

```

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

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

dim(data)[[2]] -> num
num

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

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]]) -> m1h
var(data[[3]]) -> v1h
pi
#metodo dei momenti 1h
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)

```

```

#metodo dei minimi quadrati lh
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

#metodo della massima verosimiglianza lh
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

#test di Pearson lh
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
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
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
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
#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")

#15 minuti.....
data15 <- data[[2]]
anni <- data[[1]]
anni <- anni[!is.na(data15)]
data15 <- data15[!is.na(data15)]
ecdf(data[[2]]) -> x
x
plot(x, xlab="h[mm]", ylab="P[H<h]", main="Frequenza di non superamento")
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
pi
#metodo dei momenti 15min
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)

#metodo dei minimi quadrati 15min
ec = ecdf(data15)
length(data15)
Fi = ec(sort(data15))
Fi
Y = -(log(-log(Fi[-25])))
Y
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

#metodo della massima verosimiglianza 15min
load(MASS)

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

#test di Pearson 15min
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
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
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
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
#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")

#3 ore.....
ecdf(data[[4]]) -> z
z
plot(z, xlab="h[mm]", ylab="P[h<h]", main="Frequenza di non superamento")

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
pi
#metodo dei momenti 3h
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)

#metodo dei minimi quadrati 3h
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

#metodo della massima verosimiglianza 3h
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

#test di Pearson 3h
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
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
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
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
#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")

#6 ore.....
ecdf(data[[5]]) -> u
u
plot(u, xlab="h[mm]", ylab="P[h<h]", main="Frequenza di non superamento")

```

```

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
pi
#metodo dei momenti 6h
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)

#metodo dei minimi quadrati 6h
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

#metodo della massima verosimiglianza 6h
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

#test di Pearson 6h
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
qqgumbel (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
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
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
#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")

#12 ore.....
ecdf(data[[6]]) -> v
v
plot(v, xlab="h[mm]", ylab="P[h<h]", main="Frequenza di non superamento")

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
pi
#metodo dei momenti 12h
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)

#metodo dei minimi quadrati 12h
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

#metodo della massima verosimiglianza 12h
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

#test di Pearson 6h
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
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
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
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
#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")

#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")
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
pi
#metodo dei momenti 24h
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)
```

```
#metodo dei minimi quadrati 24h
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
```

```
#metodo della massima verosimiglianza 24 ore
load(MASS)
```

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

```
#test di Pearson 24h
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
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
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
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
#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")

#calcolo delle curve di possibilità pluviometrica
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)
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, 0.50, "6h", cex=0.8)
text(41, 0.40, "12h", cex=0.8)
text(53, 0.3, "24h", cex=0.8)

#densità di probabilità
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)

#LSPP con tempi di ritorno
#10anni
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
ft10$coefficients
exp(ft10$coefficients[[1]])

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

```

```

#5anni
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
ft05$coefficients
exp(ft05$coefficients[[1]])

#plot
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)

#linearizzo le curve
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)

```