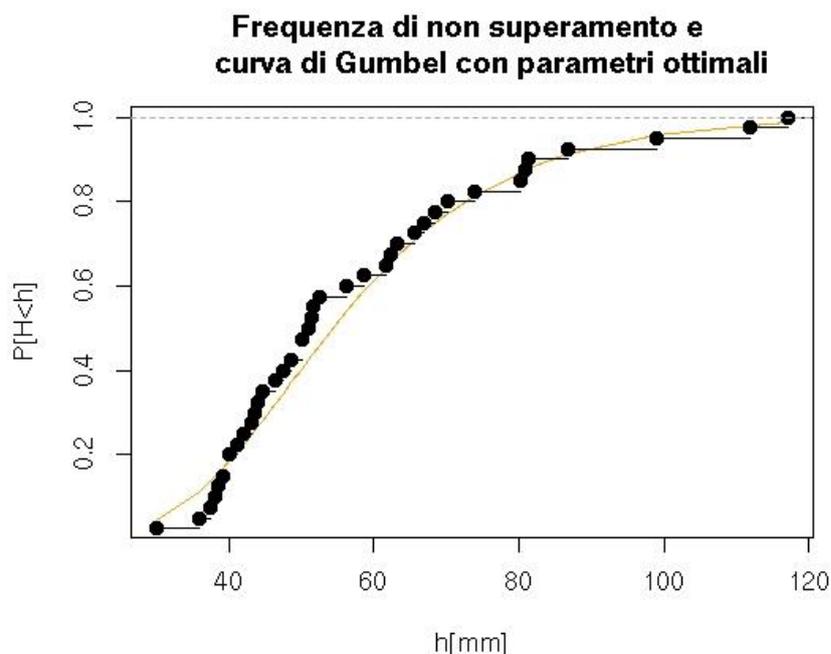
**Figura 22:** Q-Q Plot per 12 ore

### 2.6.1 Metodo dei momenti

```

pi
b1.gumbel = sqrt(6*v12h)/pi
b1.gumbel
eulergamma = 0.577216
a1.gumbel = (m12h - b1.gumbel * eulergamma)
a1.gumbel
vv <- sort(data[[6]])
vv
plot(vv,pgumbel(vv,loc=a1.gumbel,scale=b1.gumbel),xlab="h[mm]",
      ylab="P[H<h]",main="Frequenza di non superamento e
      curva di Gumbel con parametri ottimali",col="orange",type="l")
plot(ecdf(data[[6]]),xlab="h[mm]",main="Frequenza di non superamento",
      add=T)

```



**Figura 23:** ecdf e curva di Gumbel per 12 ore con parametri ottimali

### 2.6.2 Metodo dei minimi quadrati

```

ec = ecdf(data[[6]])
length(data[[6]])
Fi = ec(sort(data[[6]]))
Fi
Y = -(log(-log(Fi[-40])))
Y
X = sort(data[[6]])[-40]
X
lsfit(X,Y) -> fts
fts$coefficients
b2.gumbel = fts$coefficients[[2]]^-1
b2.gumbel
a2.gumbel = -fts$coefficients[[1]]*b2.gumbel
a2.gumbel
b1.gumbel
a1.gumbel

```

### 2.6.3 Metodo della massima verosimiglianza

```
load(MASS)
fitdistr(data[[6]],densfun=dgumbel,start=list(loc=a1.gumbel,
      scale=b1.gumbel)) -> mlab
mlab
a3.gumbel <- 48.871039
b3.gumbel <- 14.166813
```

### 2.6.4 Test di Pearson

```
q = c(0.2, 0.4, 0.6, 0.8)
q
vect = c(a1.gumbel,b1.gumbel,a2.gumbel,b2.gumbel,a3.gumbel,b3.gumbel)
vect
```

```
#a1 b1 (metodo dei momenti)
qgumbel (q, loc=a1.gumbel, scale=b1.gumbel) -> qi
qi
ec(qi)*40
c(0, ec(qi)*40) -> no1
no1
c(ec(qi)*40, 40) -> no2
no2
no2 - no1 -> no
no
0.2*length(data[[6]]) -> deltapi
deltapi
X1 = sum((no - deltapi)^2/deltapi)
X1
```

```
#a2 b2 (metodo dei minimi quadrati)
qgumbel (q, loc=a2.gumbel, scale=b2.gumbel) -> qi
qi
ec(qi)*40
c(0, ec(qi)*40) -> no1
no1
c(ec(qi)*40, 40) -> no2
no2
no2 - no1 -> no
no
0.2*length(data[[6]]) -> deltapi
deltapi
X2 = sum((no - deltapi)^2/deltapi)
X2
```

```
#a3 b3 (metodo della massima verosimiglianza)
qgumbel (q, loc=a3.gumbel, scale=b3.gumbel) -> qi
qi
ec(qi)*40
c(0, ec(qi)*40) -> no1
no1
c(ec(qi)*40, 40) -> no2
no2
no2 - no1 -> no
```

```
no
0.2*length(data[[6]]) -> deltapi
deltapi
X3 = sum((no - deltapi)^2/deltapi)
X3
```

	M. momenti	M. minimi quadrati	M. massima verosimiglianza
a.gumbel	48.45171	47.34606	48.871039
b.gumbel	16.11752	16.24681	14.166813
T. Pearson	X1 = 0.25	X2 = 1.75	X3 = 2

```
#plot della soluzione migliore 12h (metodo momenti)
plot(X,pgumbel(X,loc=a1.gumbel,scale=b1.gumbel),xlab="h[mm]",
      ylab="P[H<h]",col="orange",type="l",main="Test di Pearson 12 ore")
```

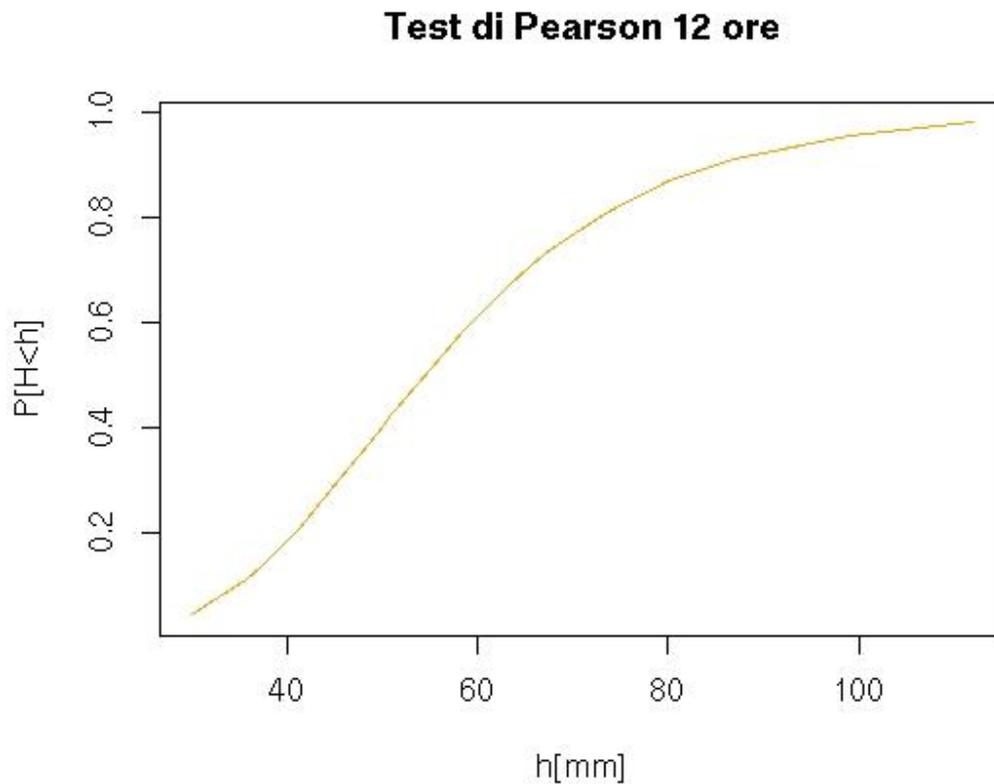


Figura 24: Test di Pearson per 12 ore con il metodo dei momenti

### 2.7 Analisi per 24 ore

```

data24 <- data[[7]]
anni <- data[[7]]
anni <- anni[!is.na(data24)]
data24 <- data24[!is.na(data24)]
ecdf(data[[7]]) -> w
w
plot(w,xlab="h[mm] ",ylab="P[H<h] ",main="Frequenza di non superamento")

```

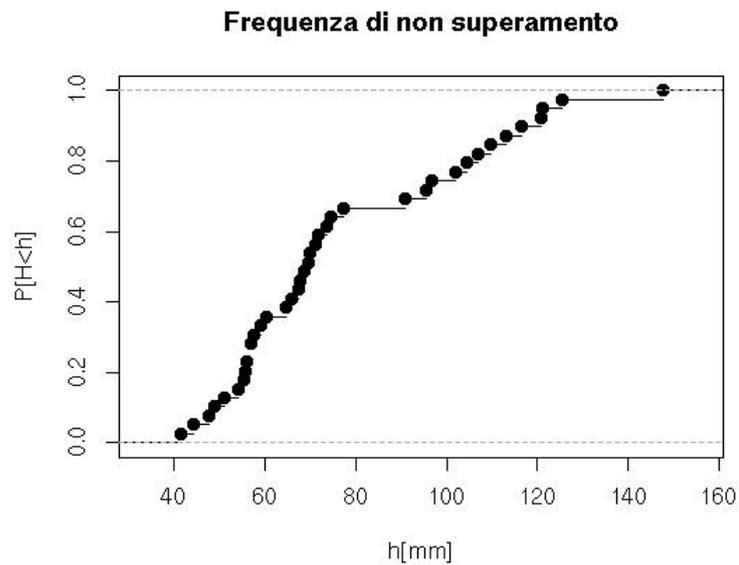


Figura 25: ecdf per 24 ore

```

mean(data24)
var(data24)
sd(data24)
h.norm = (data24 - mean(data24))/sd(data24)
qqnorm(h.norm)
abline(0,1)
mean(data24) -> m24
var(data24) -> v24

```

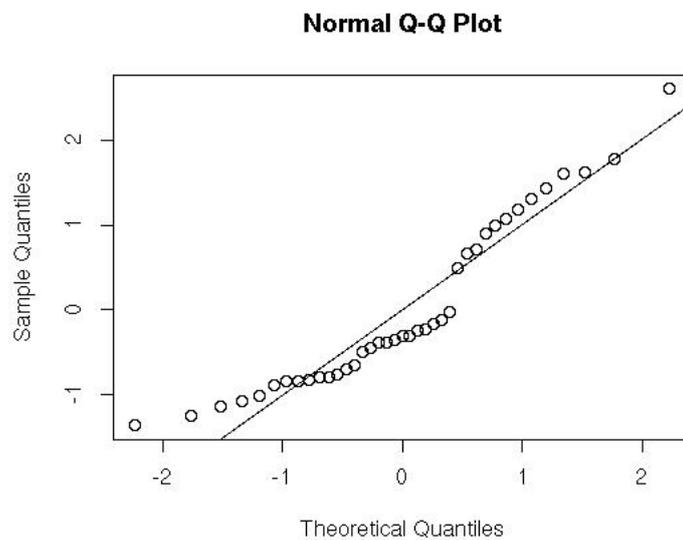


Figura 26: Q-Q Plot per 24 ore

### 2.7.1 Metodo dei momenti

```

pi
b1.gumbel = sqrt(6*v24)/pi
b1.gumbel
eulergamma = 0.577216
a1.gumbel = (m24 - b1.gumbel * eulergamma)
a1.gumbel
ww <- sort(data24)
ww
plot(ww,pgumbel(ww,loc=a1.gumbel,scale=b1.gumbel),xlab="h[mm]",
      ylab="P[H<h]",main="Frequenza di non superamento e curva di Gumbel
      con parametri ottimali",col="yellow",type="l")
plot(ecdf(data24),xlab="h[mm]",main="Frequenza di non superamento",add=T)

```

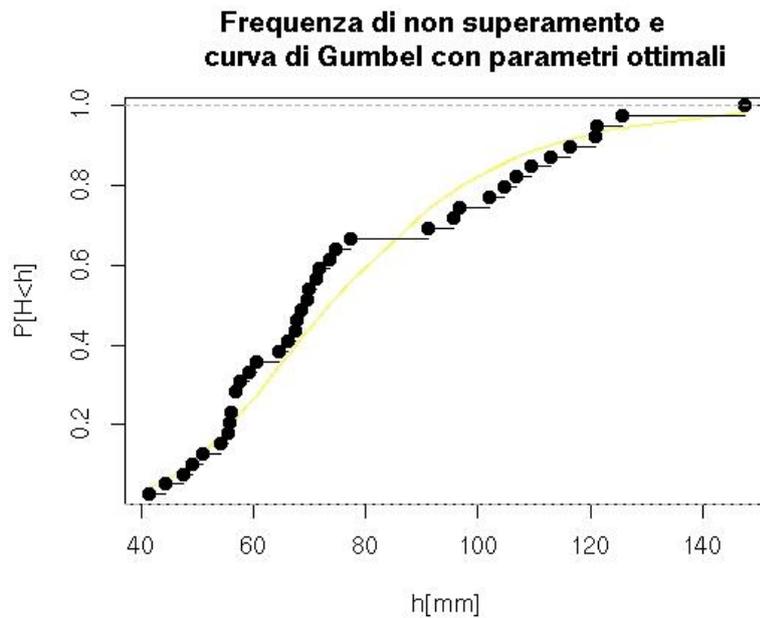


Figura 27: ecdf e curva di Gumbel per 24 ore con parametri ottimali

### 2.7.2 Metodo dei minimi quadrati

```

ec = ecdf(data24)
length(data24)
Fi = ec(sort(data24))
Fi
Y = -(log(-log(Fi[-39])))
Y
X = sort(data24)[-39]
X
lsfit(X,Y) -> fts
fts$coefficients
b2.gumbel = fts$coefficients[[2]]^-1
b2.gumbel
a2.gumbel = -fts$coefficients[[1]]*b2.gumbel
a2.gumbel
b1.gumbel
a1.gumbel

```

### 2.7.3 Metodo della massima verosimiglianza

```
load(MASS)
fitdistr(data24, densfun = dgumbel, start=list(loc=a1.gumbel,
      scale=b1.gumbel)) -> mlab
mlab
a3.gumbel <- 65.815532
b3.gumbel <- 19.711047
```

### 2.7.4 Test di Pearson

```
q = c(0.2, 0.4, 0.6, 0.8)
q
vect = c(a1.gumbel, b1.gumbel, a2.gumbel, b2.gumbel, a3.gumbel, b3.gumbel)
vect
```

```
#a1 b1 (metodo dei momenti)
qgumbel (q, loc=a1.gumbel, scale=b1.gumbel) -> qi
qi
ec(qi)*39
c(0, ec(qi)*39) -> no1
no1
c(ec(qi)*39, 39) -> no2
no2
no2 - no1 -> no
no
0.2*length(data24) -> deltapi
deltapi
X1 = sum((no - deltapi)^2/deltapi)
X1
```

```
#a2 b2 (metodo dei minimi quadrati)
qgumbel (q, loc=a2.gumbel, scale=b2.gumbel) -> qi
qi
ec(qi)*39
c(0, ec(qi)*39) -> no1
no1
c(ec(qi)*39, 39) -> no2
no2
no2 - no1 -> no
no
0.2*length(data24) -> deltapi
deltapi
X2 = sum((no - deltapi)^2/deltapi)
X2
```

```
#a3 b3 (metodo della massima verosimiglianza)
qgumbel (q, loc=a3.gumbel, scale=b3.gumbel) -> qi
qi
ec(qi)*39
c(0, ec(qi)*39) -> no1
no1
c(ec(qi)*39, 39) -> no2
no2
no2 - no1 -> no
```

```
no
0.2*length(data24) -> deltapi
deltapi
X3 = sum((no - deltapi)^2/deltapi)
X3
```

	M. momenti	M. minimi quadrati	M. massima verosimiglianza
a.gumbel	65.82873	64.13108	65.815532
b.gumbel	20.91736	21.93151	19.7111047
T.Pearson	X1 = 3.948718	X2 = 6.512821	X3 = 8.564103

```
#plot della soluzione migliore 24h (metodo momenti)
plot(X,pgumbel(X,loc=a1.gumbel,scale=b1.gumbel),xlab="h[mm]",
      ylab="P[H<h]",col="yellow",type="l",main="Test di Pearson 24 ore")
```

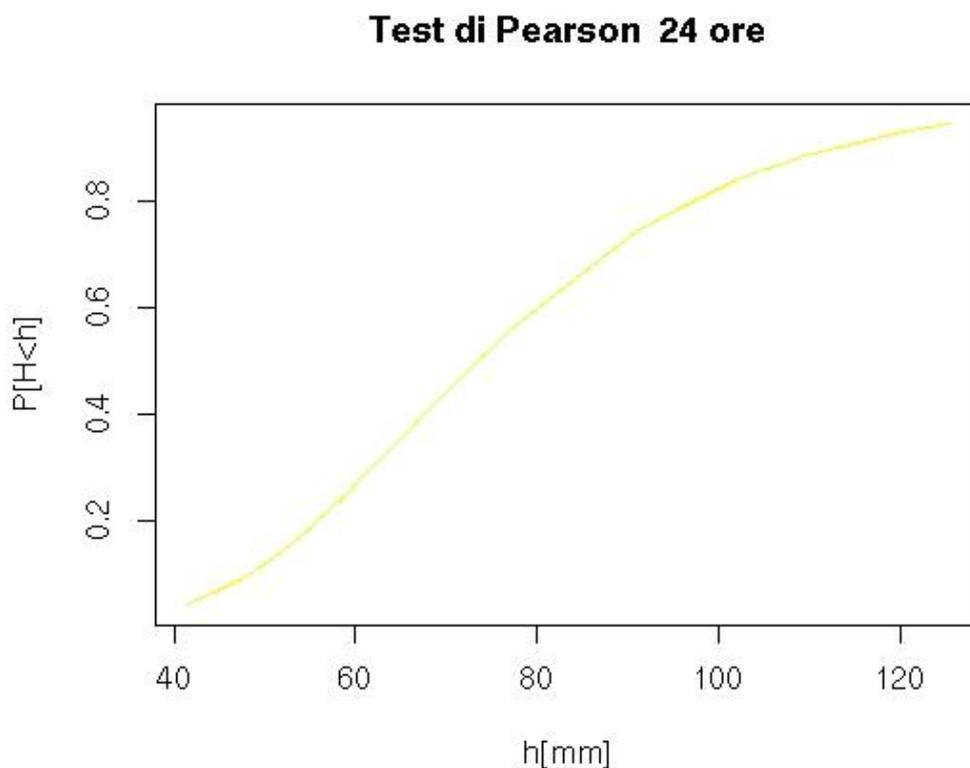


Figura 28: Test di Pearson per 24 ore con il metodo dei momenti

### 3. Costruzione delle curve di possibilità pluviometrica

#### 3.1 Curve di Gumbel

Sono stati costruiti tre vettori contenenti rispettivamente le durate e le coppie di parametri (a,b) migliori, inseriti poi all'interno di una matrice.

```
Durate <- c("0.25h", "1h", "3h", "6h", "12h", "24h")
a.gumbel <- c(10.2120423, 16.62057, 24.44839, 35.07829, 48.45171, 65.82873)
b.gumbel <- c(3.2731753, 5.622907, 8.016804, 10.77015, 16.11752, 20.91736)
summary.laste <- data.frame(Durate, a.gumbel, b.gumbel)
View(summary.laste)
```

Si elencano I comandi per la rappresentazione di più curve di possibilità di Gumbel su uno stesso grafico.

```
seq(from=1, to=150, by=0.1) -> p
p
a.gumbel[1]
b.gumbel[1]
a.gumbel[2]
b.gumbel[2]
```

```
plot(p, pgumbel(p, loc=a.gumbel[1], scale=b.gumbel[1]), type="l", col="blue",
      xlab="Precipitazioni[mm]", ylab="P[h]", main="LSPP di Trento Laste")
lines(p, pgumbel(p, loc=a.gumbel[2], scale=b.gumbel[2]), type="l",
       col="red")
lines(p, pgumbel(p, loc=a.gumbel[3], scale=b.gumbel[3]), type="l",
       col="green")
lines(p, pgumbel(p, loc=a.gumbel[4], scale=b.gumbel[4]), type="l",
       col="purple")
lines(p, pgumbel(p, loc=a.gumbel[5], scale=b.gumbel[5]), type="l",
       col="orange")
lines(p, pgumbel(p, loc=a.gumbel[6], scale=b.gumbel[6]), type="l",
       col="yellow")
```

```
text(10, 0.8, "0.25h", cex=0.8)
text(20, 0.7, "1h", cex=0.8)
text(25, 0.6, "3h", cex=0.8)
text(33, 0.5, "6h", cex=0.8)
text(41, 0.4, "12h", cex=0.8)
text(53, 0.3, "24h", cex=0.8)
```

### LSPP di Trento Laste

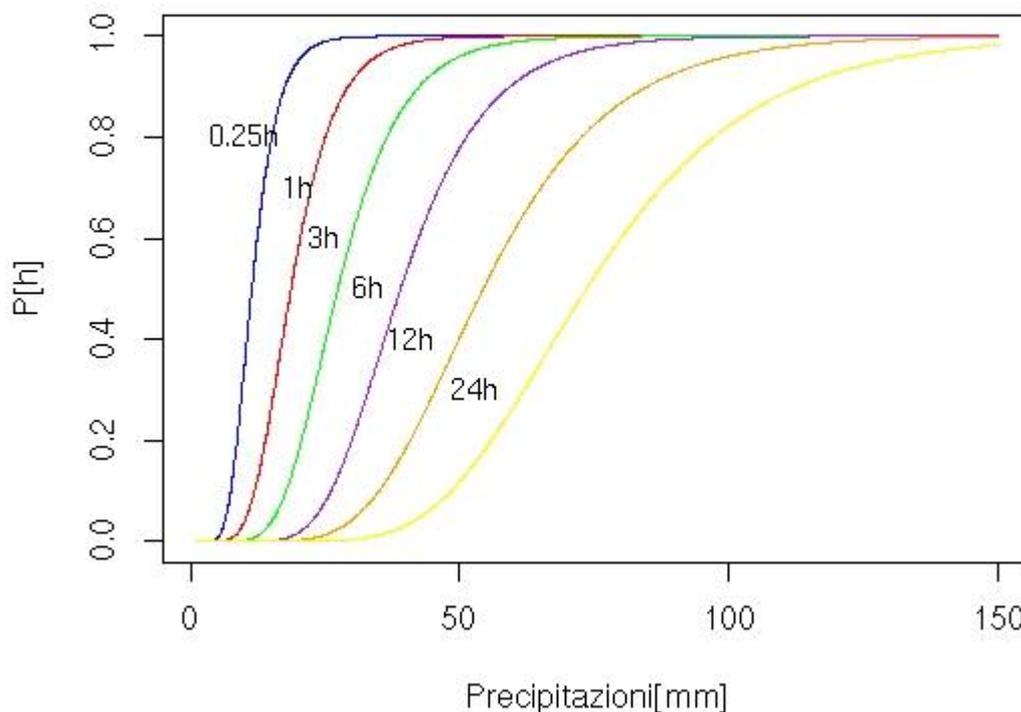


Figura 29: Curve di Gumbel

### 3.2 Curve di densità di probabilità legate alle curve di Gumbel

```
P <- function(p) {1-1/p}
```

```
P(5)
```

```
P(10)
```

```
P(20)
```

```
length(p)
```

```
rep(0.9,length(p)) -> d
```

```
length(d)
```

```
lines(p,d,col="black")
```

```
plot(p,dgumbel(p,loc=a.gumbel[1],scale=b.gumbel[1]),col="blue",type="l",
      xlab="Precipitazioni[mm]",ylab="P[h]",main="Densità di Probabilità")
```

```
lines(p,dgumbel(p,loc=a.gumbel[2],scale=b.gumbel[2]),col="red",
      type="l")
```

```
lines(p,dgumbel(p,loc=a.gumbel[3],scale=b.gumbel[3]),col="green",
      type="l")
```

```
lines(p,dgumbel(p,loc=a.gumbel[4],scale=b.gumbel[4]),col="purple",
      type="l")
```

```
lines(p,dgumbel(p,loc=a.gumbel[5],scale=b.gumbel[5]),col="orange",
      type="l")
```

```
lines(p,dgumbel(p,loc=a.gumbel[6],scale=b.gumbel[6]),col="yellow",
      type="l")
```

```
text (17,0.10,"0.25h",cex=0.8)  
text (21,0.065,"1h",cex=0.8)  
text (31,0.045,"3h",cex=0.8)  
text (45,0.03,"6h",cex=0.8)  
text (55,0.025,"12h",cex=0.8)  
text (90,0.016,"24h",cex=0.8)
```

### Densità di Probabilità

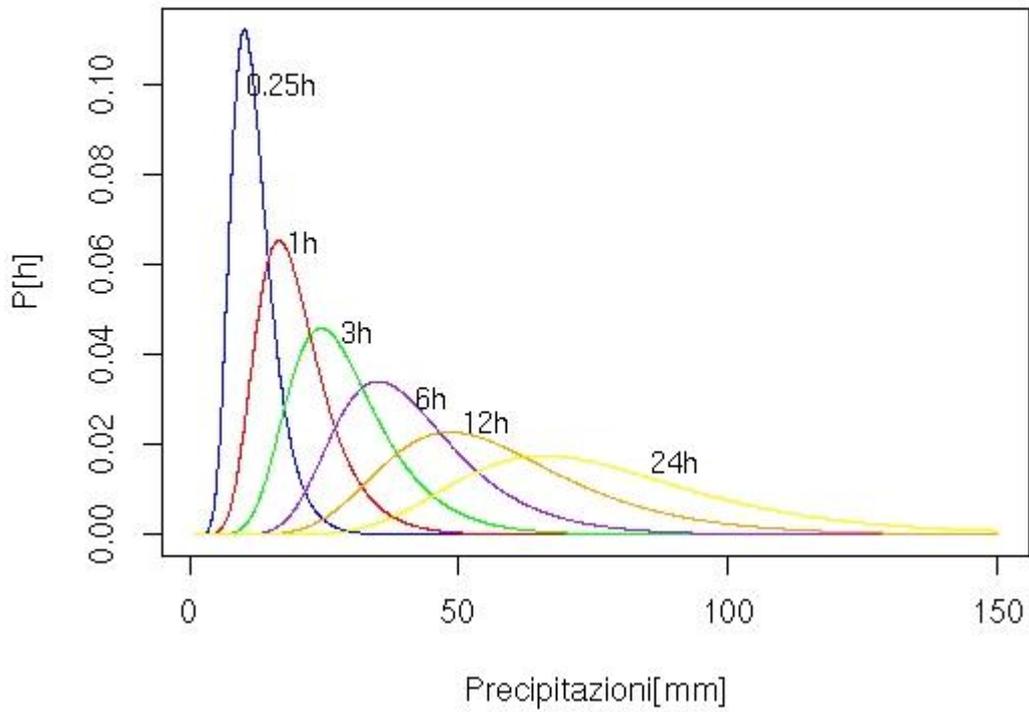


Figura 30: Curve di densità di probabilità

### 3.3 Interpolazione con tempi di ritorno

Vengono tracciate le curve di Gumbel con tempi di ritorno di 5, 10 e 20 anni.

#### 3.3.1 Tempo di ritorno di 5 anni

```
h5<-c(qgumbel(P(05),loc=a.gumbel[1],scale=b.gumbel[1]),
      qgumbel(P(05),loc=a.gumbel[2],scale=b.gumbel[2]),
      qgumbel(P(05),loc=a.gumbel[3],scale=b.gumbel[3]),
      qgumbel(P(05),loc=a.gumbel[4],scale=b.gumbel[4]),
      qgumbel(P(05),loc=a.gumbel[5],scale=b.gumbel[5]),
      qgumbel(P(05),loc=a.gumbel[6],scale=b.gumbel[6]))
```

```
h5
dd <- c(0.25,1,3,6,12,24)
idf <- data.frame(dd,h5)
View(idf)
logh5 <- log(h5)
logdd <- log(dd)
library("MASS",lib.loc="/Library/Frameworks/R.frameworks")
lsfit(logdd,logh5) -> ft05
```

Si trovano attraverso interpolazione lineare coefficiente ed esponente delle curve di possibilità pluviometrica tramite il comando:

```
ft05$coefficients
che restituisce in output:
Intercept      x
3.2327066      0.4096592
```

Per ottenere i veri coefficienti l'intercetta va esponenziata:

```
exp(ft05$coefficients[[1]])
e restituisce:
25.34817
```

#### 3.3.2 Tempo di ritorno di 10 anni

```
h10 <- c(qgumbel(P(10),loc=a.gumbel[1],scale=b.gumbel[1]),
         qgumbel(P(10),loc=a.gumbel[2],scale=b.gumbel[2]),
         qgumbel(P(10),loc=a.gumbel[3],scale=b.gumbel[3]),
         qgumbel(P(10),loc=a.gumbel[4],scale=b.gumbel[4]),
         qgumbel(P(10),loc=a.gumbel[5],scale=b.gumbel[5]),
         qgumbel(P(10),loc=a.gumbel[6],scale=b.gumbel[6]))
```

```
h10
dd <- c(0.25,1,3,6,12,24)
idf <- data.frame(dd,h10)
View(idf)
```

```
logh10 <- log(h10)
logdd <- log(dd)
library("MASS", lib.loc="/Library/Frameworks/R.frameworks")
lsfit(logdd, logh10) -> ft10
```

Si trovano attraverso interpolazione lineare coefficiente ed esponente delle curve di possibilità pluviometrica tramite il comando:

```
ft10$coefficients
```

che restituisce in output:

```
Intercept      x
3.3847115      0.4092448
```

Per ottenere i veri coefficienti l'intercetta va esponenziata:

```
exp(ft10$coefficients[[1]])
```

e restituisce:

```
29.50948
```

### 3.3.3 Tempo di ritorno di 20 anni

```
h20 <- c(qgumbel(P(20), loc=a.gumbel[1], scale=b.gumbel[1]),
        qgumbel(P(20), loc=a.gumbel[2], scale=b.gumbel[2]),
        qgumbel(P(20), loc=a.gumbel[3], scale=b.gumbel[3]),
        qgumbel(P(20), loc=a.gumbel[4], scale=b.gumbel[4]),
        qgumbel(P(20), loc=a.gumbel[5], scale=b.gumbel[5]),
        qgumbel(P(20), loc=a.gumbel[6], scale=b.gumbel[6]))
```

```
h20
```

```
dd <- c(0.25, 1, 3, 6, 12, 24)
```

```
idf <- data.frame(dd, h20)
```

```
View(idf)
```

```
logh20 <- log(h20)
```

```
logdd <- log(dd)
```

```
library("MASS", lib.loc="/Library/Frameworks/R.frameworks")
```

```
lsfit(logdd, logh20) -> ft20
```

Si trovano attraverso interpolazione lineare coefficiente ed esponente delle curve di possibilità pluviometrica tramite il comando:

```
ft20$coefficients
```

che restituisce in output:

```
Intercept      x
3.5115732      0.4089436
```

Per ottenere i veri coefficienti l'intercetta va esponenziata:

```
exp(ft20$coefficients[[1]])
```

e restituisce:

```
33.50093
```

### 3.4 LSPP

Le Linee Segnalatrici di Possibilità Pluviometrica sono governate dalla seguente equazione:  $h(t_p, T_r) = a(T_r)t_p^n$ , dove  $h$  è l'altezza di precipitazione,  $t_p$  è la durata considerata,  $T_r$  è il tempo di ritorno ed  $n$  è un esponente non dipendente dal tempo di ritorno.

```
plot(dd, exp(ft10$coefficients[[1]]) * dd^ft10$coefficients[[2]], type="l",
      col="blue", xlab="t[ore]", ylab="h[mm]", main="Linee Segnalatrici
      di Possibilità Pluviometrica")
lines(dd, exp(ft20$coefficients[[1]]) * dd^ft20$coefficients[[2]],
      type="l", col="red")
lines(dd, exp(ft05$coefficients[[1]]) * dd^ft05$coefficients[[2]],
      type="l", col="green")

points(dd, h10, pch=0)
points(dd, h20, pch=1)
points(dd, h5, pch=2)
```

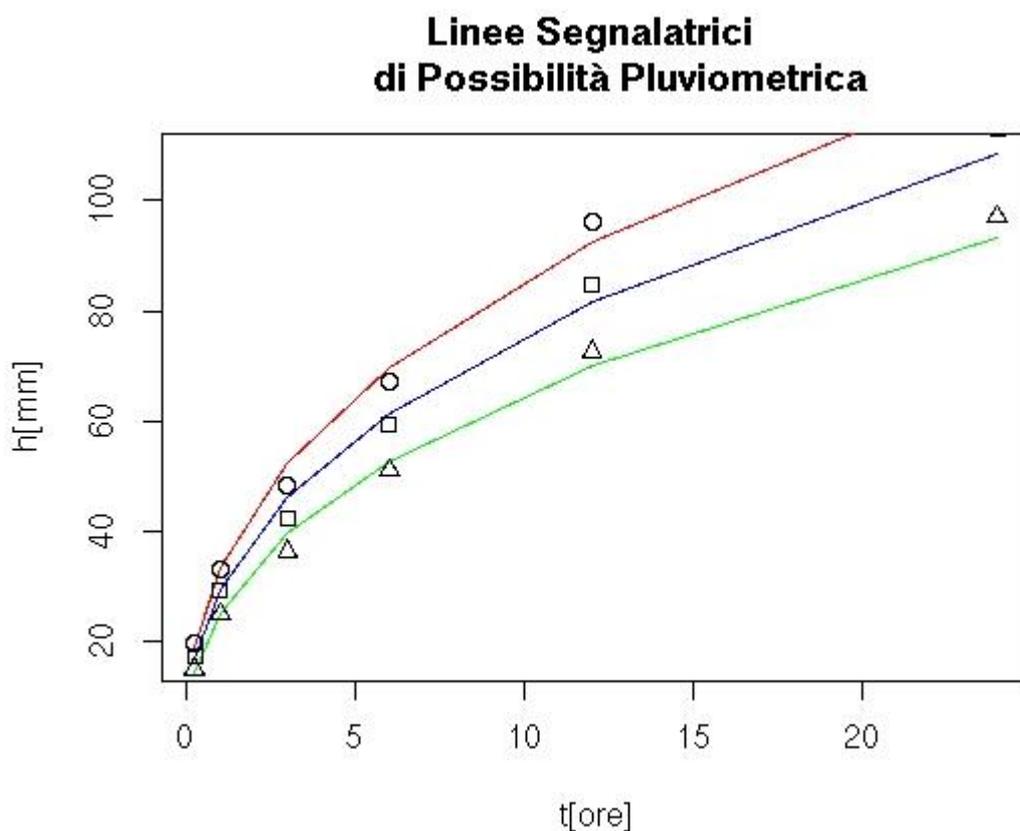


Figura 31: Linee Segnalatrici di Possibilità Pluviometrica

In verde si ha la curva con tempo di ritorno 5 anni, in blu con tempo di ritorno di 10 anni ed in rosso con tempo di ritorno di 20 anni.

### 3.5 LSPP linearizzate

Le curve sopra sono state linearizzate e visualizzate tramite un grafico bi-logaritmico.

```
plot(dd,exp(ft10$coefficients[[1]])*dd^ft10$coefficients[[2]],type="l",
      col="blue",xlab="t[ore]",ylab="h[mm]",main="Linee Segnalatrici
      di Possibilità Pluviometrica",log="xy")
lines(dd,exp(ft20$coefficients[[1]])*dd^ft20$coefficients[[2]],
      type="l",col="red",log="xy")
lines(dd,exp(ft05$coefficients[[1]])*dd^ft05$coefficients[[2]],
      type="l",col="green",log="xy")
points(dd,h10,pch=0)
points(dd,h20,pch=1)
points(dd,h5,pch=2)
```

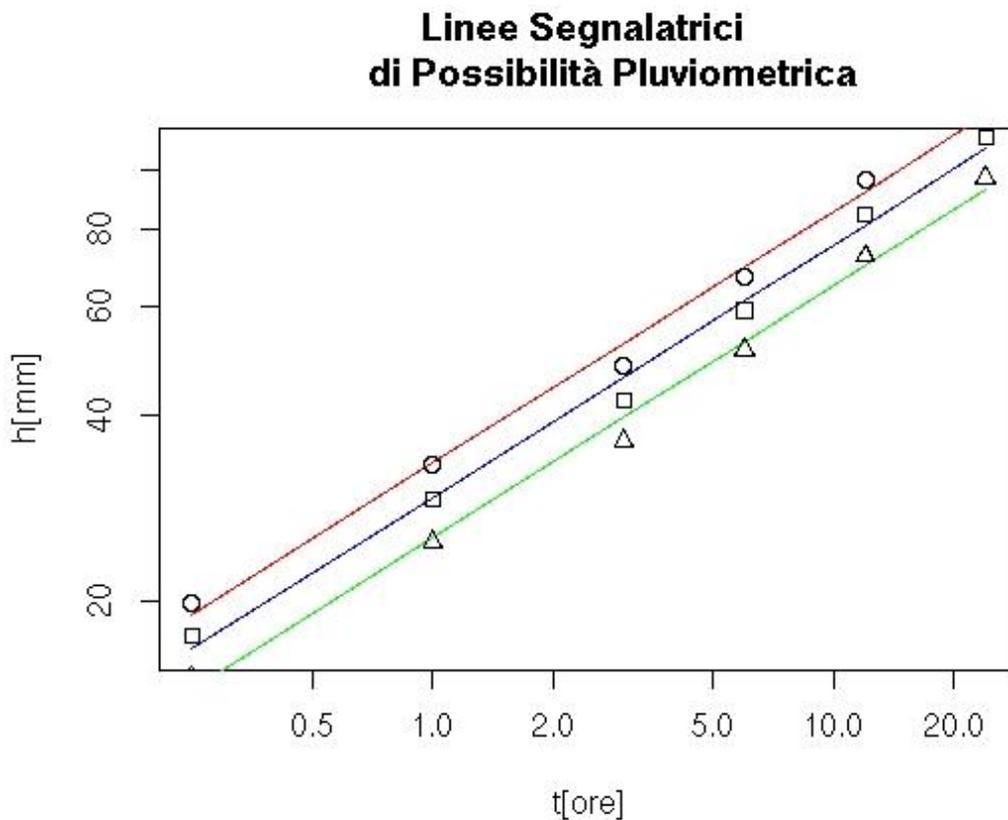


Figura 32: Linee Segnalatrici di Possibilità Pluviometrica, grafico bi-logaritmico

In verde si ha la curva con tempo di ritorno di 5 anni, in blu con tempo di ritorno di 10 anni ed in rosso con tempo di ritorno di 20 anni.

**A. Programmi utilizzati**

- *RStudio*: calcolo delle curve di possibilità pluviometrica  
<http://r-project.org>

**B. Riferimenti bibliografici**

- R. Rigon, M. Dall'Amico "Finding the LSPP with R": lucidi del corso di Idrologia, 6 marzo 2012  
<http://www.slideshare.net/SlidesAboutHydrology/presentations>
- Meteotrentino: reperimento dati pluviometrici  
<http://www.meteotrentino.it>