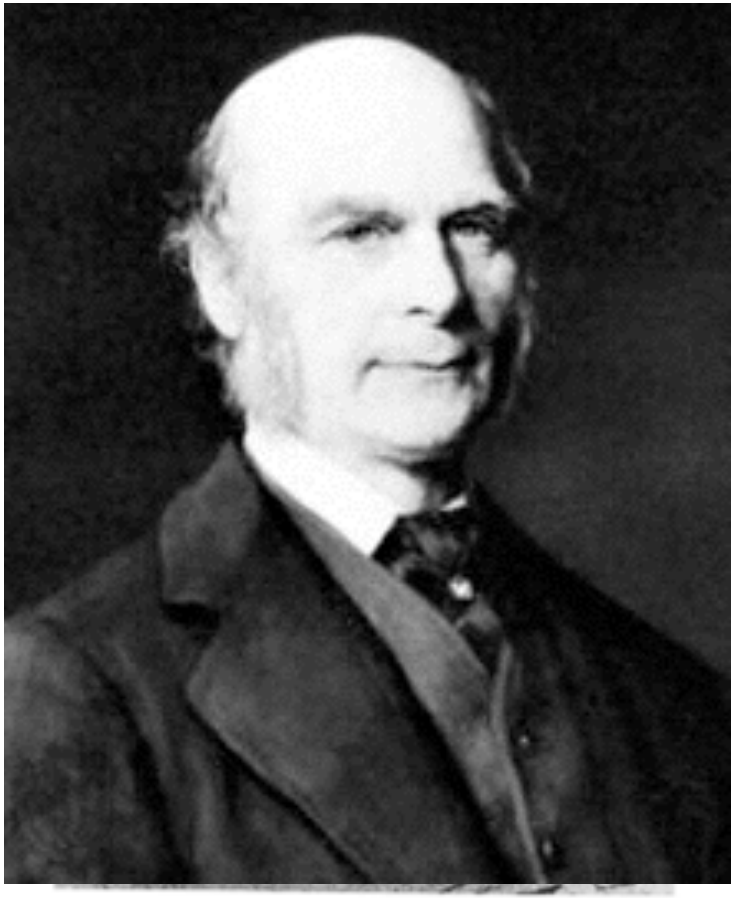
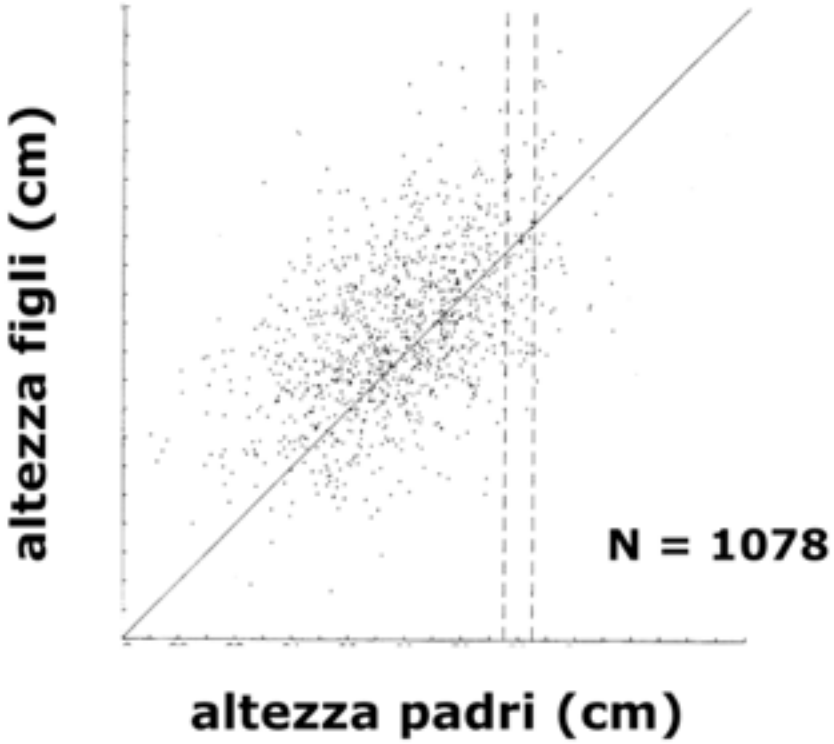
prevedere y in funzione di x, per variabili linearmente associate

concetto di regressione

Francis Galton
(1822-1911)





idea di base

se x e y sono associate, posso usare x per prevedere y

(con una certa approssimazione)

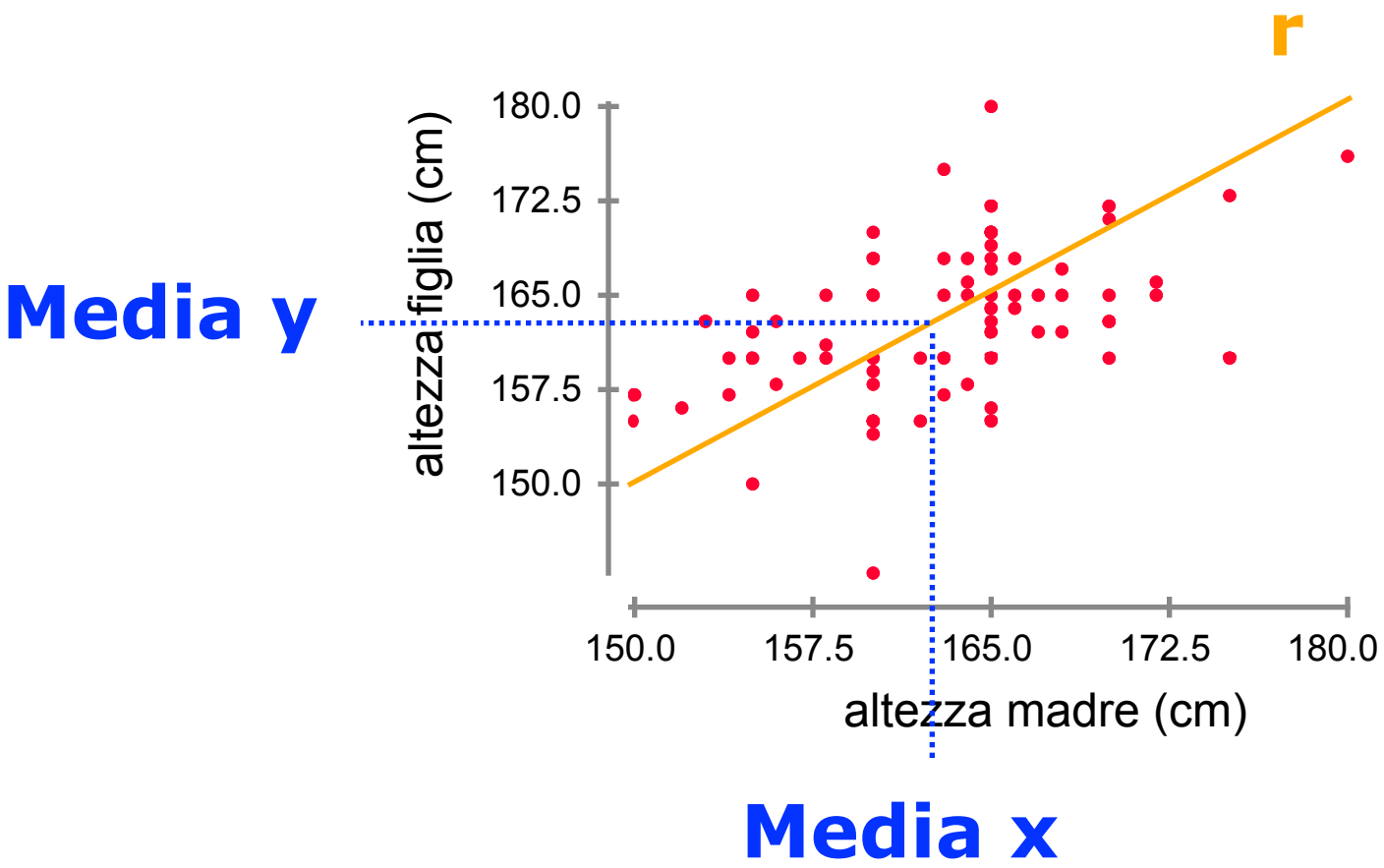
per farlo devo conoscere:

Mx, DSx

My, DSy

r

esempio


r
Media y
180.0
172.5
165.0
157.5
150.0
altezza figlia (cm)
150.0
157.5
165.0
172.5
180.0
altezza madre (cm)
Media x

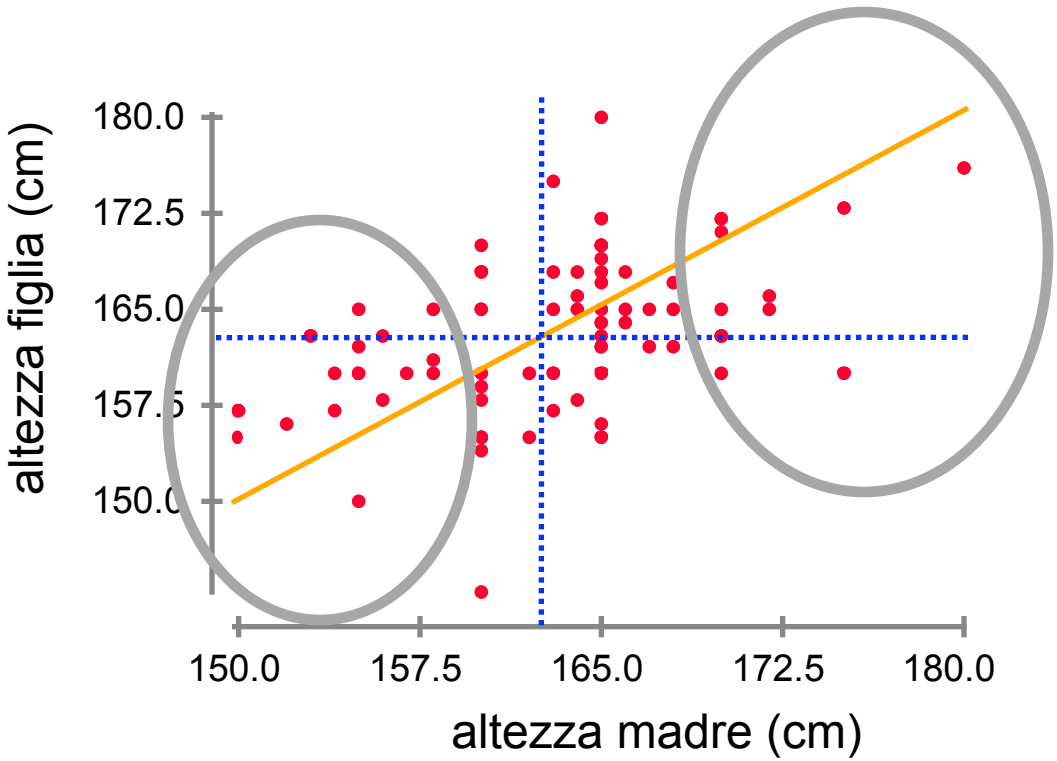
problema

è evidente che y tende a crescere al crescere di x

ma di quanto?

la risposta NON è la retta delle DS!

il motivo


180.0
172.5
165.0
157.5
150.0
altezza figlia (cm)
150.0
157.5
165.0
172.5
180.0
altezza madre (cm)

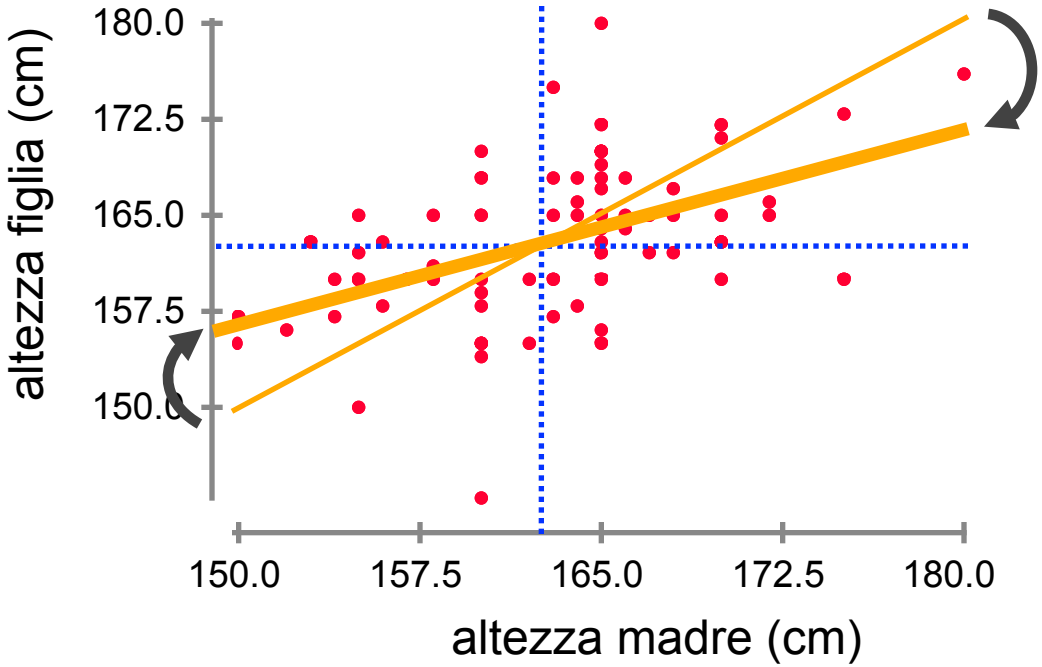
y --> My

per gli x sopra Mx, gli y tendono a spostarsi verso il basso

per gli x sotto Mx, gli y tendono a spostarsi verso l'alto

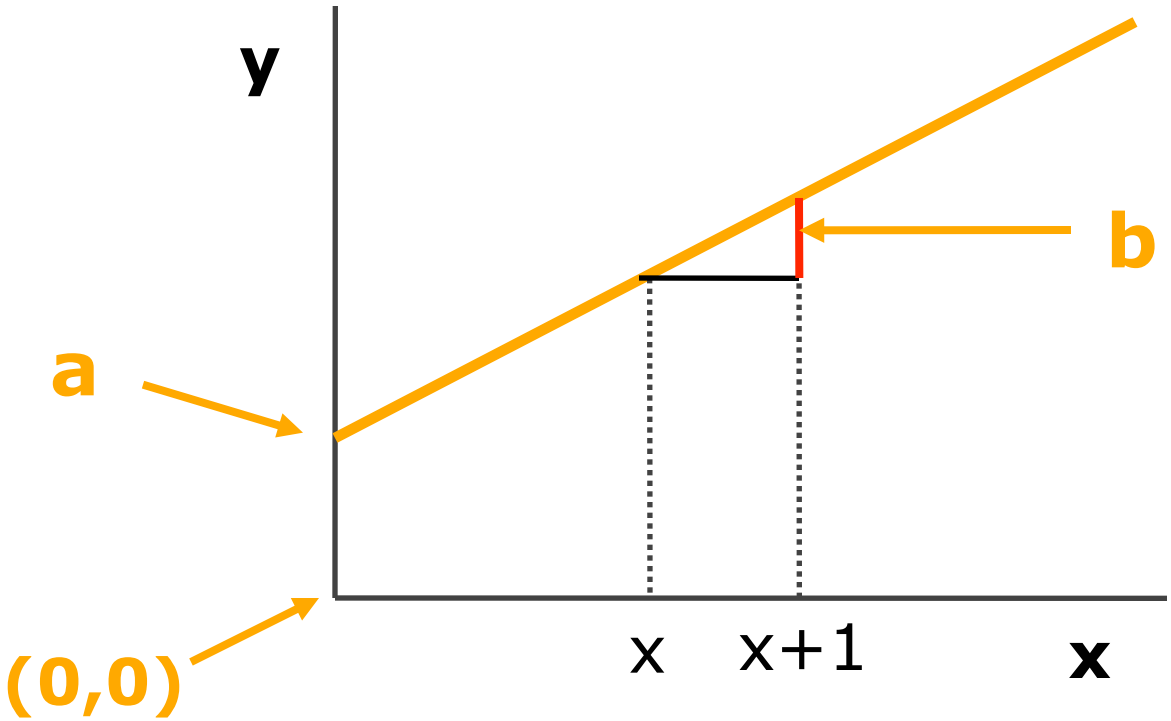
"Sir" Francis Galton chiamava questo fenomeno *regression toward mediocrity*

soluzione:



la retta di regressione

$y = a + bx$



figlia = a + b x madre

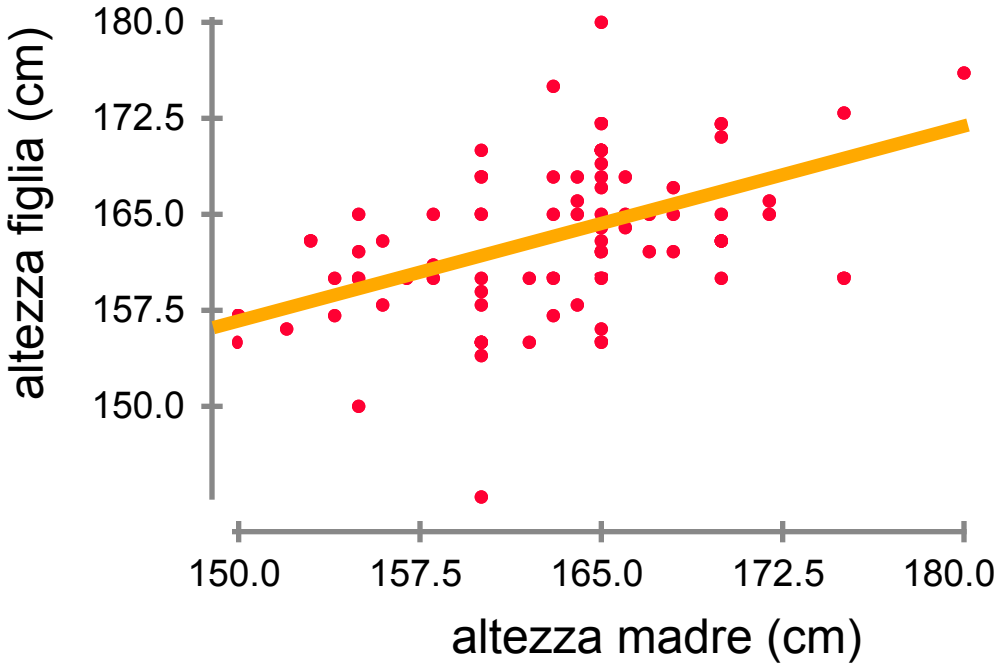


Grafico delle medie

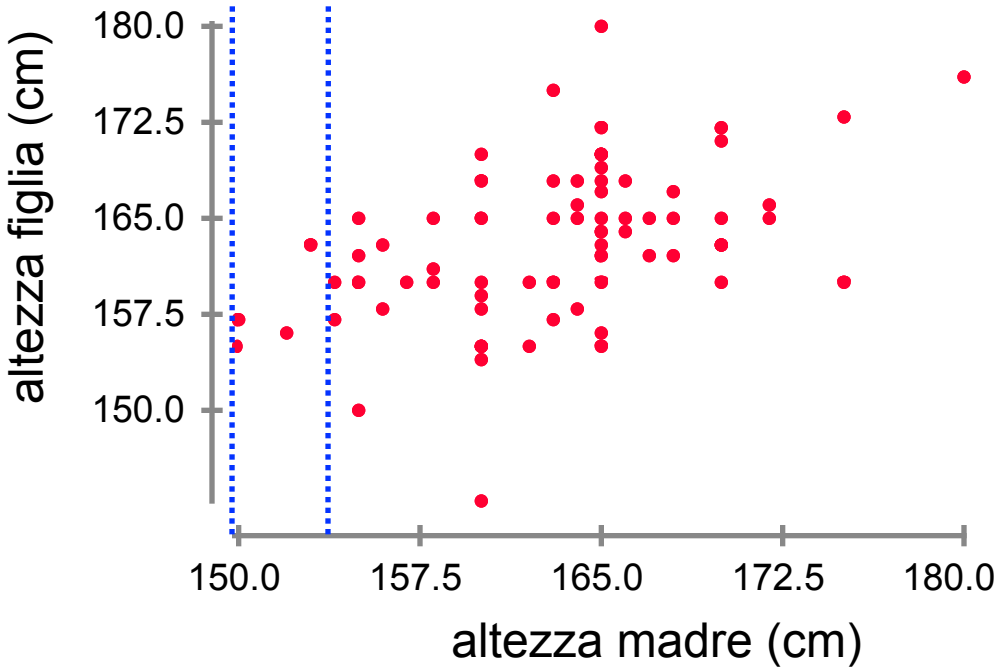


grafico delle medie

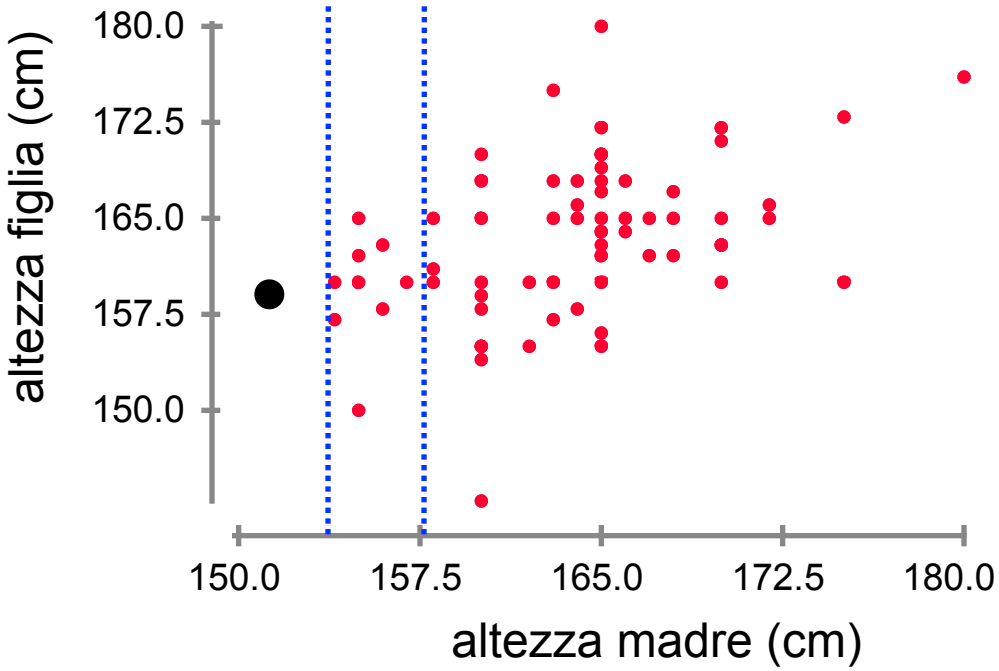


grafico delle medie

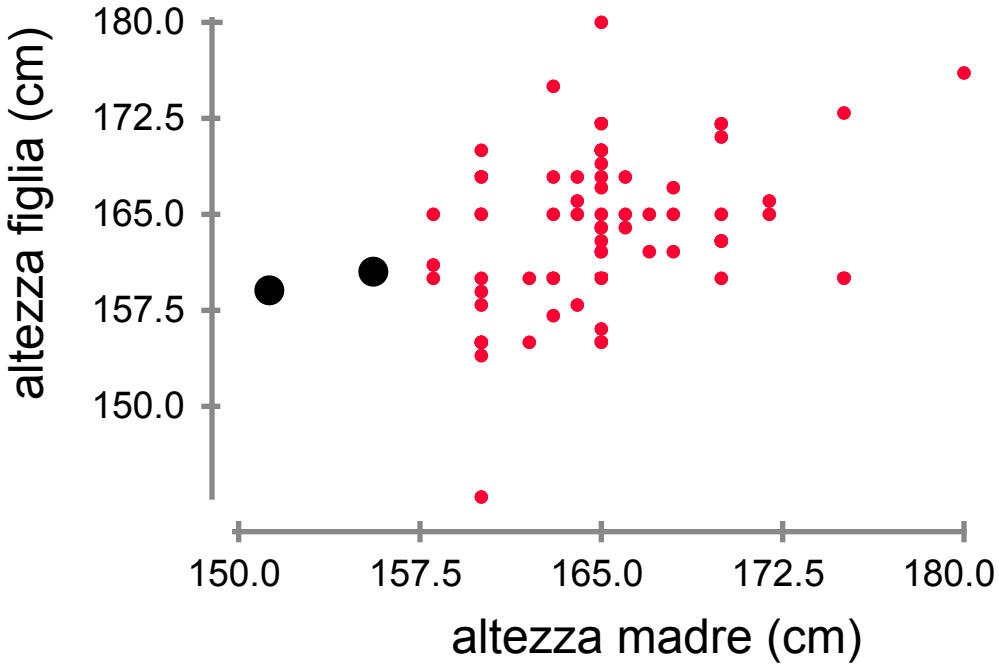


grafico delle medie

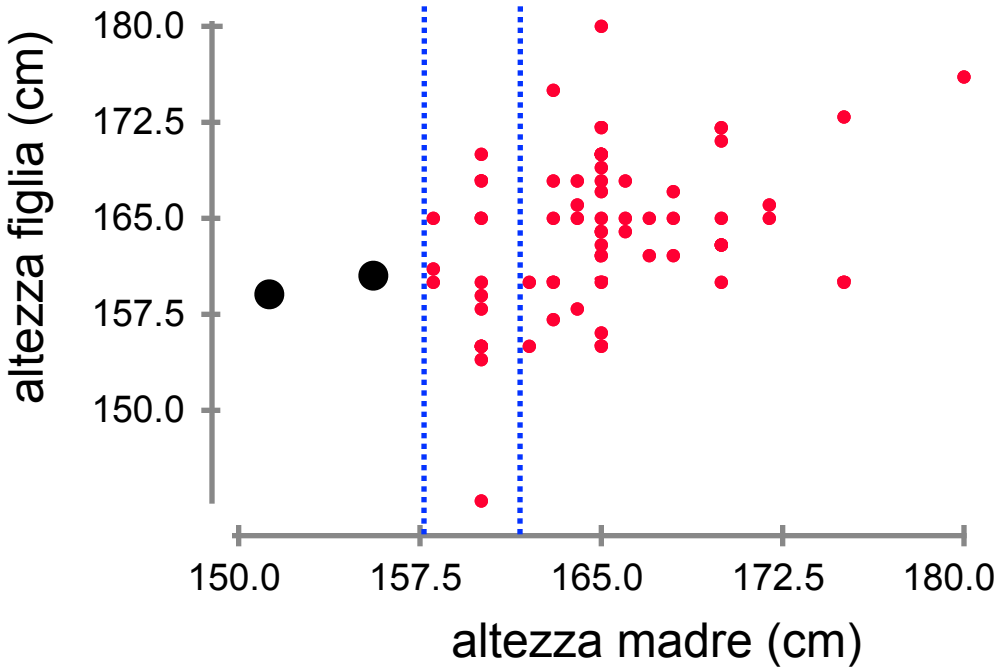


grafico delle medie

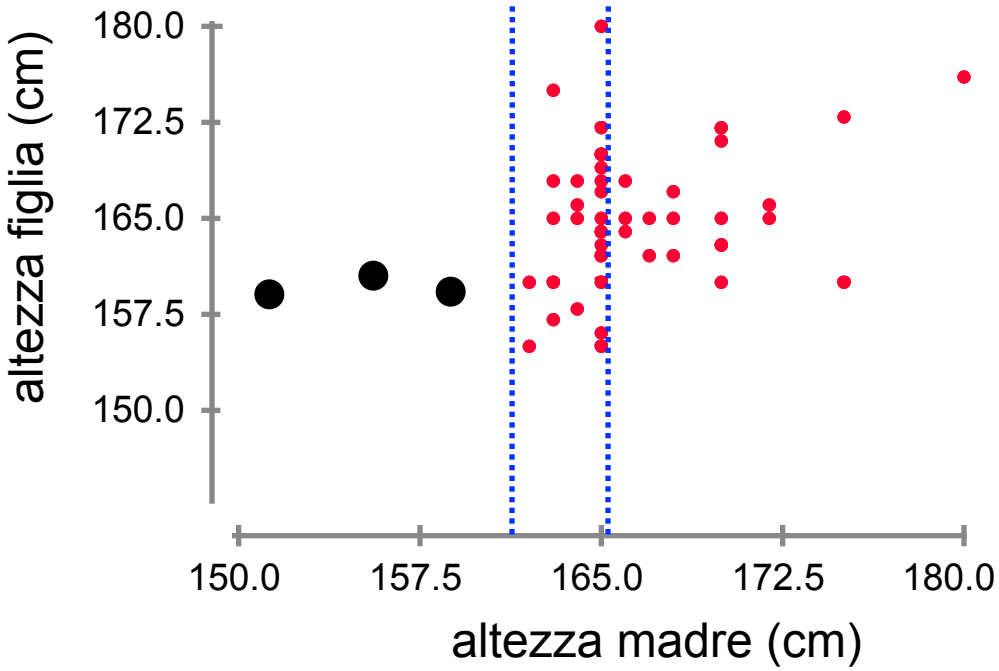
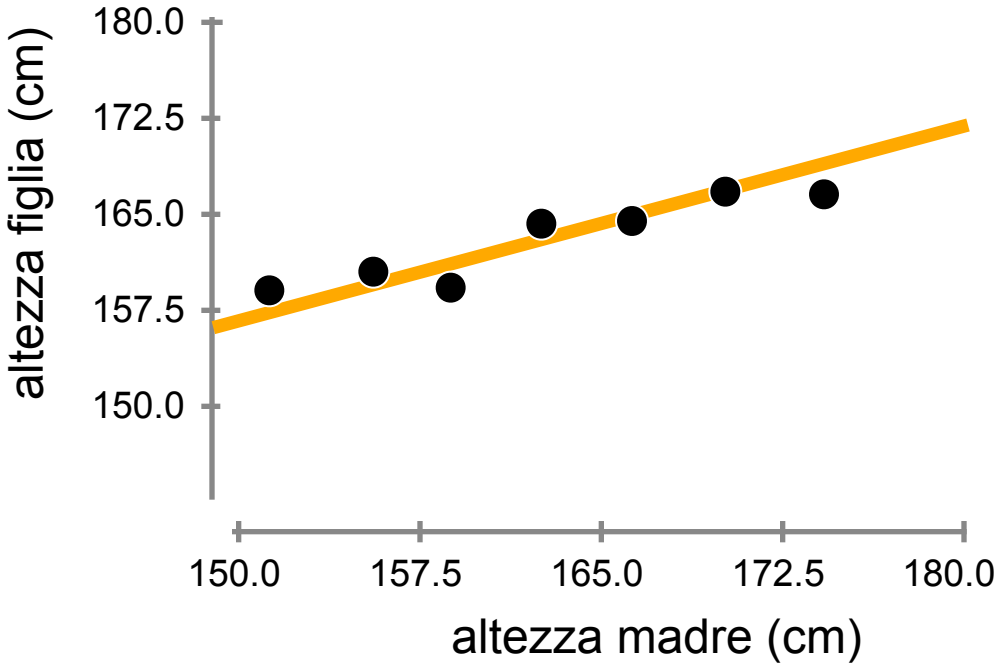
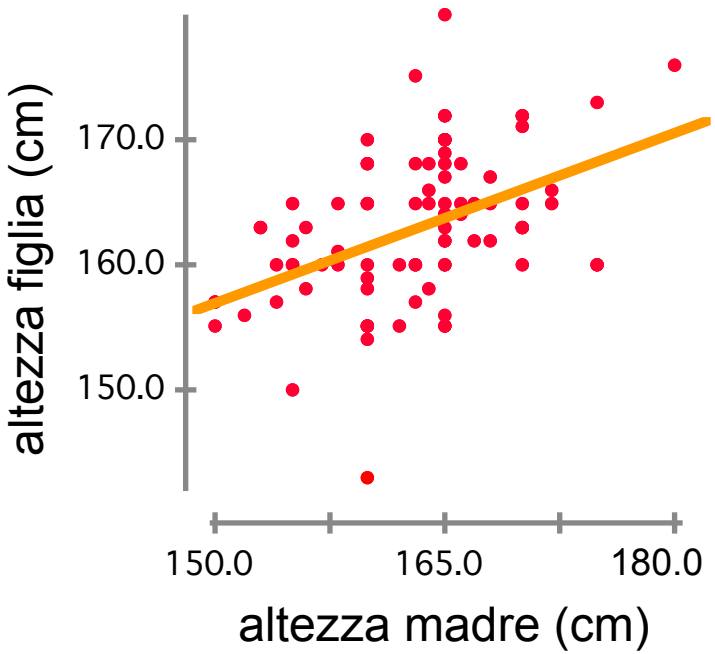
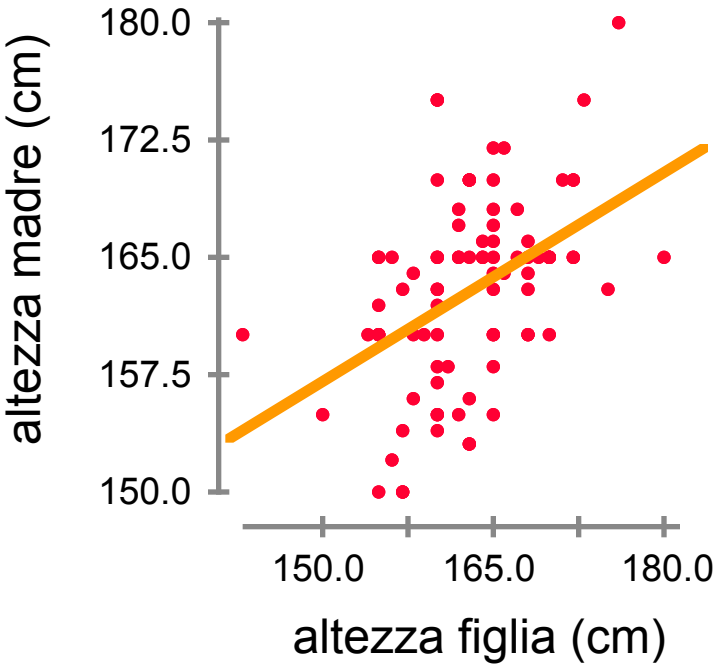


grafico delle medie



due rette diverse





quella di prima

l'altra

le rette di regressione sono due

r (x.y) = r (y.x)

la retta delle DS è una sola!

ma b (x.y) ≠ b (y.x)

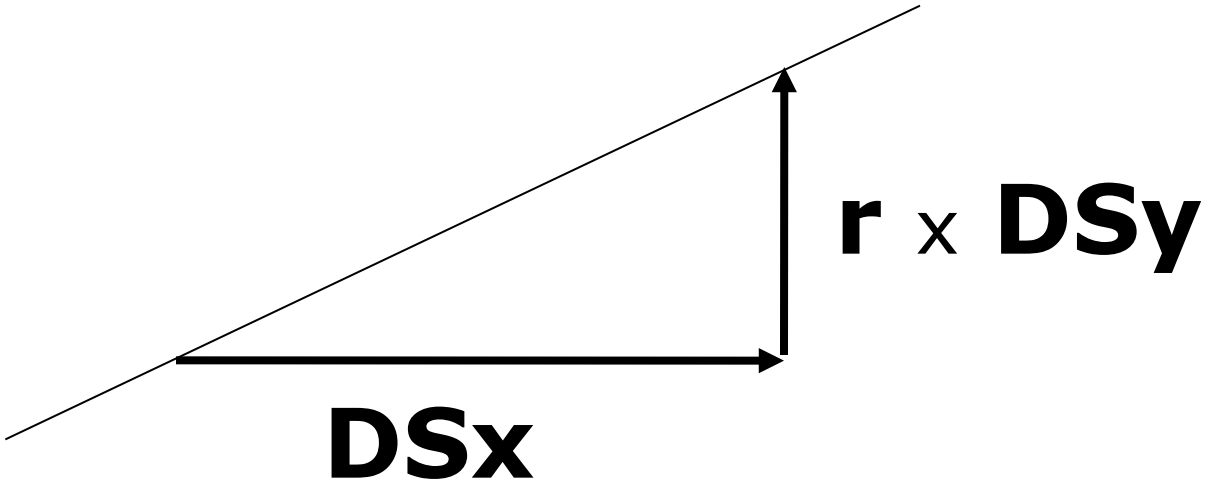
e anche a (x.y) ≠ a (y.x)

salvo un caso particolare

quale?

r, b e le DS

r misura di quanto cresce y, in unità standard, quando x cresce di 1 DSx



r, b e le DS

r misura di quanto cresce y, in unità standard, quando x cresce di 1 DSx

$$b = \frac{r\ DSy}{DSx}$$

quindi se DSy=DSx cosa succede?

r, b e le DS

se DSy = DSx, SDy/SDx = 1 e b = r

pertanto b(y.x) = b(x.y)

dunque sono parallele; dato che hanno un punto in comune (My, Mx)

anche a(y.x) = a(x.y)

retta di regressione = retta delle DS

i parametri della retta

$$b = r \frac{DSy}{DSx}$$

$$My = a + bMx$$

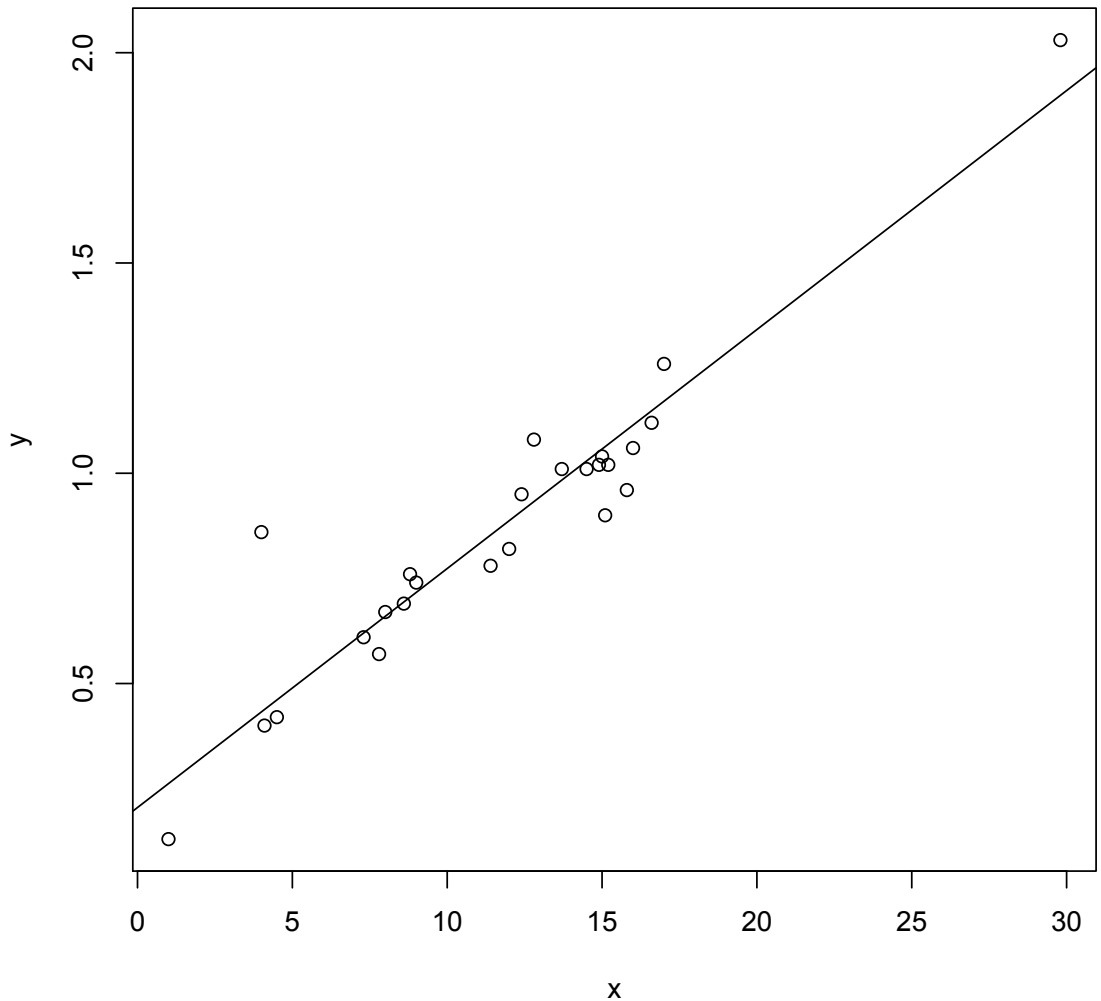
$$a = My - bMx$$

```
> df <- read.table("SIGAR.txt", header = TRUE)
> names(df)

[1] "MARCA"       "CATRAME.mg"  "NICOTINA.mg"
"PESO.g"      "AC.mg"

> y <- df$NICOTINA.mg
> x <- df$CATRAME.mg
> byx <- cor(x, y) * sd(y)/sd(x)
> ayx <- mean(y) - byx * mean(x)

> c(ayx, byx)

[1] 0.20515580 0.05682731
```

```
> c(ayx, byx)

[1] 0.20515580 0.05682731

> lm(y ~ x)

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
    0.20516      0.05683
```

```
> c(ayx, byx)

[1] 0.20515580 0.05682731

> bxy <- cor(x, y) * sd(x)/sd(y)
> axy <- mean(x) - bxy * mean(y)
> c(axy, bxy)

[1] -1.932751 15.683194
> lm(x ~ y)

Call:
lm(formula = x ~ y)

Coefficients:
(Intercept)            y
    -1.933       15.683
```

```
> plot(x, y)
> abline(lm(y ~ x))
```



```
> o <- lm(y ~ x)
> summary(o)

Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.16325 -0.05439 -0.01915  0.02631  0.42753 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 0.205156   0.054405   3.771 0.000992 ***
x           0.056827   0.004139  13.728 1.44e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Residual standard error: 0.1193 on 23 degrees of freedom
Multiple R-squared: 0.8912, Adjusted R-squared: 0.8865 
F-statistic: 188.5 on 1 and 23 DF,  p-value: 1.442e-12
```

```
> o <- lm(y ~ x)
> coef(o)

(Intercept)           x
 0.20515580  0.05682731

> c <- coef(o)
> c[[1]]

[1] 0.2051558

> c[[2]]

[1] 0.05682731
```

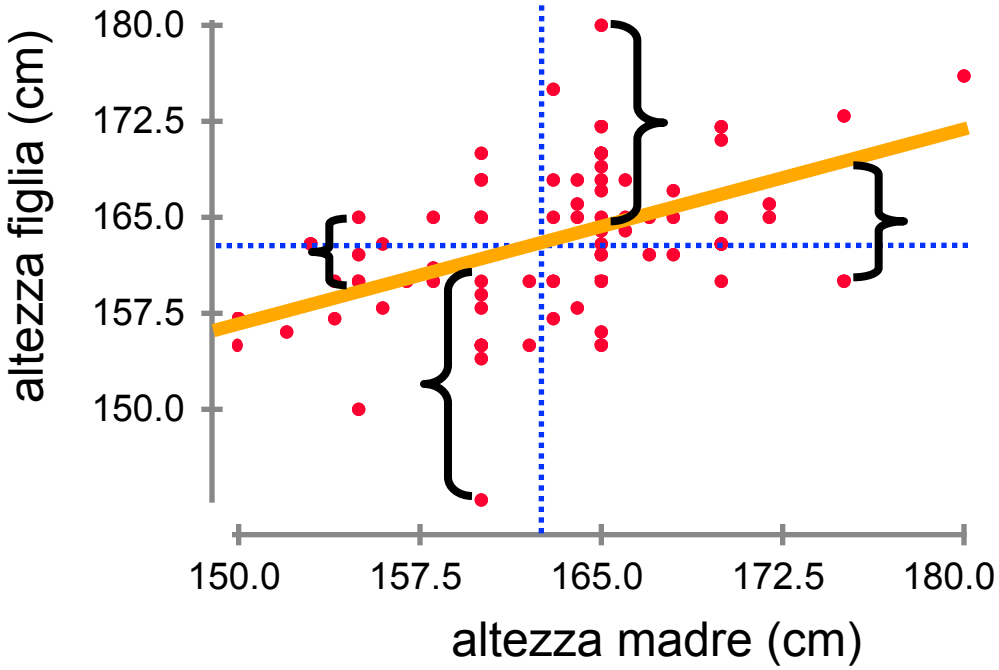
l’errore di previsione

i limiti della regressione

il metodo della regressione non è perfetto

salvo casi particolari, le osservazioni y tipicamente non sono esattamente sulla retta di regressione

I residui


180.0
172.5
165.0
157.5
150.0
altezza figlia (cm)
150.0
157.5
165.0
172.5
180.0
altezza madre (cm)

Definizione

i residui (o errori) della regressione sono gli scarti: valore osservato - valore sulla retta

i valori sulla retta sono i valori previsti dal lm, indicati anche con il simbolo $\hat{y}$

(che si legge *y-hat*)

errore RMS

l’errore RMS è il RMS dei residui

(detto anche RMSE - *root mean square error)*

l’errore RMS misura l'errore globale di previsione

errore RMS

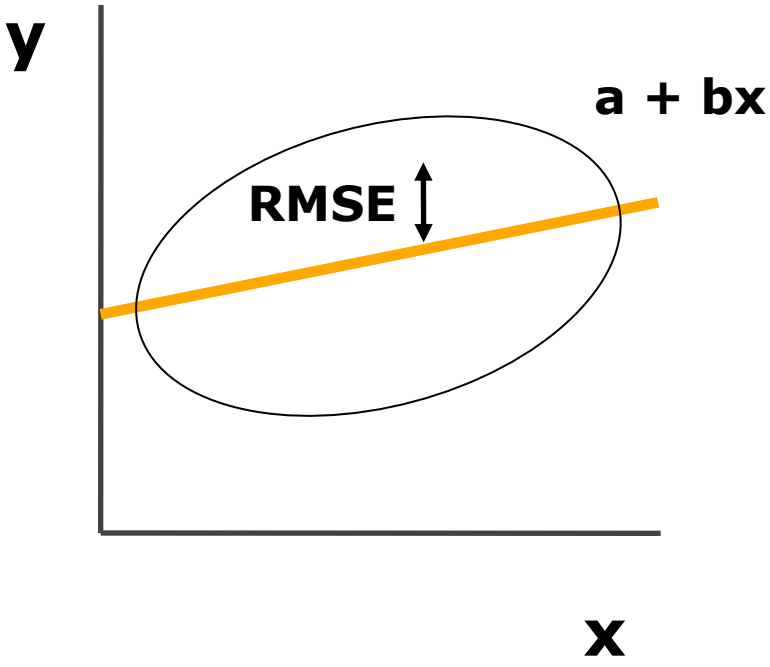
$$\text{RMSE} = \sqrt{\frac{\text{somma(residui^2)}}{\text{n - 2}}}$$

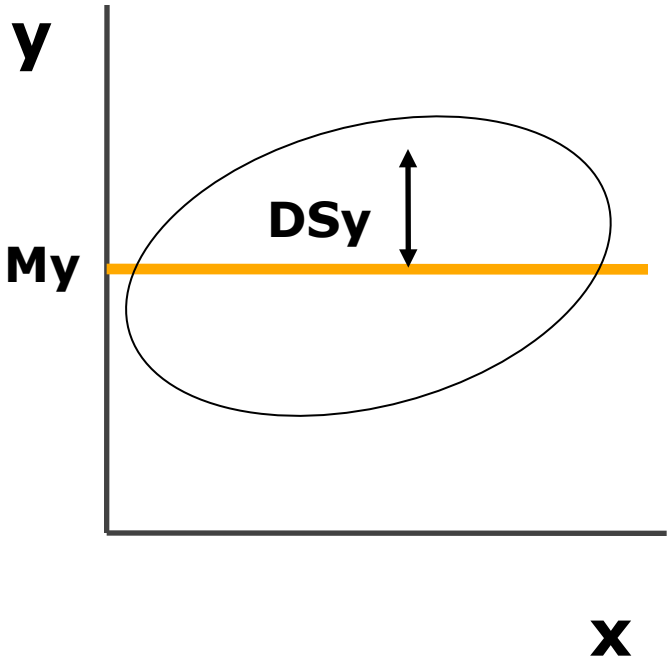
altra definizione del RMSE

$$RMSE = \sqrt{1 - r^2}\, DSy$$

quanto vale RMSE
se r = 1?
se r = 0?

Il RMSE e la DSy





Minimi quadrati

il metodo della regressione viene chiamato "dei minimi quadrati" (*least squares o anche OLS)*

perché la retta di regressione è la retta per cui il RMSE è il più piccolo possibile

```
> df <- read.table("SIGAR.txt", header = TRUE)

> y <- df$NICOTINA.mg
> x <- df$CATRAME.mg

> o <- lm(y ~ x)
> c <- coef(o)
> yhat <- c[[1]] + c[[2]] * x
> res <- y - yhat
> res

 [1]  0.427534953 -0.054392778  0.131390330  0.010225709 -0.038147778
 [6] -0.017565467  0.054763861  0.040185541 -0.028489165 -0.031882736
[11]  0.026310037 -0.163248198 -0.078408828 -0.072987148  0.023398398
[16] -0.131983114  0.088779911  0.147454617 -0.143027316 -0.040878702
[21] -0.019151812 -0.009995173 -0.003870677 -0.048930930 -0.067083534
```

```
> df <- read.table("SIGAR.txt", header = TRUE)

> y <- df$NICOTINA.mg
> x <- df$CATRAME.mg

> o <- lm(y ~ x)
> resid(o)          # estrae i residui direttamente da o

         1          2           3           4           5           6
 0.427534953 -0.054392778  0.131390330  0.010225709 -0.038147778 -0.017565467
         7          8           9          10          11          12
 0.054763861  0.040185541 -0.028489165 -0.031882736  0.026310037 -0.163248198
        13         14          15          16          17          18
-0.078408828 -0.072987148  0.023398398 -0.131983114  0.088779911  0.147454617
        19         20          21          22          23          24
-0.143027316 -0.040878702 -0.019151812 -0.009995173 -0.003870677 -0.048930930
        25
-0.067083534
```

```
> df <- read.table("SIGAR.txt", header = TRUE)

> y <- df$NICOTINA.mg
> x <- df$CATRAME.mg

> o <- lm(y ~ x)
> fitted(o)    # estrae gli yhat  direttamente da o

       1        2        3        4        5        6        7        8
0.4324650 1.1143928 1.8986097 0.6597743 0.4381478 1.0575655 0.7052361 0.9098145
       9       10       11       12       13       14       15       16
1.1484892 1.0518827 0.9836900 1.0632482 0.6484088 0.8529871 0.7166016 0.2619831
      17       18       19       20       21       22       23       24
1.1712201 0.9325454 1.1030273 0.4608787 1.0291518 0.6199952 0.6938707 1.0689309
      25
0.8870835
```

```
> o <- lm(y ~ x)
> r <- resid(o)
> n <- length(x)
> rmse <- sqrt(sum(r^2)/(n - 2))
> rmse
[1] 0.1192789

> s <- sqrt(sum((y - mean(y))^2)/(n
- 2))
> sqrt(1 - cor(x, y)^2) * s
[1] 0.1192789
```

```
> summary(lm(y ~ x))

Call:
lm(formula = y ~ x)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.16325 -0.05439 -0.01915  0.02631  0.42753 

Coefficients:
            Estimate       Std. Error    t value Pr(>|t|)    
(Intercept) 0.205156   0.054405   3.771 0.000992 ***
x           0.056827   0.004139  13.728 1.44e-12 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Residual standard error: 0.1193 on 23 degrees of freedom
Multiple R-squared: 0.8912, Adjusted R-squared: 0.8865
F-statistic: 188.5 on 1 and 23 DF, p-value: 1.442e-12

diagnosticare i problemi

limiti del metodo

il metodo della regressione lineare funziona se vale una condizione detta *normalità bivariata*

ossia: se il diagramma di dispersione ha la caratteristica forma “a uovo” (sigaro, pallone da rugby)

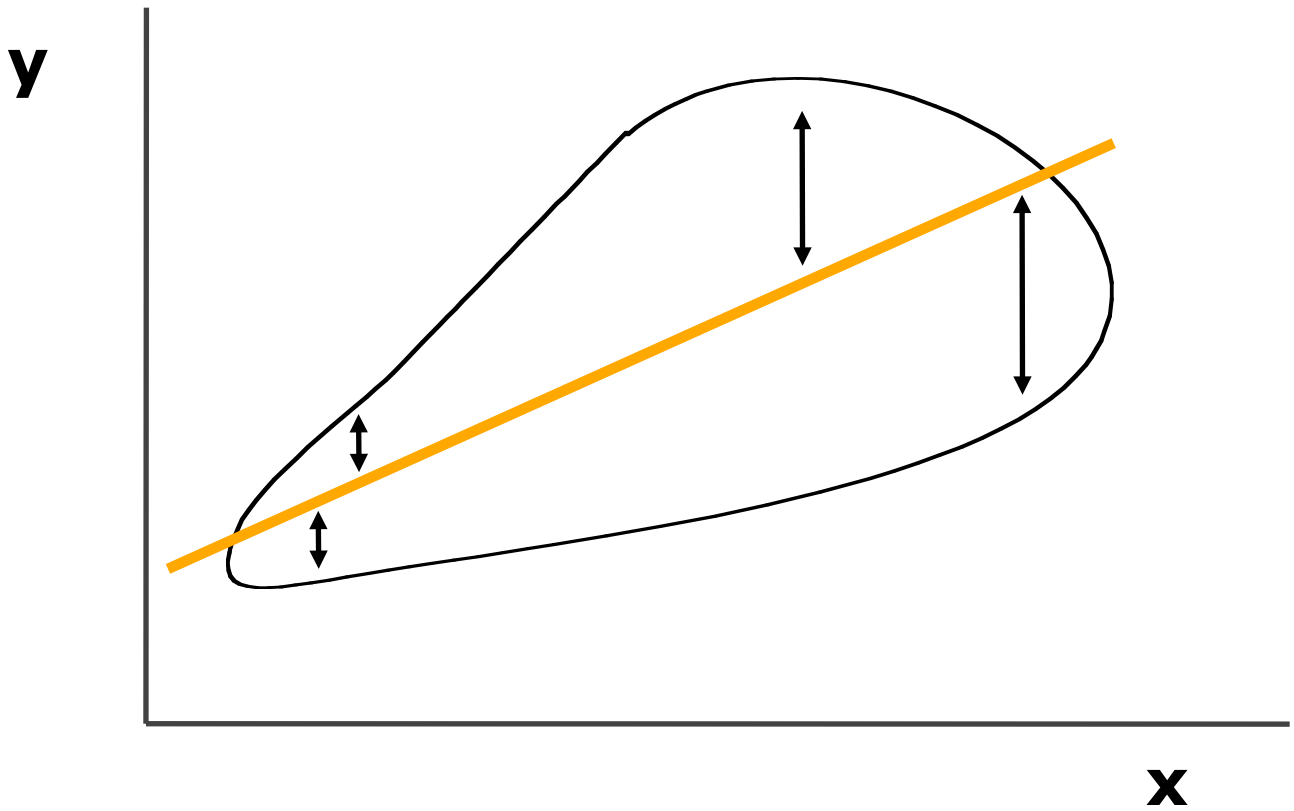
normalità bivariata

il diagramma di dispersione ha la caratteristica forma “a uovo” (sigaro, pallone da rugby).... se le distribuzioni marginali di x e y hanno la caratteristica forma “a campana” (normale, gaussiana)

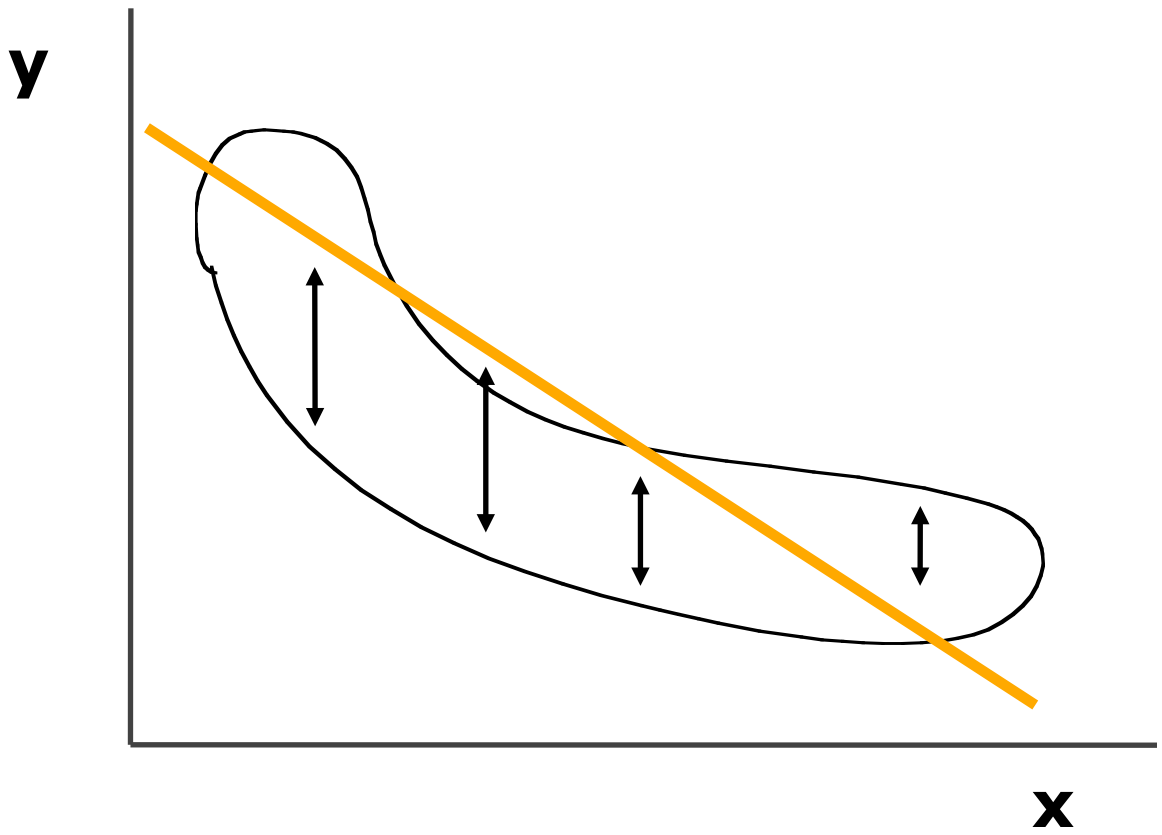
RMSE come errore di previsione

- il RMSE nella regressione misura quanto della dispersione di y **NON** è attribuibile all’effetto di x ("spiegabile" da x)
- in questo senso dovrebbe essere l'errore di previsione tipico, ma se la normalità bivariata non tiene non lo è più

eteroschedasticità



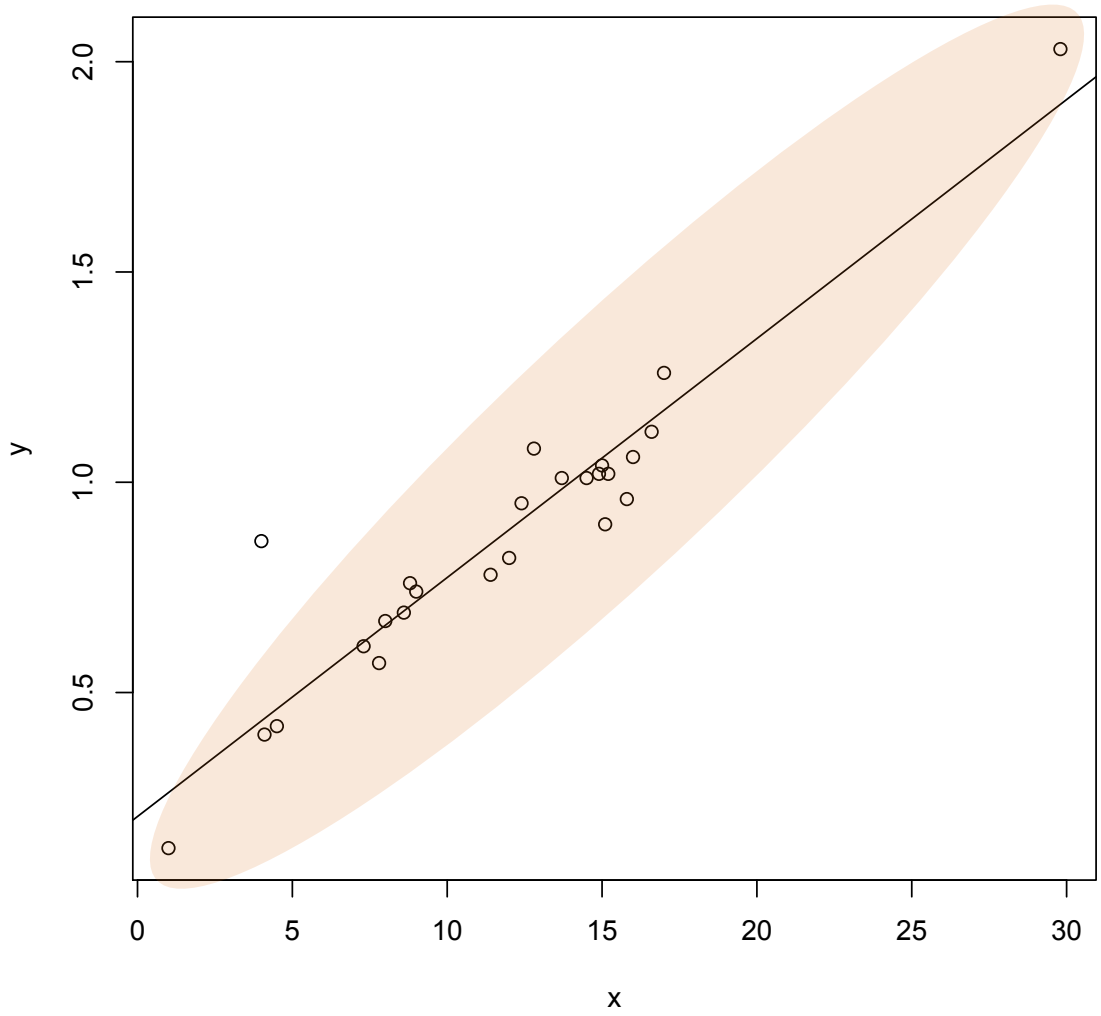
non linearità



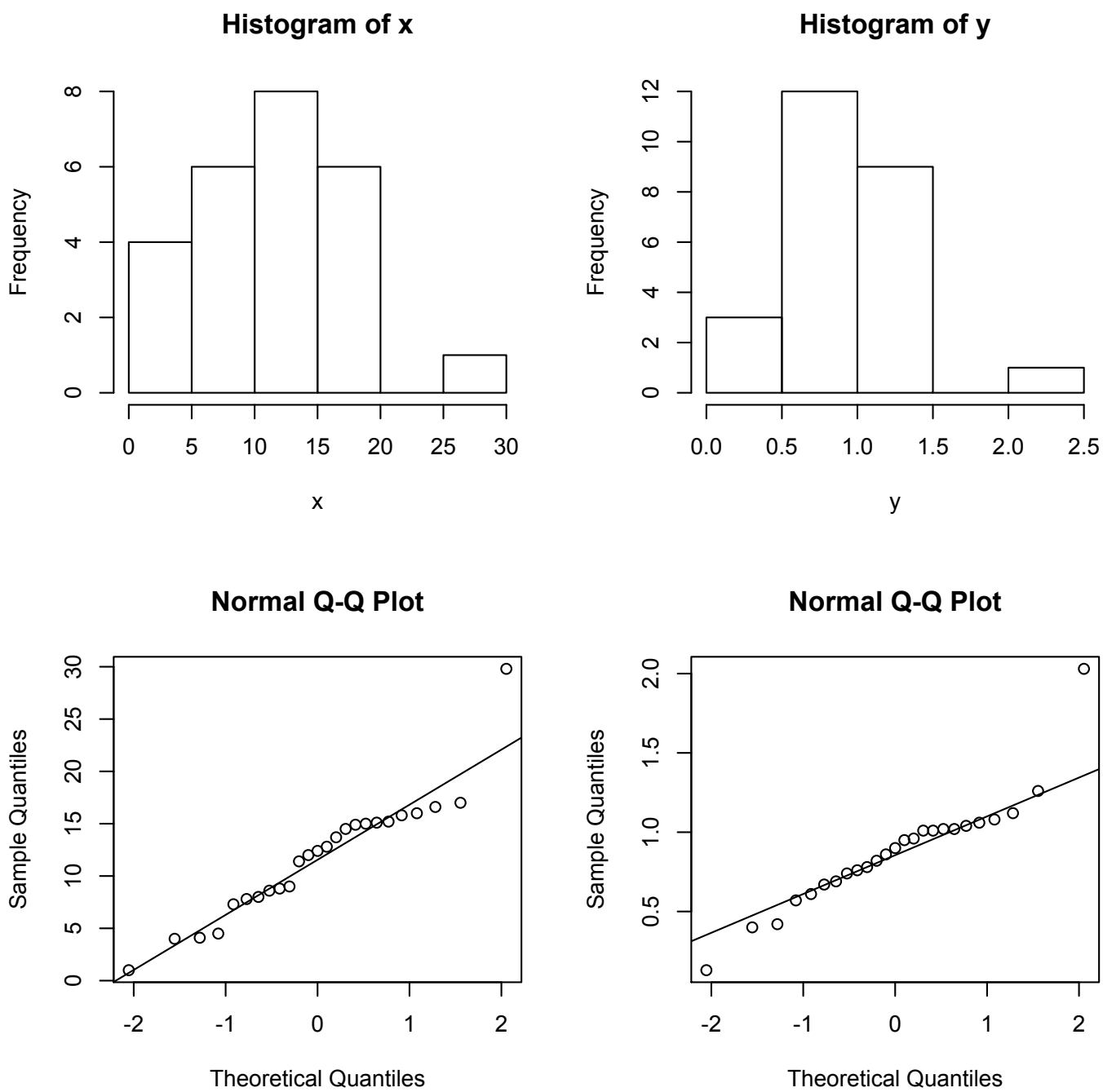
metodi di diagnosi

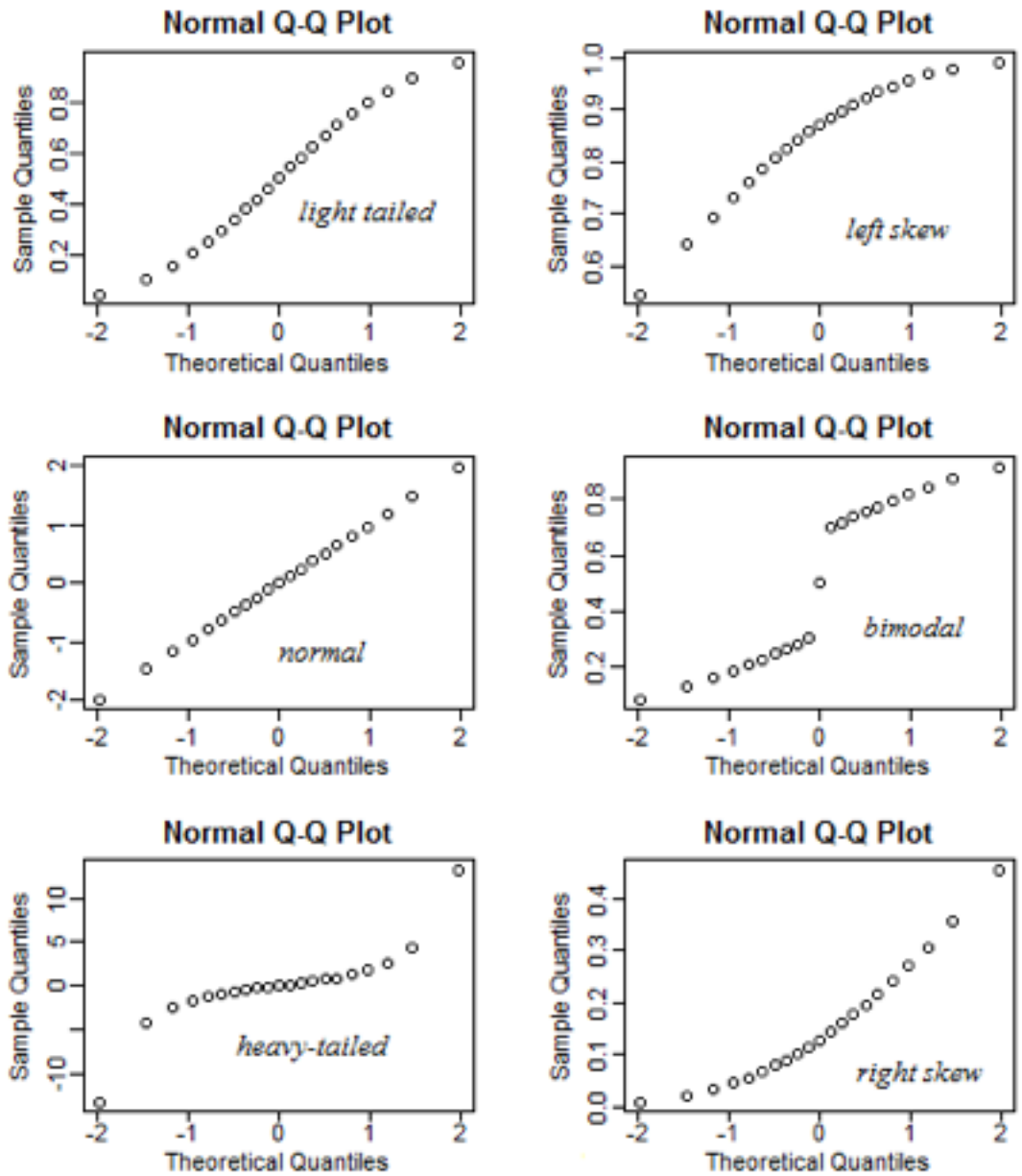
- guardare il diagramma di dispersione!
- esaminare le distribuzioni univariate di x e y
- usare qqnorm()
- esaminare la distribuzione dei residui
- altri

```
> plot(x, y)
> abline(lm(y ~ x))
```



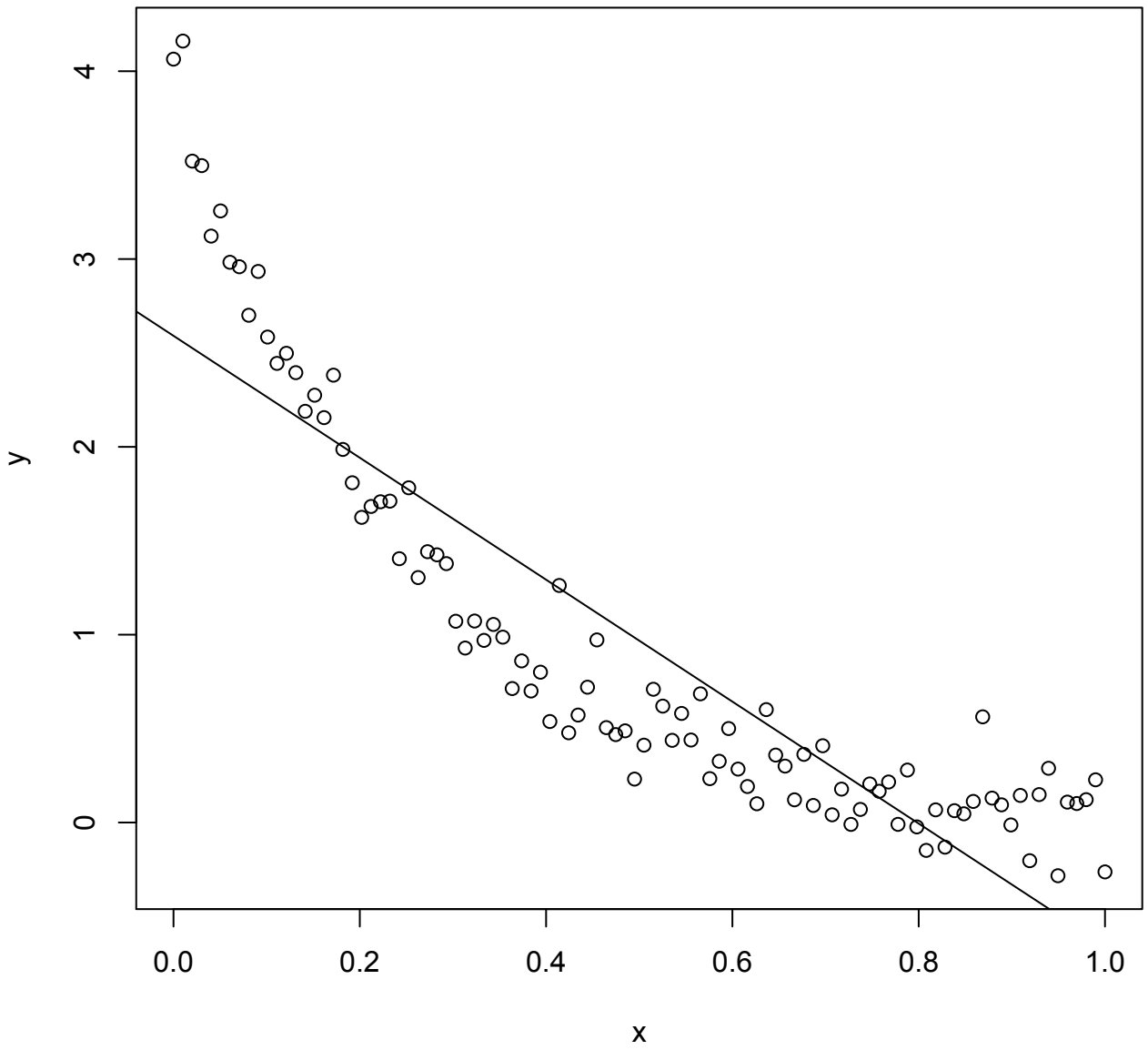
```
> op <- par(mfrow = c(2,2))
> hist(x)
> hist(y)
> qqnorm(x)
> qqline(x)
> qqnorm(y)
> qqline(y)
> par(op)
```


Histogram of x
Frequency
0
2
4
6
8
0
5
10
15
20
25
30
x
Histogram of y
Frequency
0
2
4
6
8
10
12
0.0
0.5
1.0
1.5
2.0
2.5
y
Normal Q-Q Plot
Sample Quantiles
0
5
10
15
20
25
30
-2
-1
0
1
2
Theoretical Quantiles
Normal Q-Q Plot
Sample Quantiles
0.5
1.0
1.5
2.0
-2
-1
0
1
2
Theoretical Quantiles

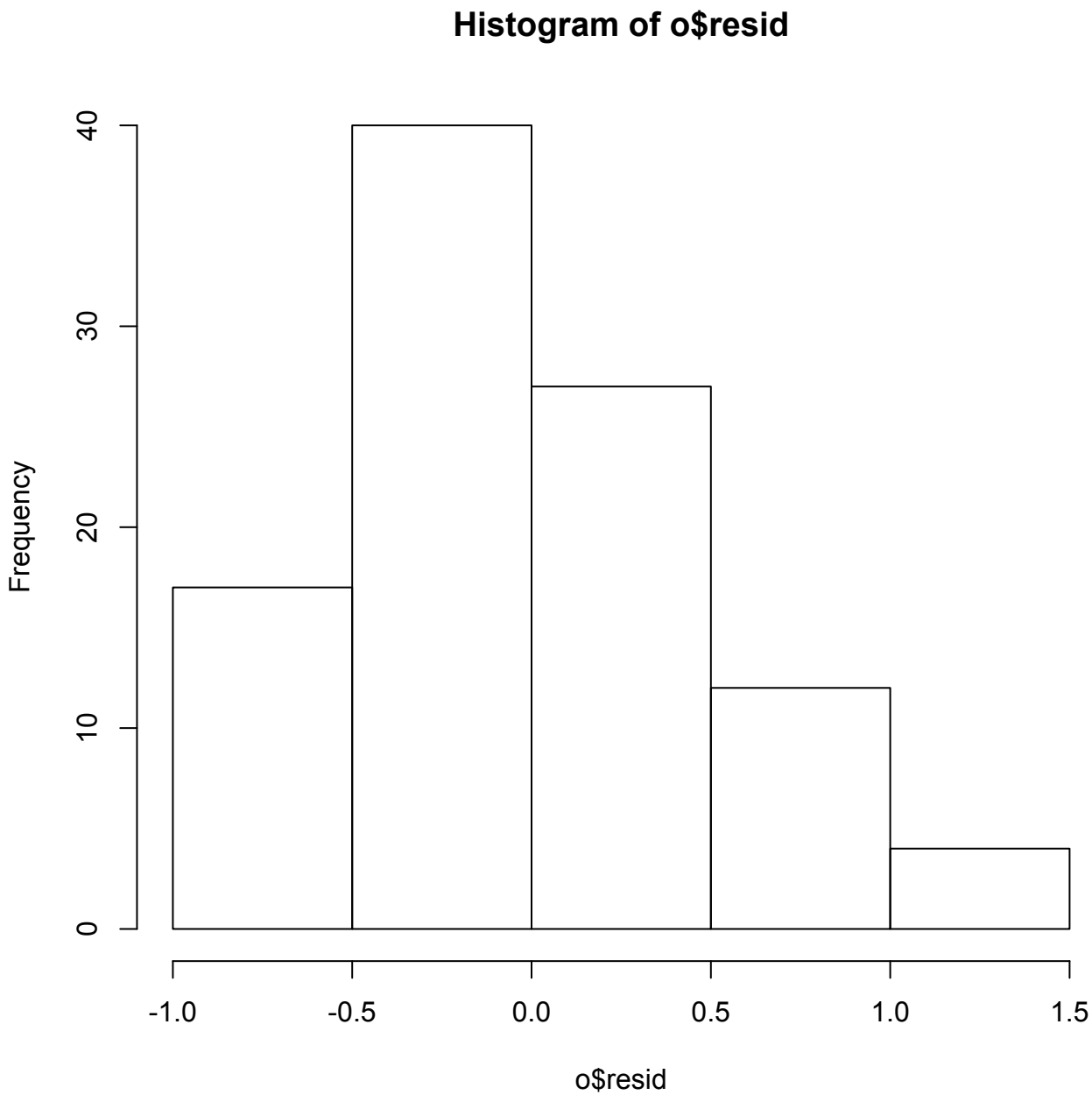

Normal Q-Q Plot
Sample Quantiles
Theoretical Quantiles
light tailed
left skew
normal
bimodal
heavy-tailed
right skew

relazione non lineare

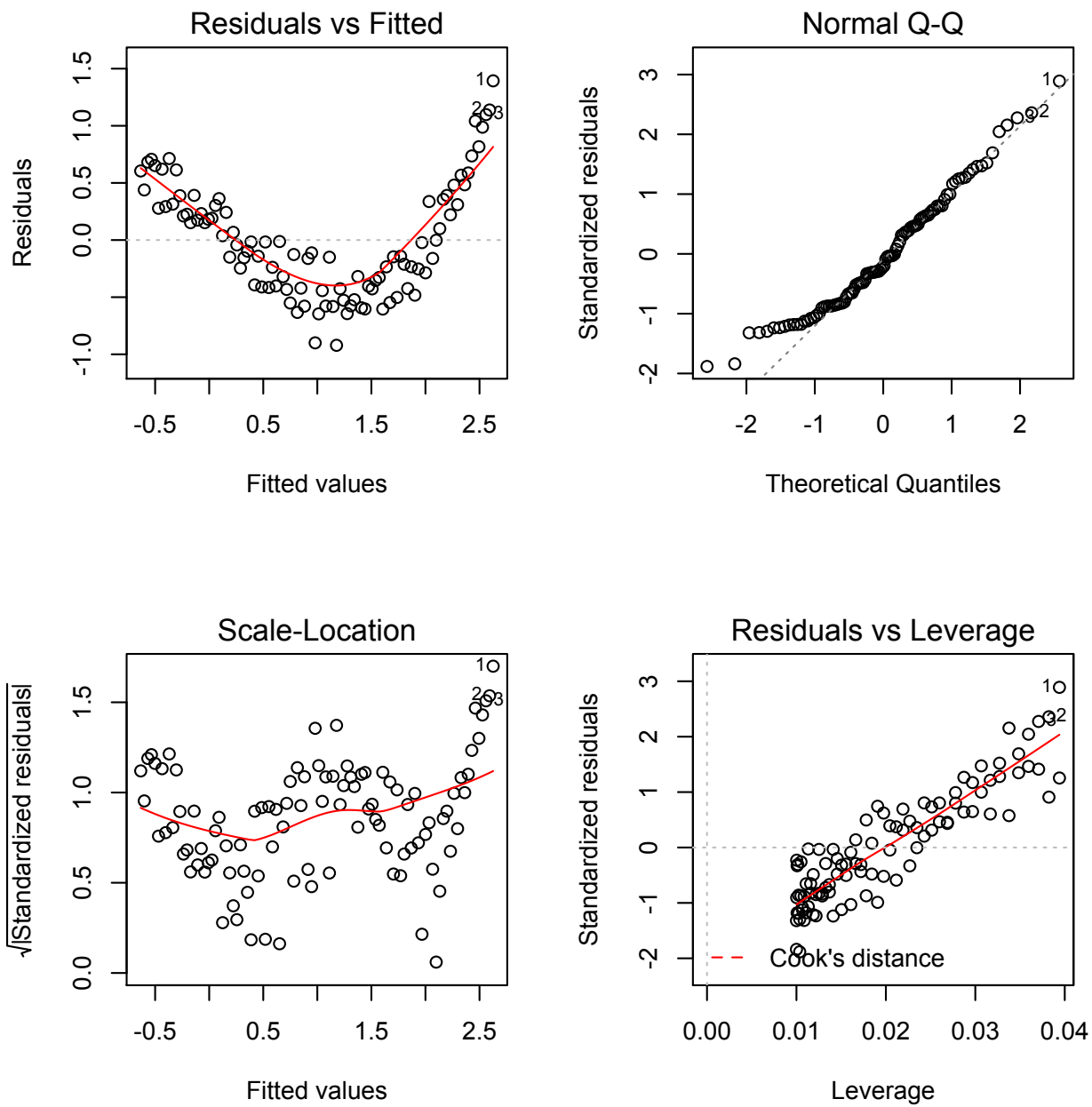
```
> x <- seq(0, 1, length = 100)
> y <- dexp(x, rate = 4)
> e <- rnorm(100)
> y <- y + e/6
> plot(y ~ x)
> abline(lm(y ~ x))
```

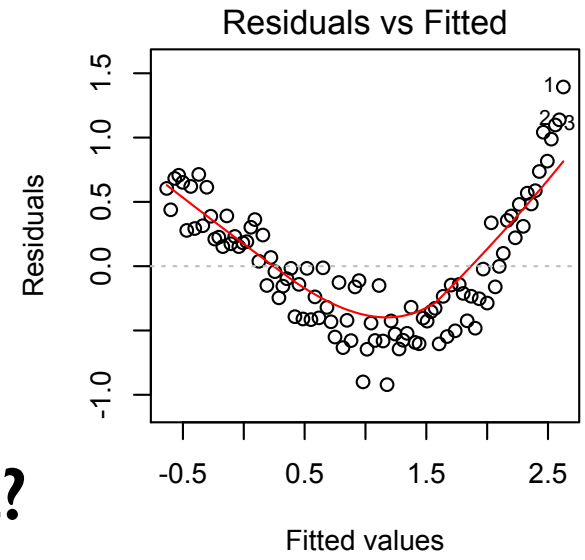

y
4
3
2
1
0
0.0
0.2
0.4
0.6
0.8
1.0
x

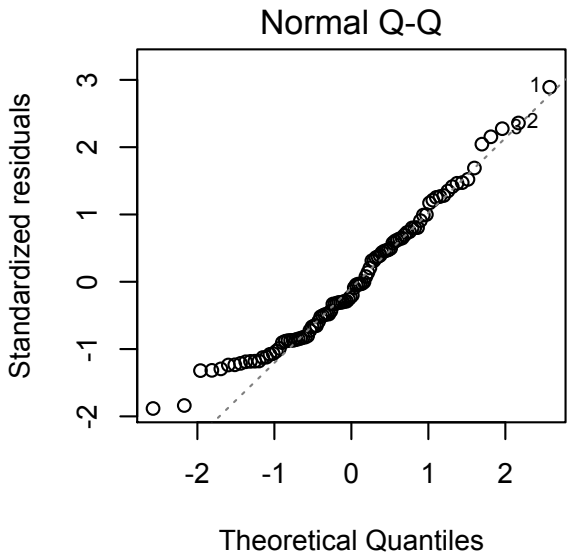
```
> o <- lm(y ~ x)
> summary(o)

Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q   Median      3Q     Max 
-0.75342 -0.37246 -0.08091  0.30232  1.60208 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.59126    0.09683   26.76   <2e-16 ***
x           -3.24615    0.16729  -19.40   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 

Residual standard error: 0.4878 on 98 degrees of freedom
Multiple R-squared: 0.7935,	Adjusted R-squared: 
0.7914 
F-statistic: 376.5 on 1 and 98 DF,  p-value: < 2.2e-16
```


Histogram of o$resid
Frequency
0
10
20
30
40
-1.0
-0.5
0.0
0.5
1.0
1.5
o$resid

```
> o <- lm(y ~ x)
> plot(o)
```

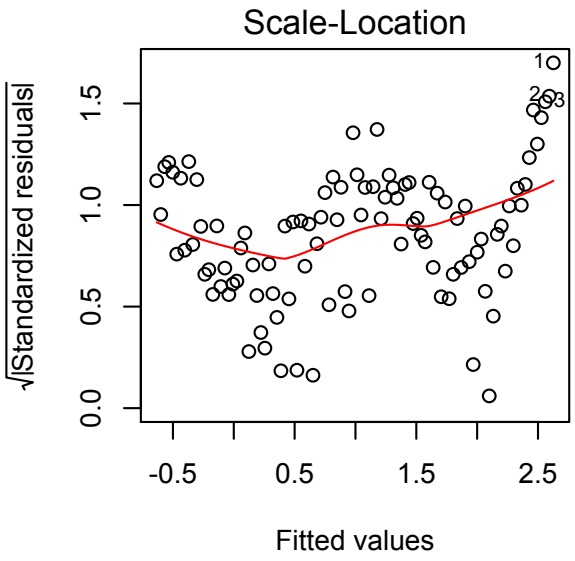

Residuals vs Fitted
Residuals
Fitted values
Normal Q-Q
Standardized residuals
Theoretical Quantiles
Scale-Location
√|Standardized residuals|
Fitted values
Residuals vs Leverage
Standardized residuals
Cook's distance
Leverage

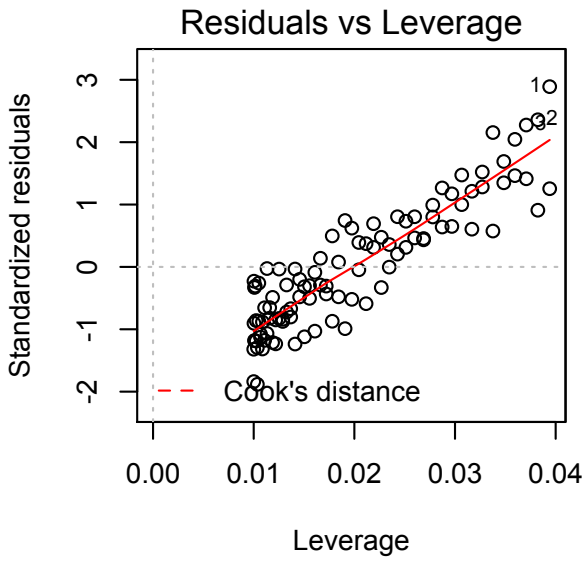




residui normali?

linearità?

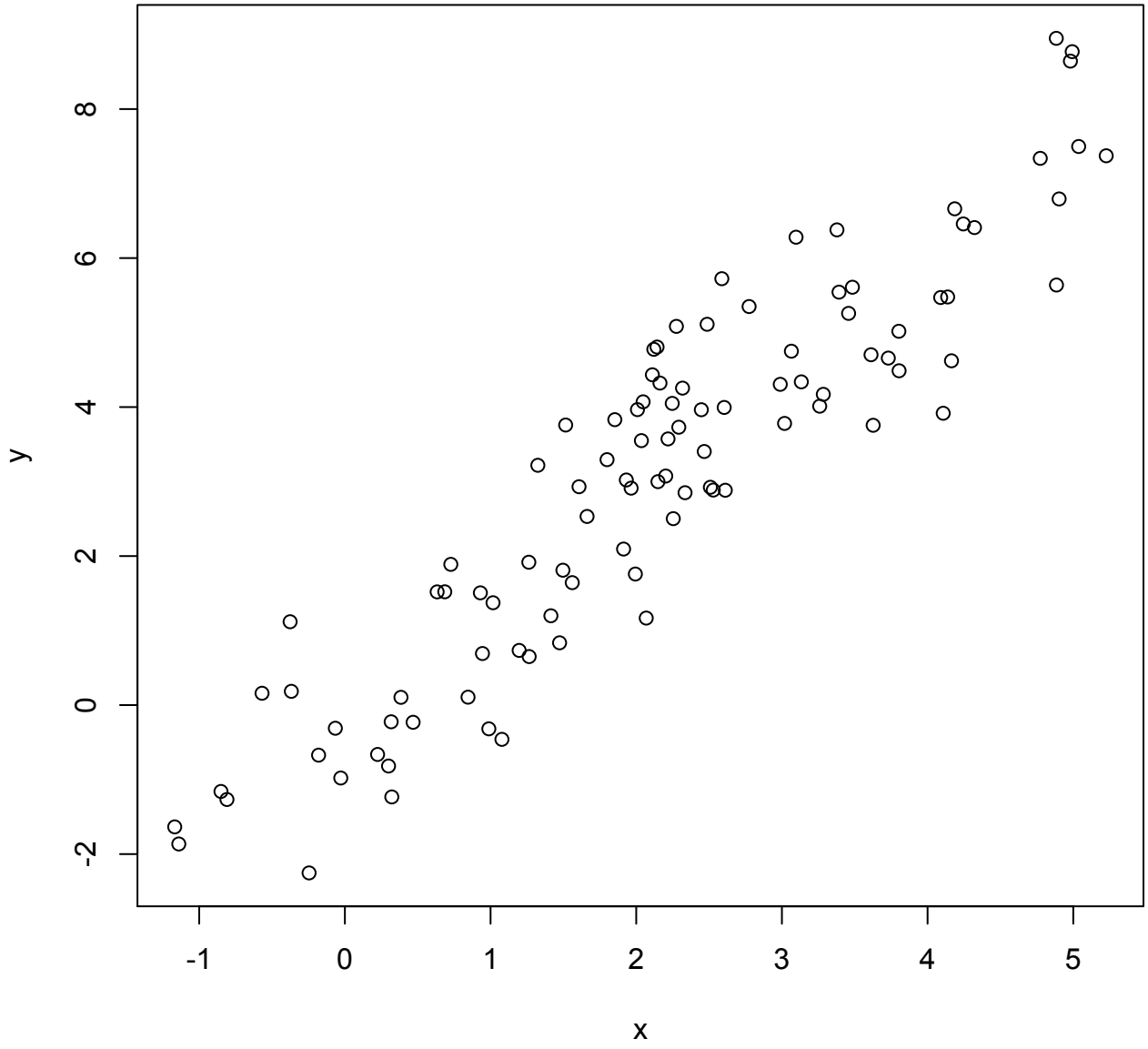




outlier?

omoschedasticità??

regressione senza intercetta



b?

a?

```
> ex <- rnorm(100)
> ey <- rnorm(100)
> b <- 1.5
> x <- runif(100, 0 , 4)
> x <- x + ex
> y <- x * b + ey
> plot(x, y)
```

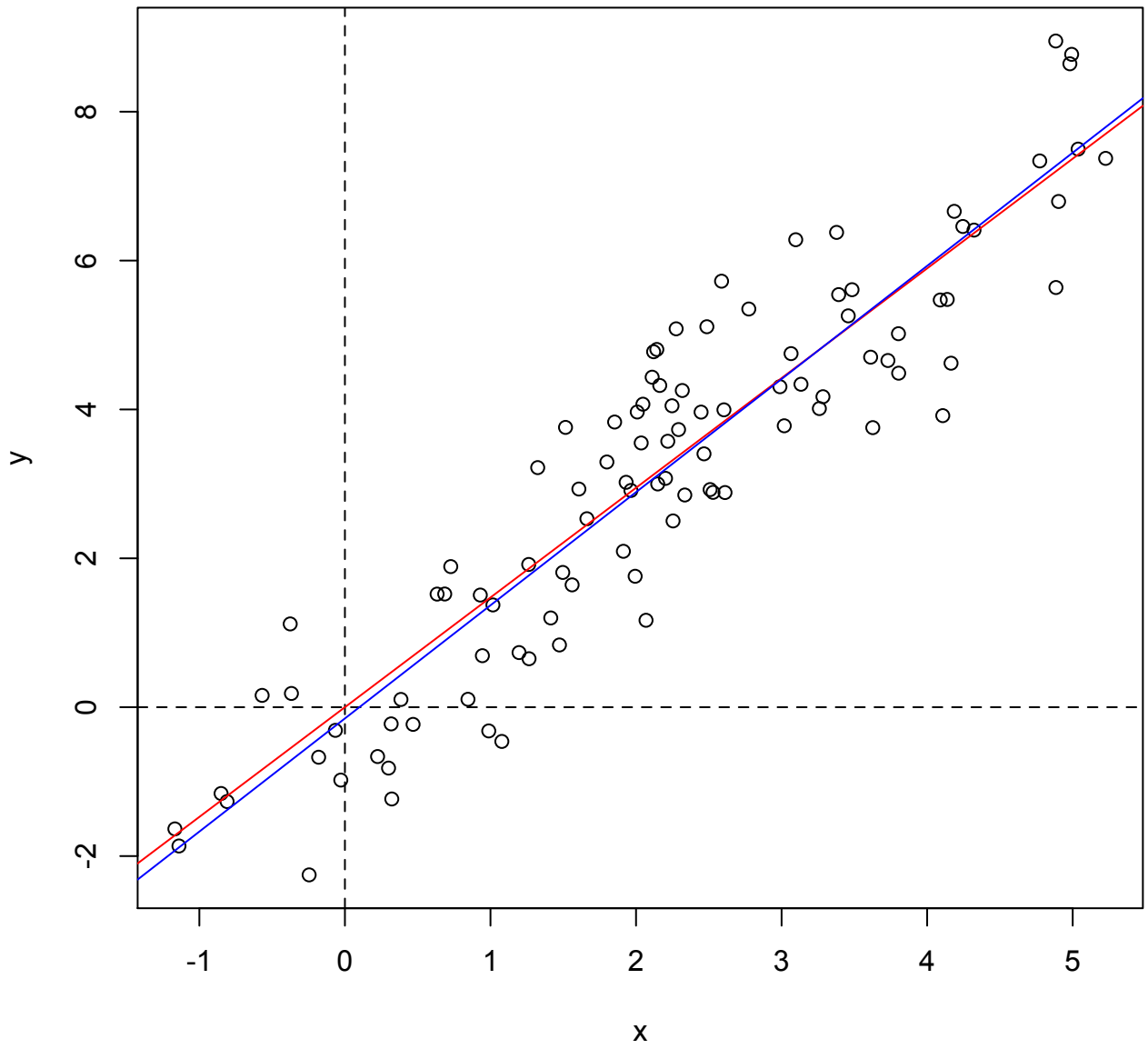
b?

a?

```
> mod1 <- lm(y ~ x)
> mod1

Call:
lm(formula = y ~ x)

Coefficients:
(Intercept)            x
     -0.151        1.520
```

```
> mod2 <- lm(y ~ x - 1)
> mod2

Call:
lm(formula = y ~ x - 1)

Coefficients:
    x
1.474
```

```
> plot(x, y)
> abline(v = 0, lty = "dashed")
> abline(h = 0, lty = "dashed")
> abline(mod2, col = "red")
> abline(mod1, col = "blue")
```

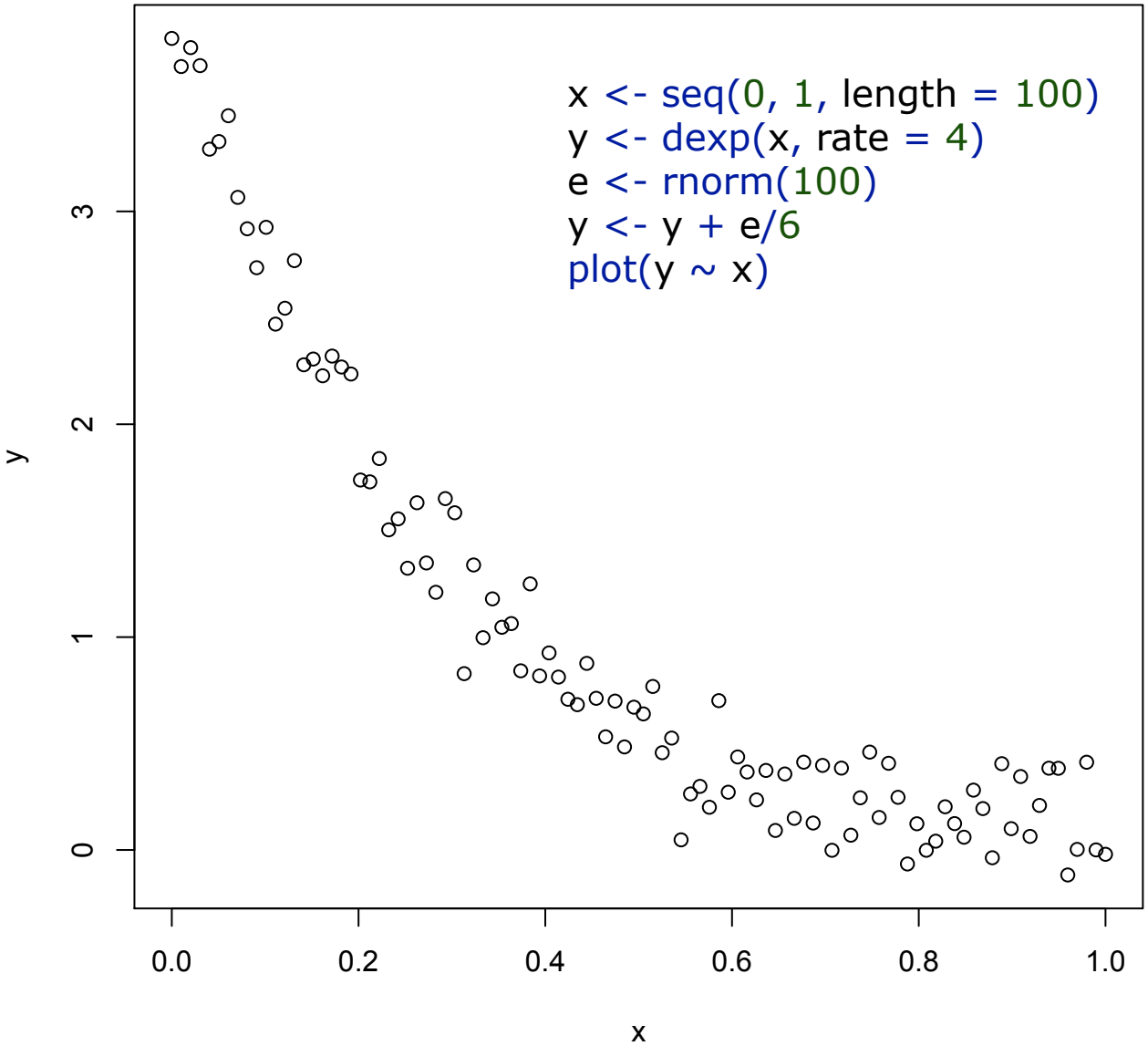

y
8
6
4
2
0
-2
-1
0
1
2
3
4
5
x

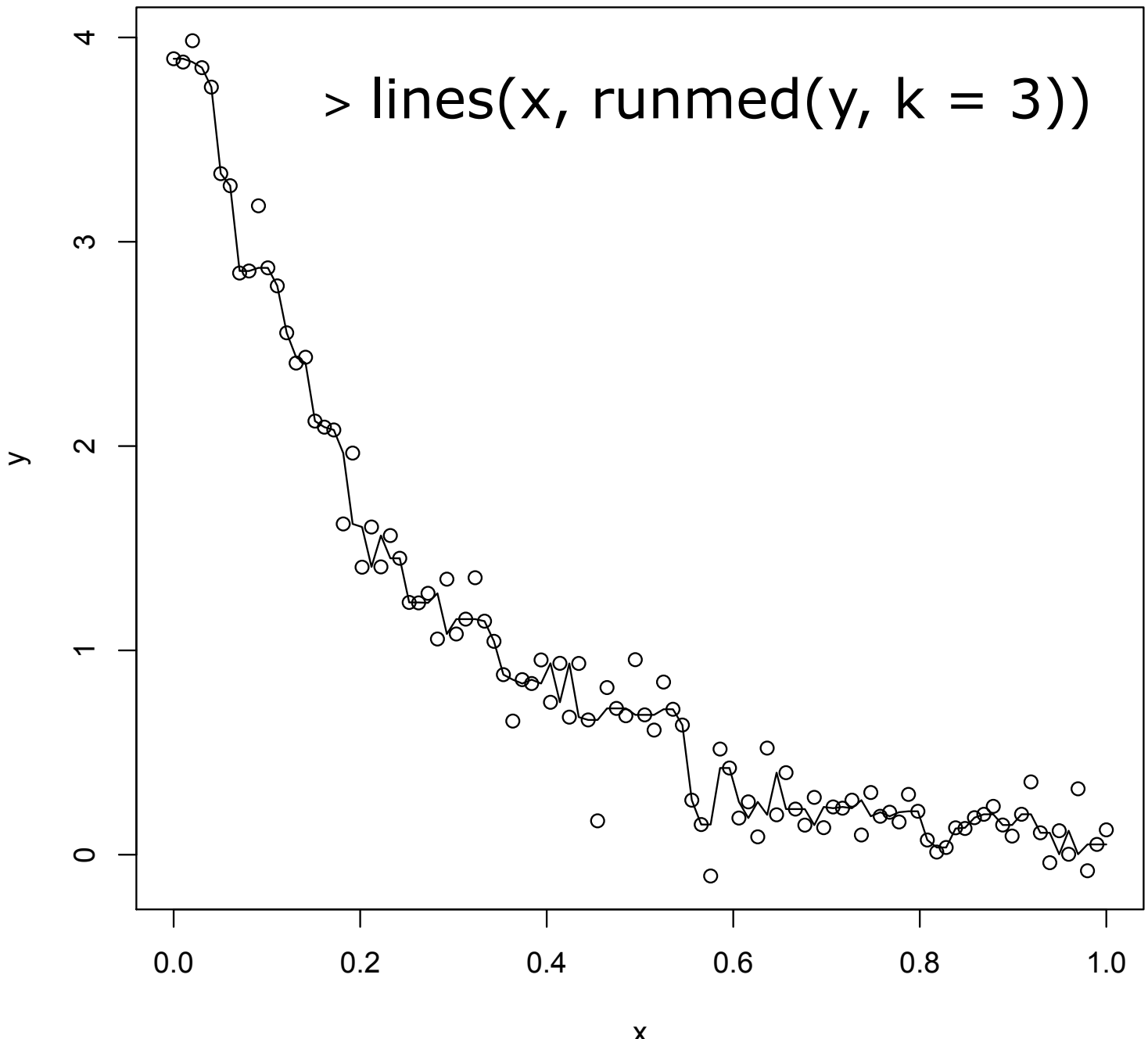
```
> summary(mod1)

Call:
lm(formula = y ~ x)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.17505 -0.74749 -0.00829  0.70837  1.94102 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) -0.15103    0.17117  -0.882     0.38    
x            1.51997    0.06468  23.500   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.004 on 98 degrees of freedom
Multiple R-squared: 0.8493,	Adjusted R-squared: 0.8478 
F-statistic: 552.3 on 1 and 98 DF,  p-value: < 2.2e-16
```

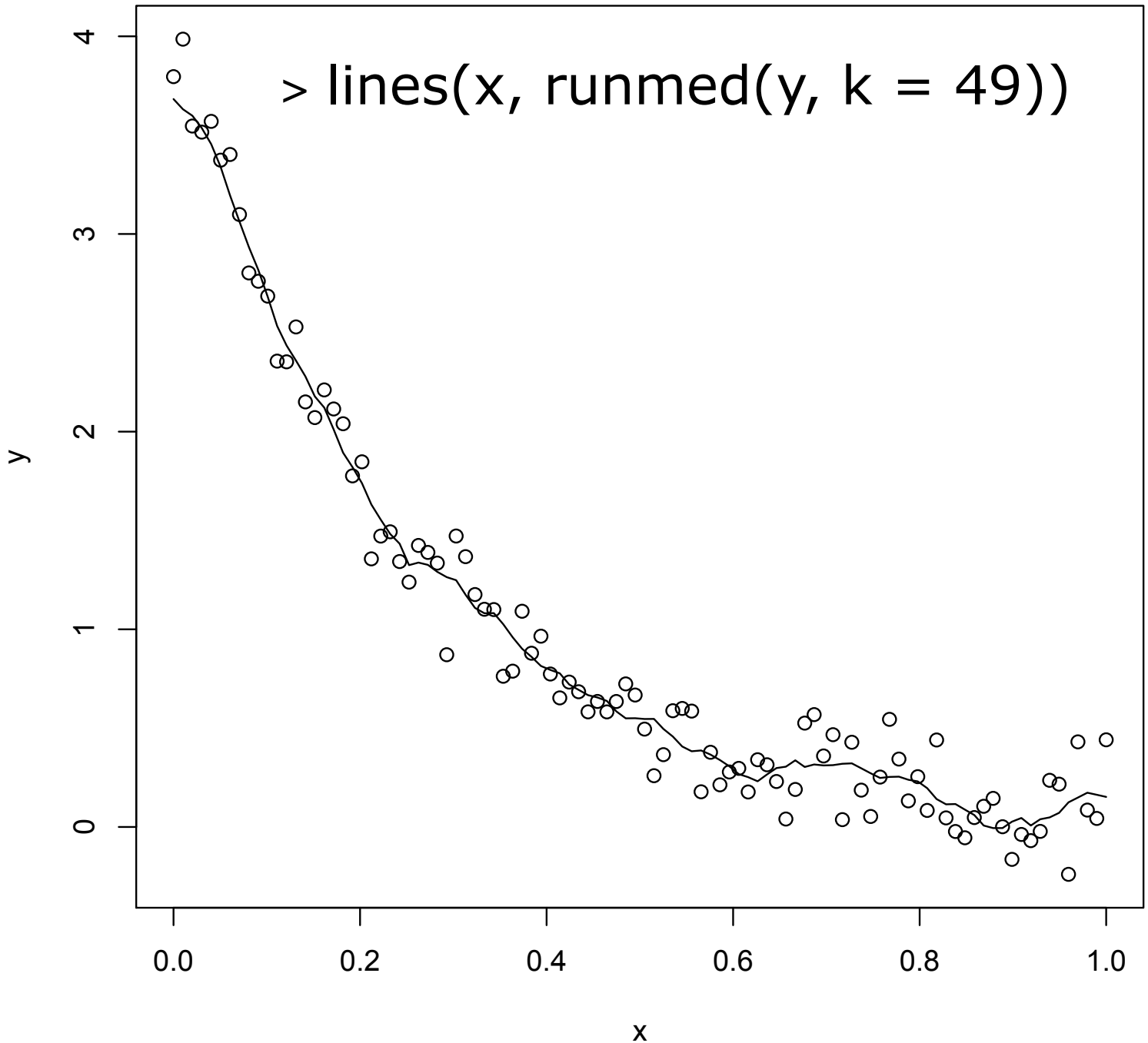
```
> summary(mod2)

Call:
lm(formula = y ~ x - 1)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.13622 -0.77844 -0.08673  0.60009  1.90960 

Coefficients:
  Estimate Std. Error t value Pr(>|t|)    
x  1.47375    0.03789    38.9   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.003 on 99 degrees of freedom
Multiple R-squared: 0.9386,	Adjusted R-squared: 0.938 
F-statistic:  1513 on 1 and 99 DF,  p-value: < 2.2e-16
```

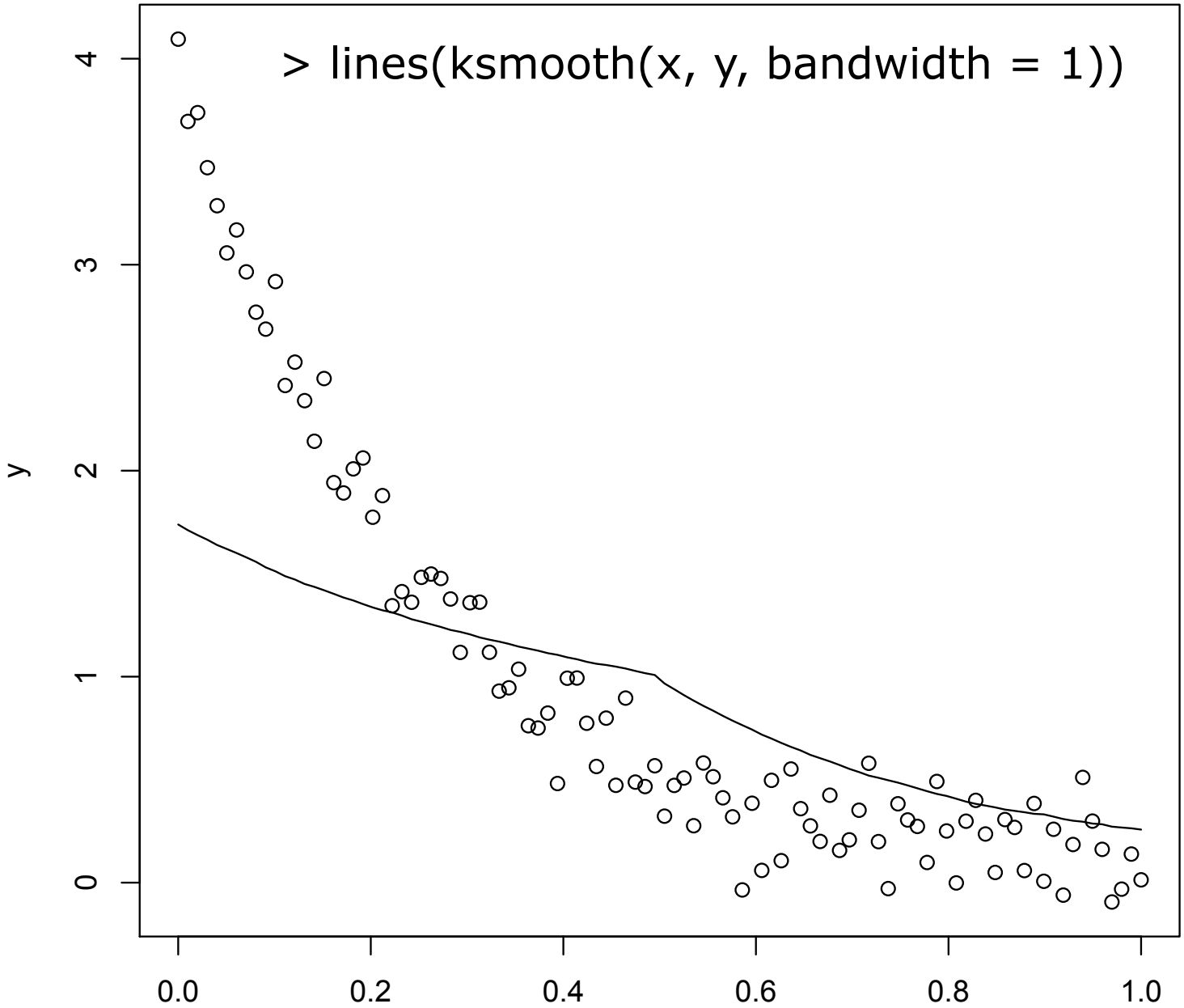
```
> anova(mod2, mod1)

Analysis of Variance Table

Model 1: y ~ x - 1
Model 2: y ~ x
  Res.Df  RSS    Df   Sum of Sq  F       Pr(>F)
1     99 99.525
2     98 98.741  1   0.78436  0.7785 0.3798
```

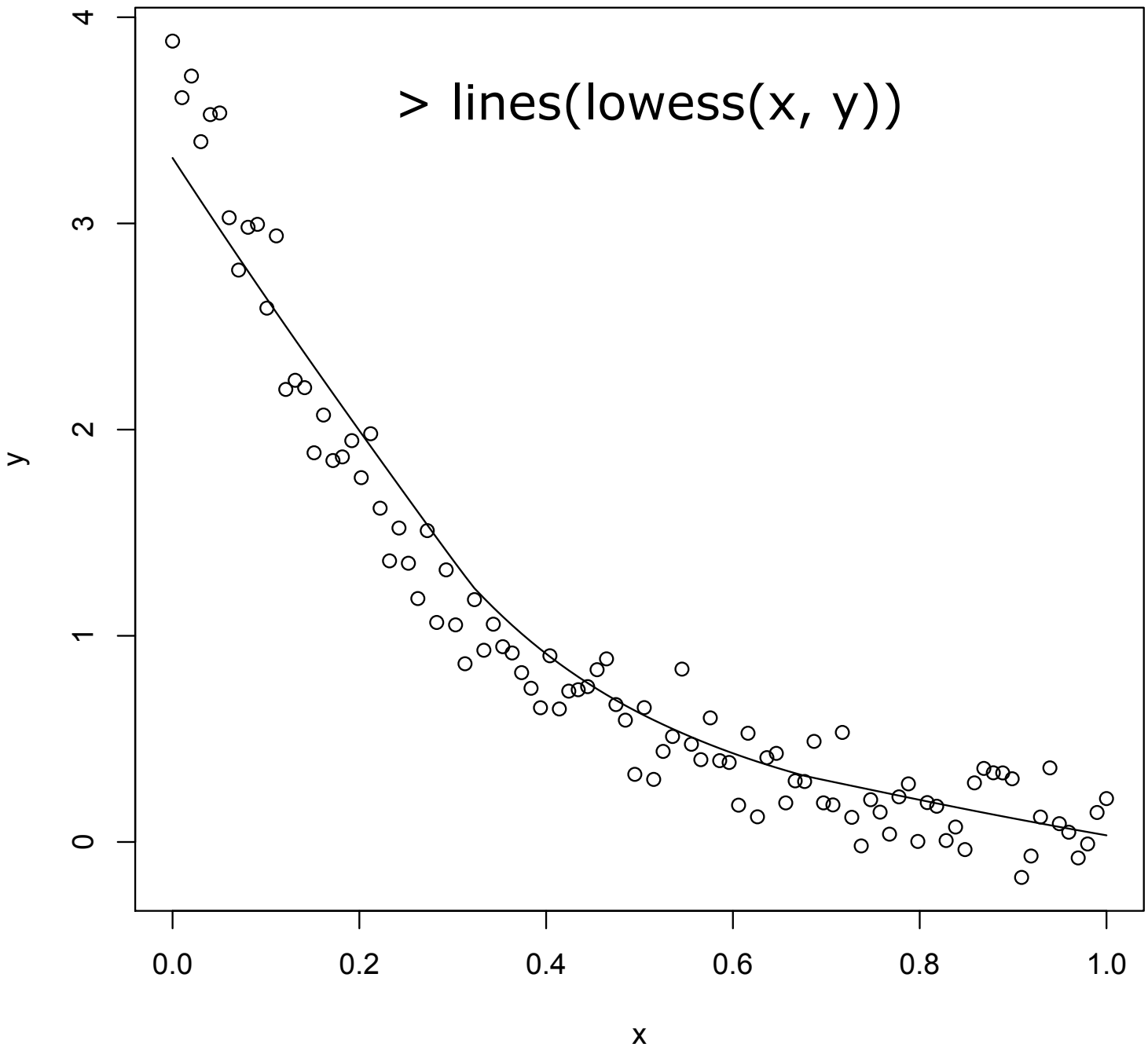
smoothers

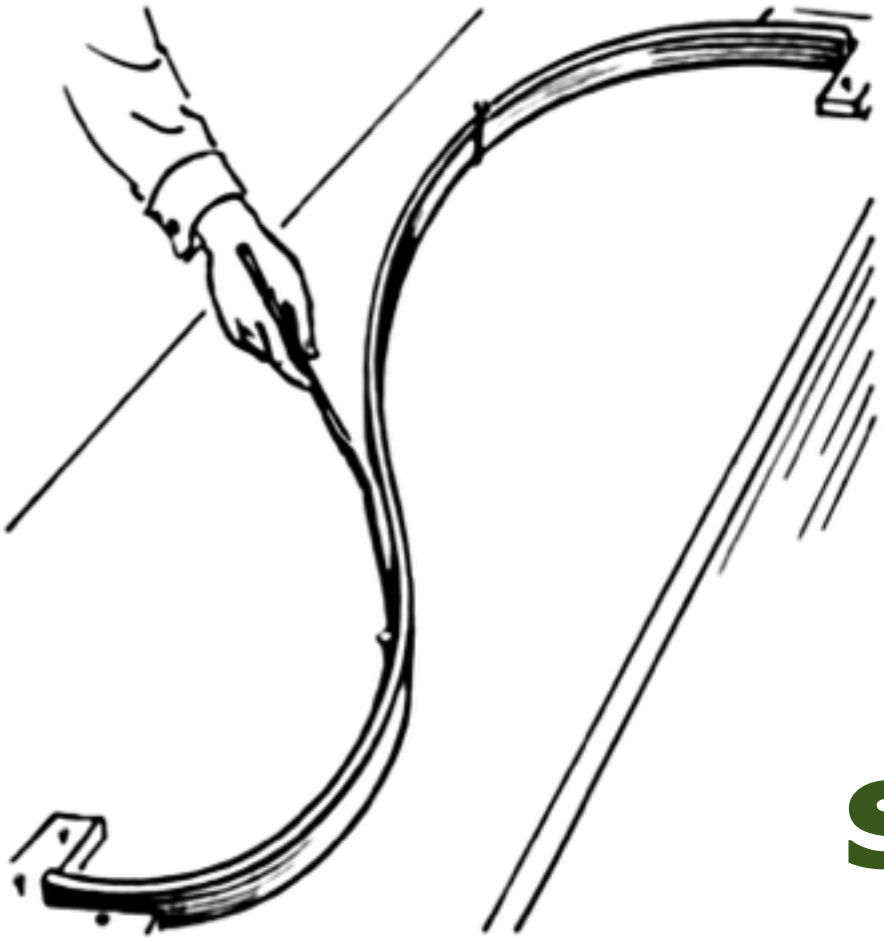

x <- seq(0, 1, length = 100)
y <- dexp(x, rate = 4)
e <- rnorm(100)
y <- y + e/6
plot(y ~ x)
y
0
1
2
3
0.0
0.2
0.4
0.6
0.8
1.0
x


> lines(x, runmed(y, k = 3))
y
x
0
1
2
3
4
0.0
0.2
0.4
0.6
0.8
1.0

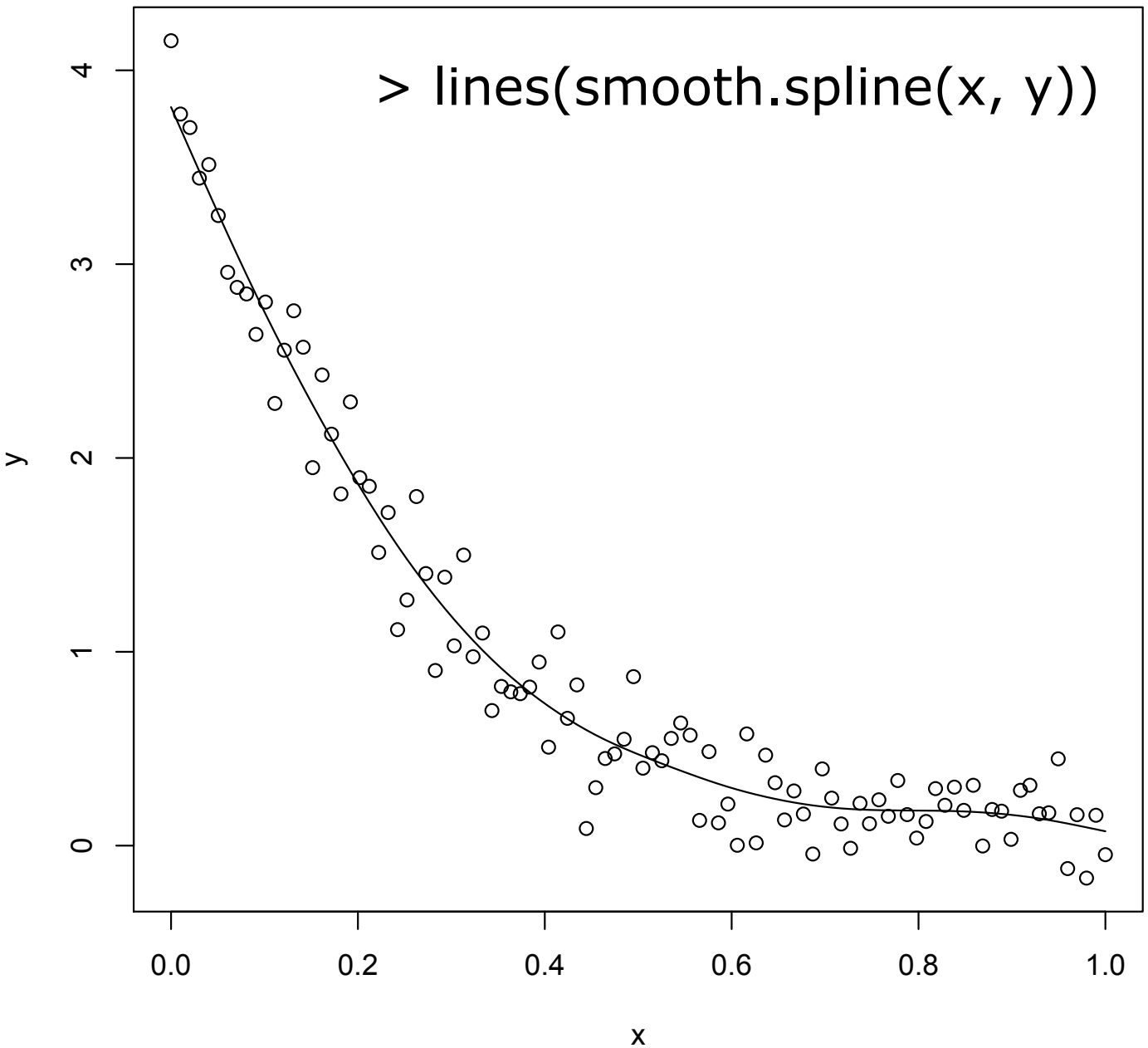

> lines(x, runmed(y, k = 49))
y
x
0
1
2
3
4
0.0
0.2
0.4
0.6
0.8
1.0


> lines(ksmooth(x, y, bandwidth = 1))
y
0
1
2
3
4
0.0
0.2
0.4
0.6
0.8
1.0


> lines(lowess(x, y))
y
x
0
1
2
3
4
0.0
0.2
0.4
0.6
0.8
1.0



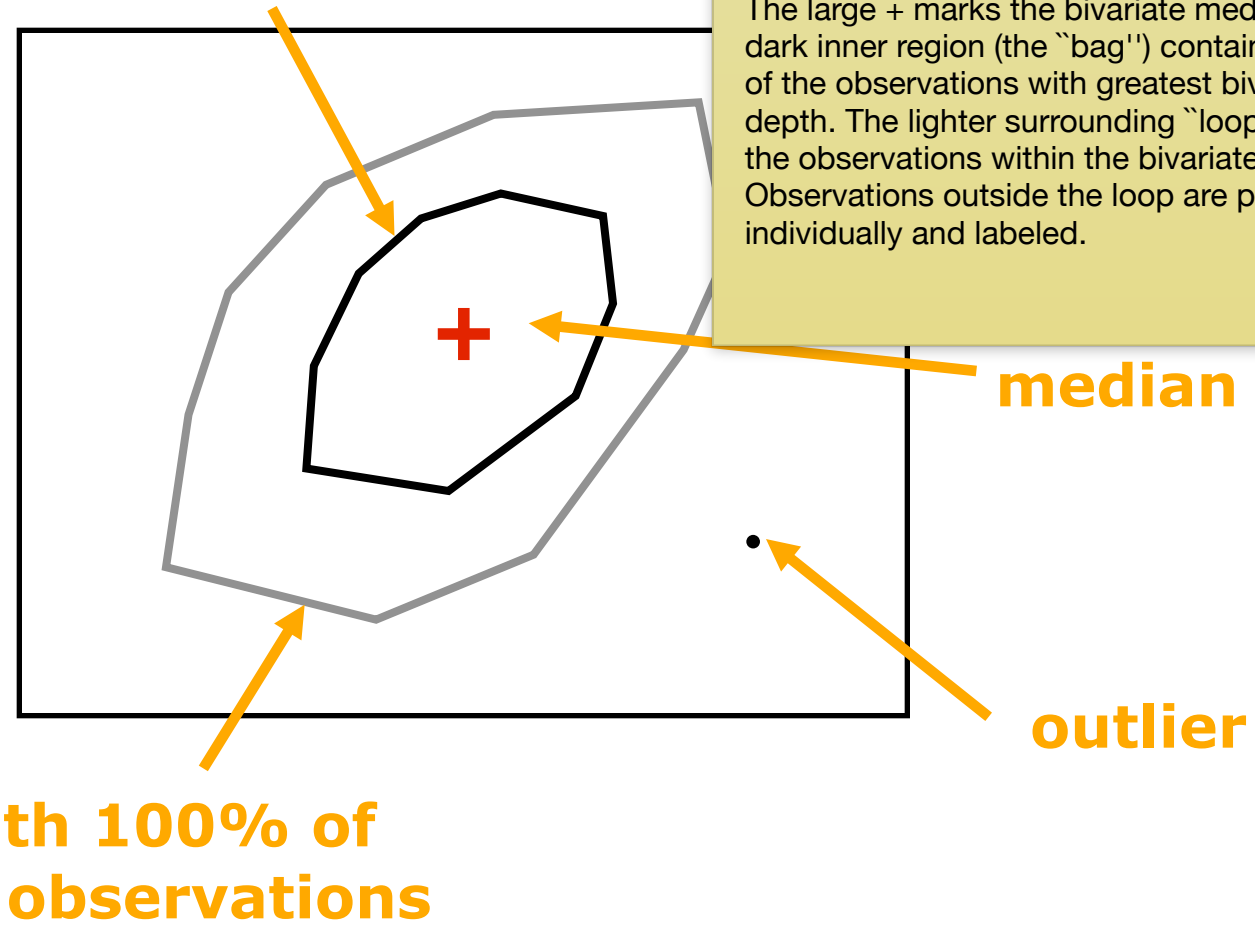
spline

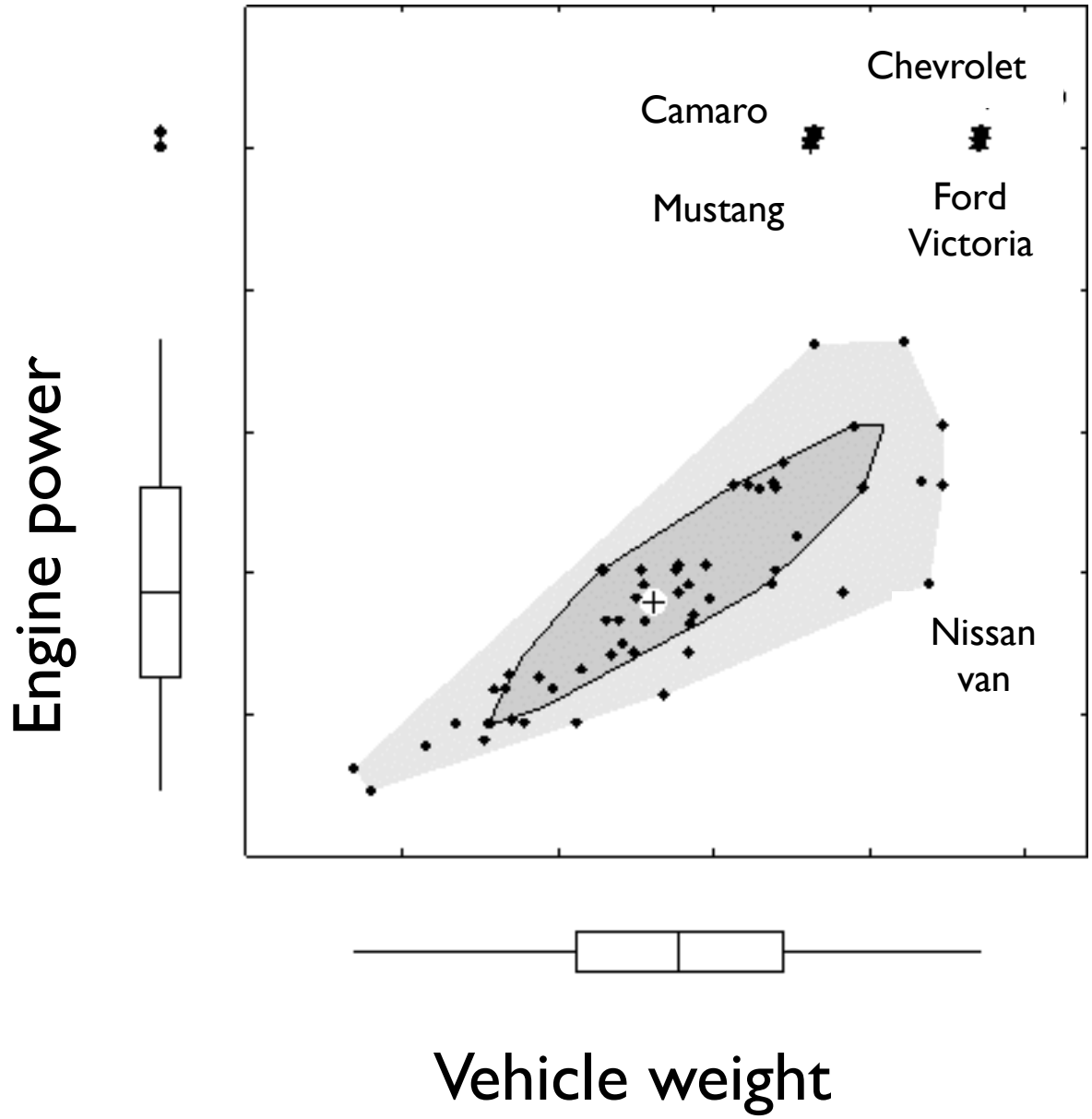

> lines(smooth.spline(x, y))
y
x
0
1
2
3
4
0.0
0.2
0.4
0.6
0.8
1.0

bag-plot

bag-plot

The univariate boxplot has been widely used since proposed by Tukey around 1971. Tukey (1975) also suggested a multivariate generalization of depth of an observation on which the boxplot is based, but no implementation of this idea had been available until quite recently. Peter Rousseeuw, Ida Ruts and John Tukeyy worked out the bivariate extension, called a bagplot, illustrated here. The large + marks the bivariate median. The dark inner region (the ``bag'') contains the 50% of the observations with greatest bivariate depth. The lighter surrounding ``loop'' marks the observations within the bivariate fences. Observations outside the loop are plotted individually and labeled.

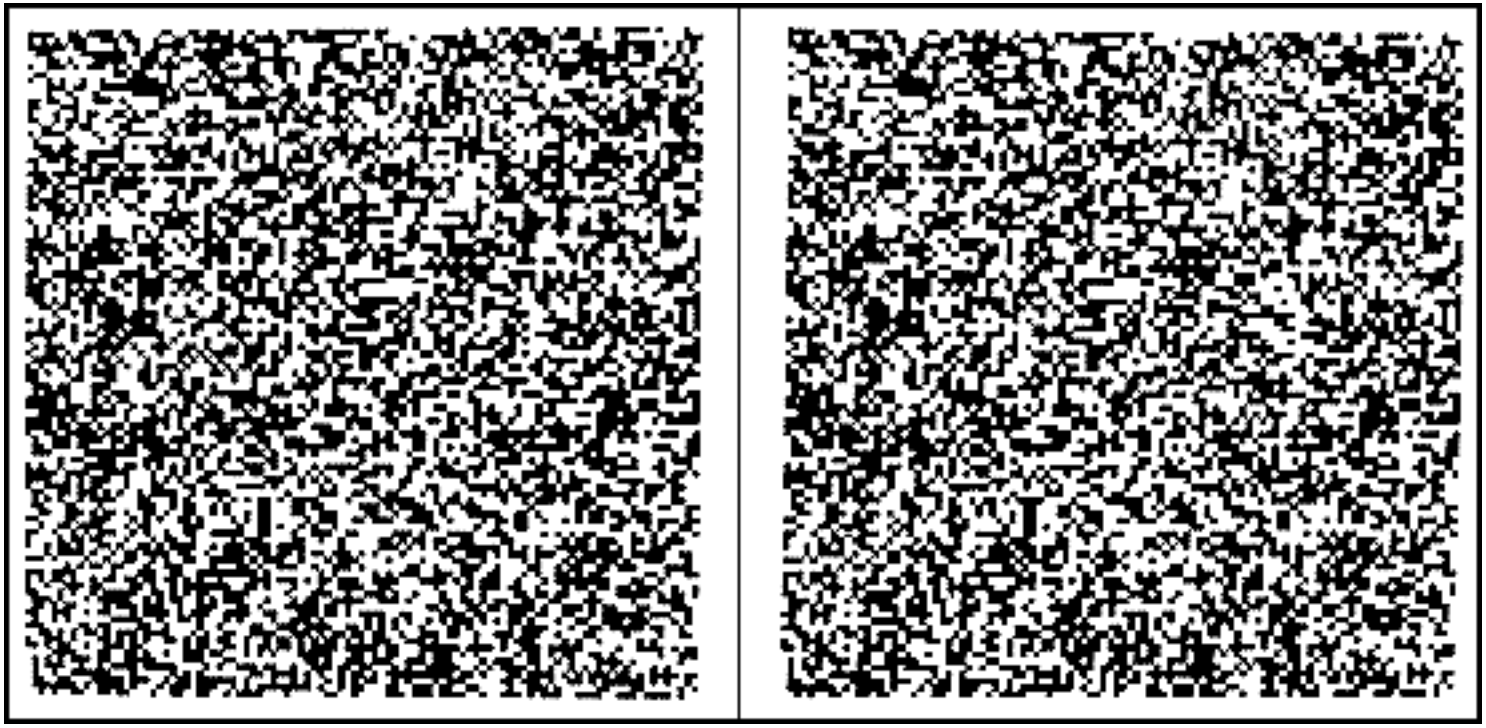




from Rousseuw et al (1999) *Am Statistician*, Fig. 1

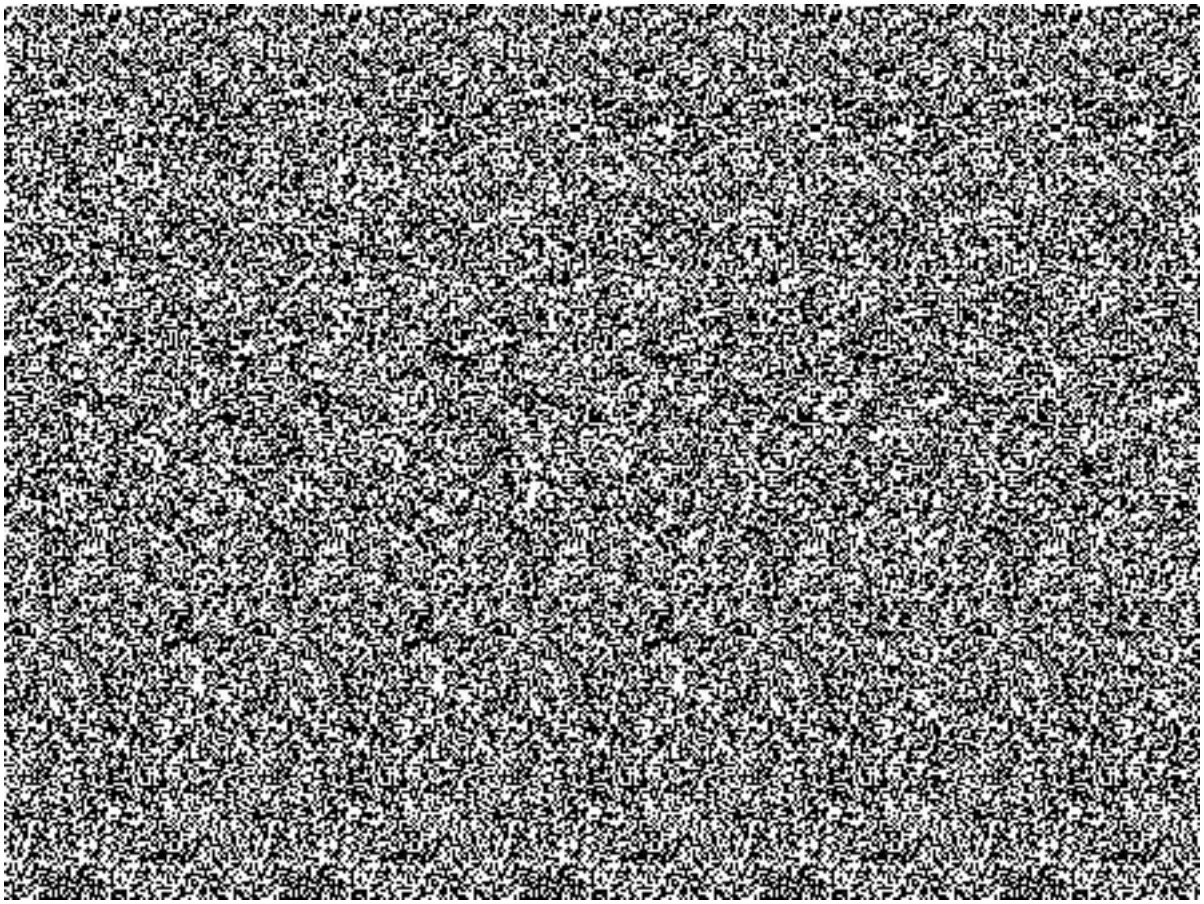
associazione fra una variabile categoriale e una numerica

due categorie

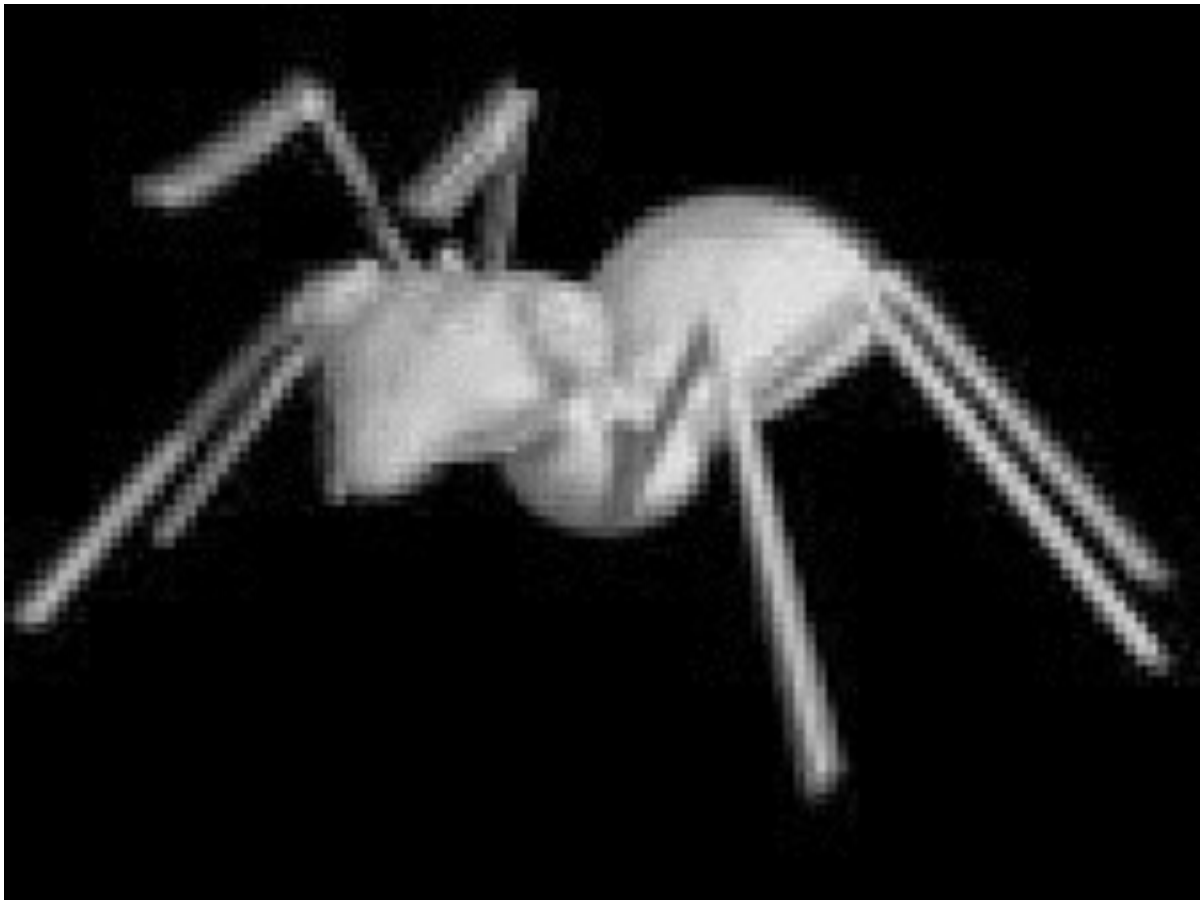


stereogramma di punti casuali
(Random Dot Stereogram)

```
> df <- read.table("~/Desktop/R/dispense/
esempi_corso/RDS.txt", header = 1)

> str(df)
'data.frame': 78 obs. of  2 variables:
 $ Ts    : num  47.2 22 20.4 19.7 17.4 ...
 $ group: Factor w/ 2 levels "NV","VV": 1 1
1 1 1 1 1 1 1 1 ...
```

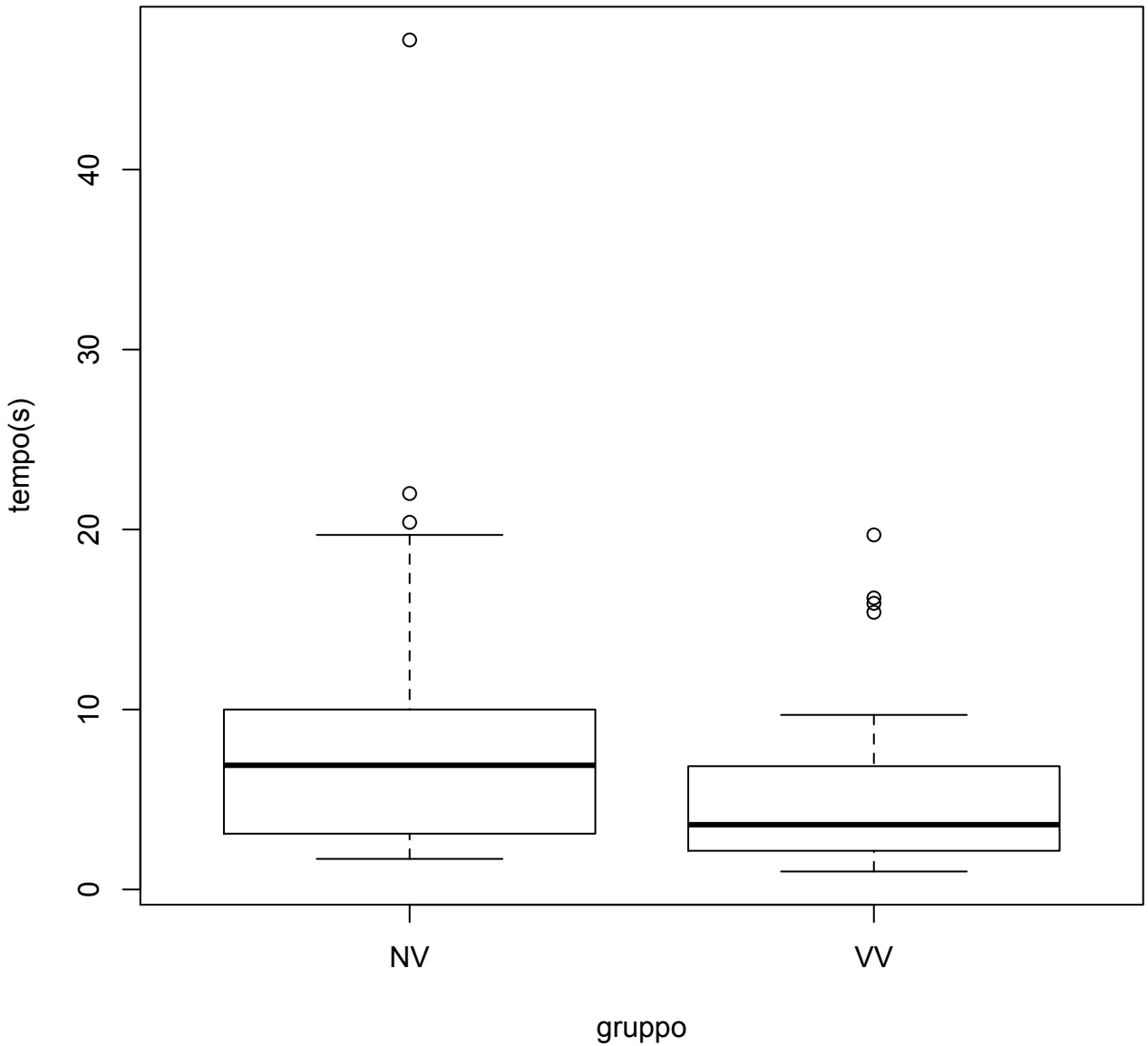


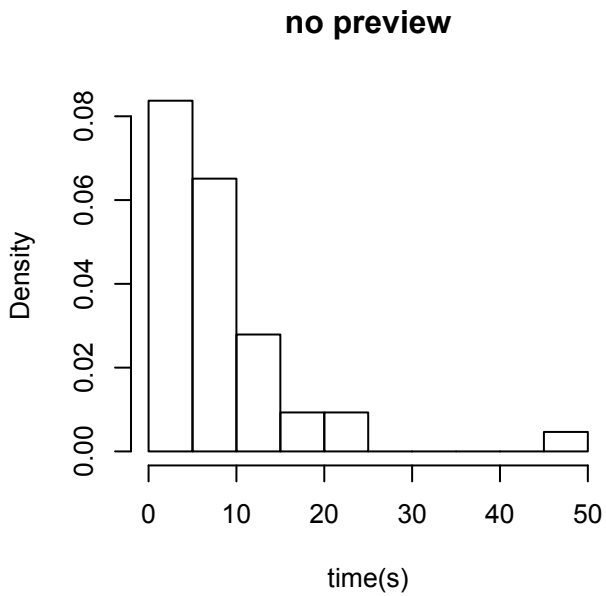
NV: nessuna informazione Visiva o Verbale

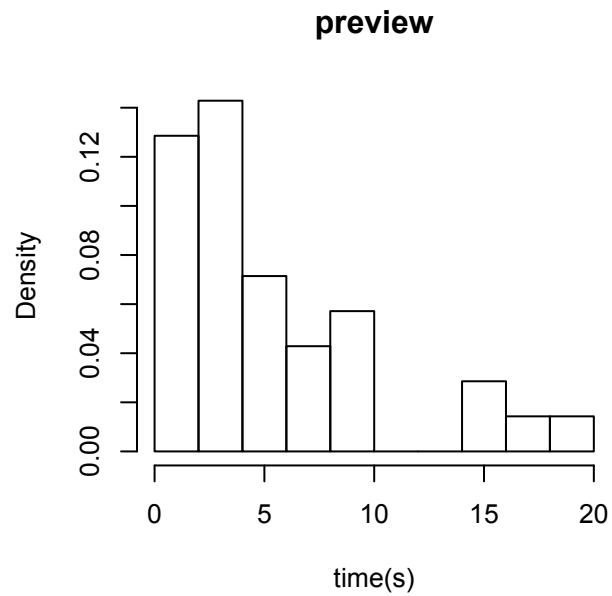


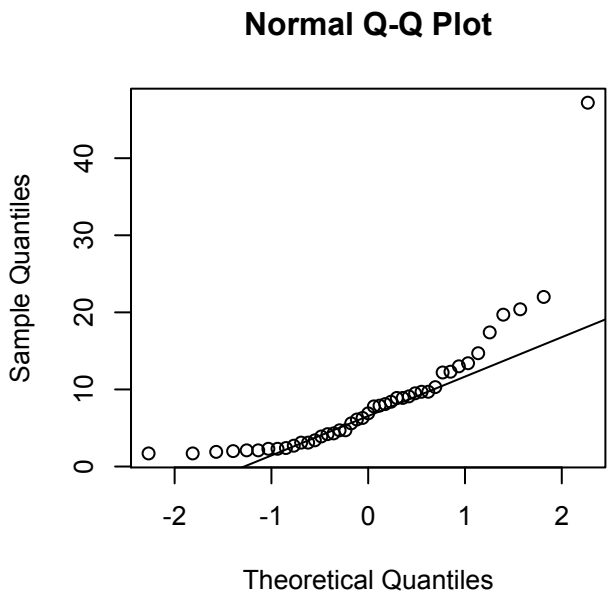
V: immagine sopra + “formica”

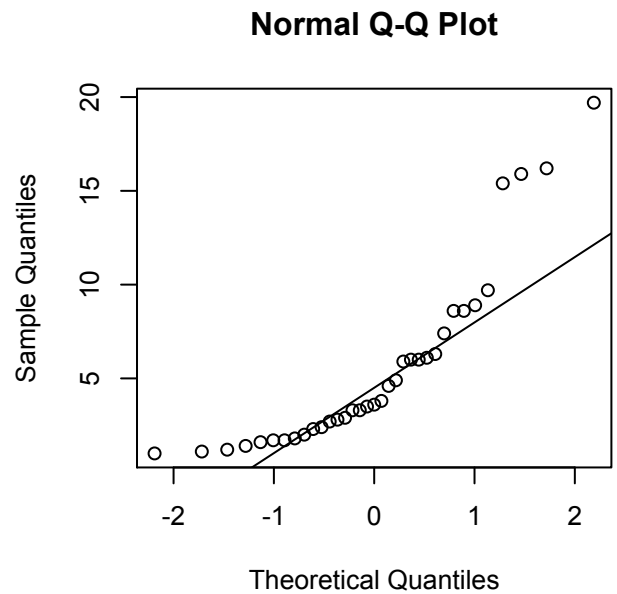
```
> boxplot(RDS$Ts ~ RDS$group, xlab = "gruppo",
ylab = "tempo(s)")
```



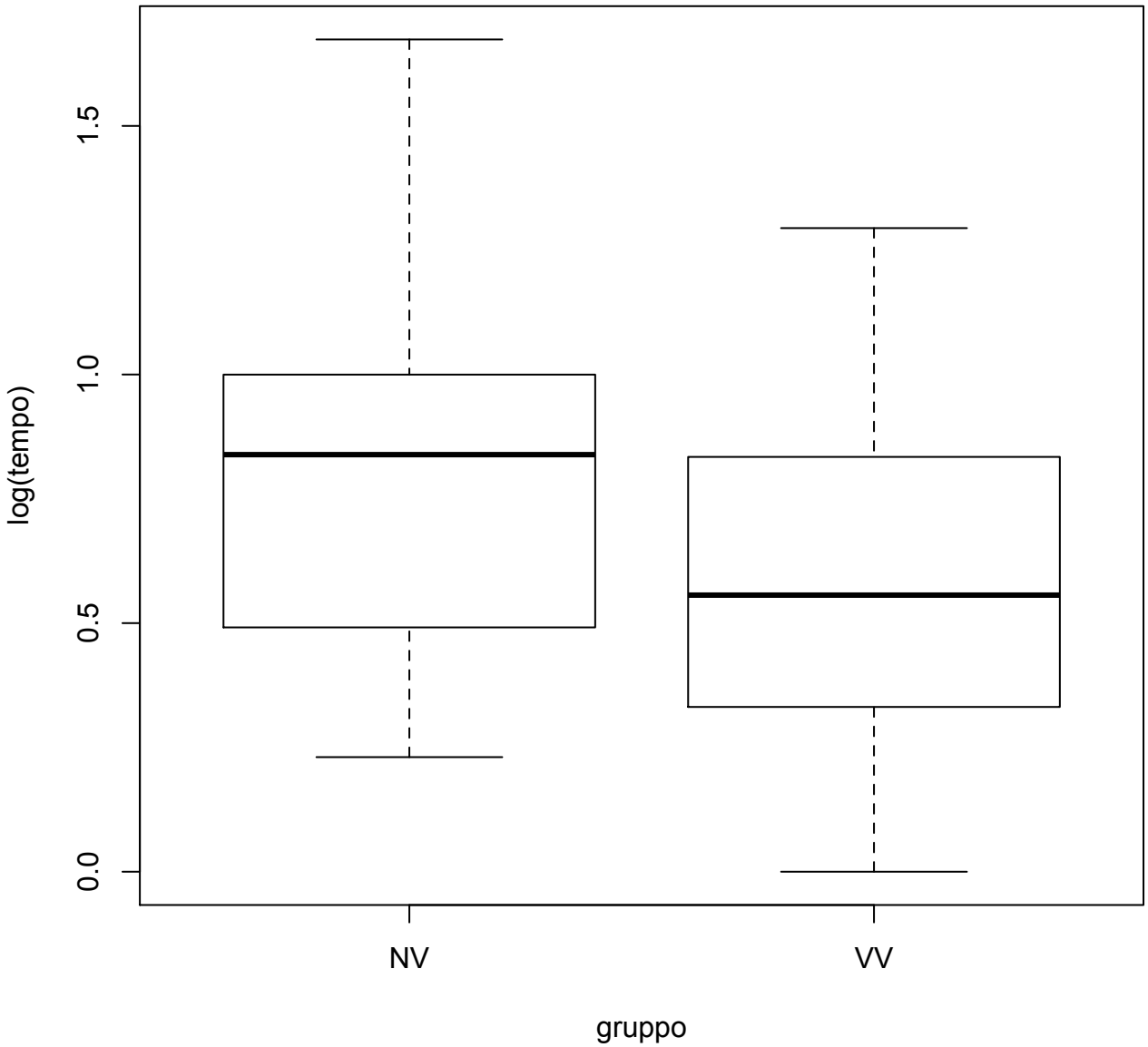

no preview
Density
0.00
0.02
0.04
0.06
0.08
0
10
20
30
40
50
time(s)

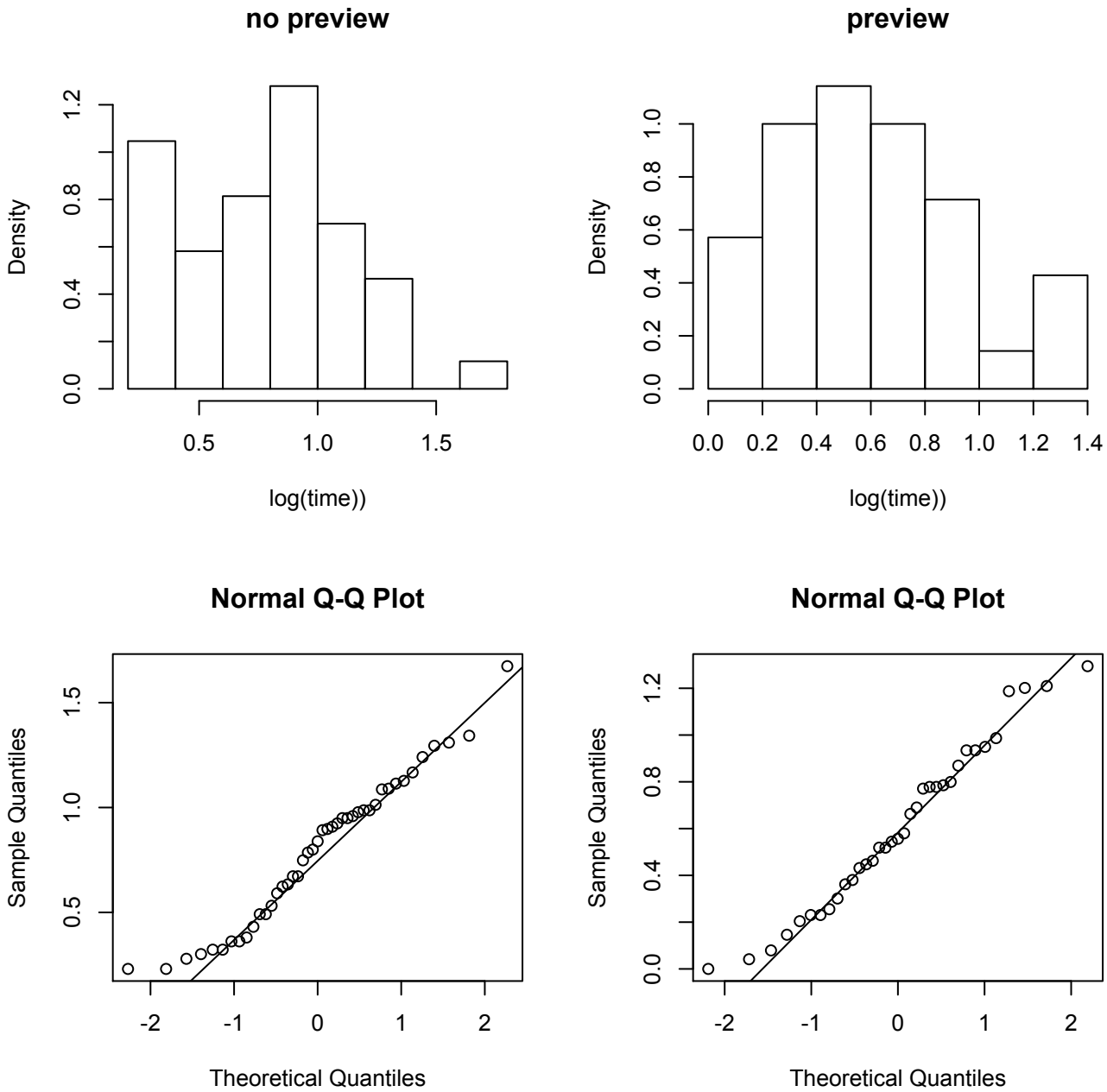

preview
Density
0.00
0.04
0.08
0.12
0
5
10
15
20
time(s)


Normal Q-Q Plot
Sample Quantiles
0
10
20
30
40
-2
-1
0
1
2
Theoretical Quantiles


Normal Q-Q Plot
Sample Quantiles
5
10
15
20
-2
-1
0
1
2
Theoretical Quantiles

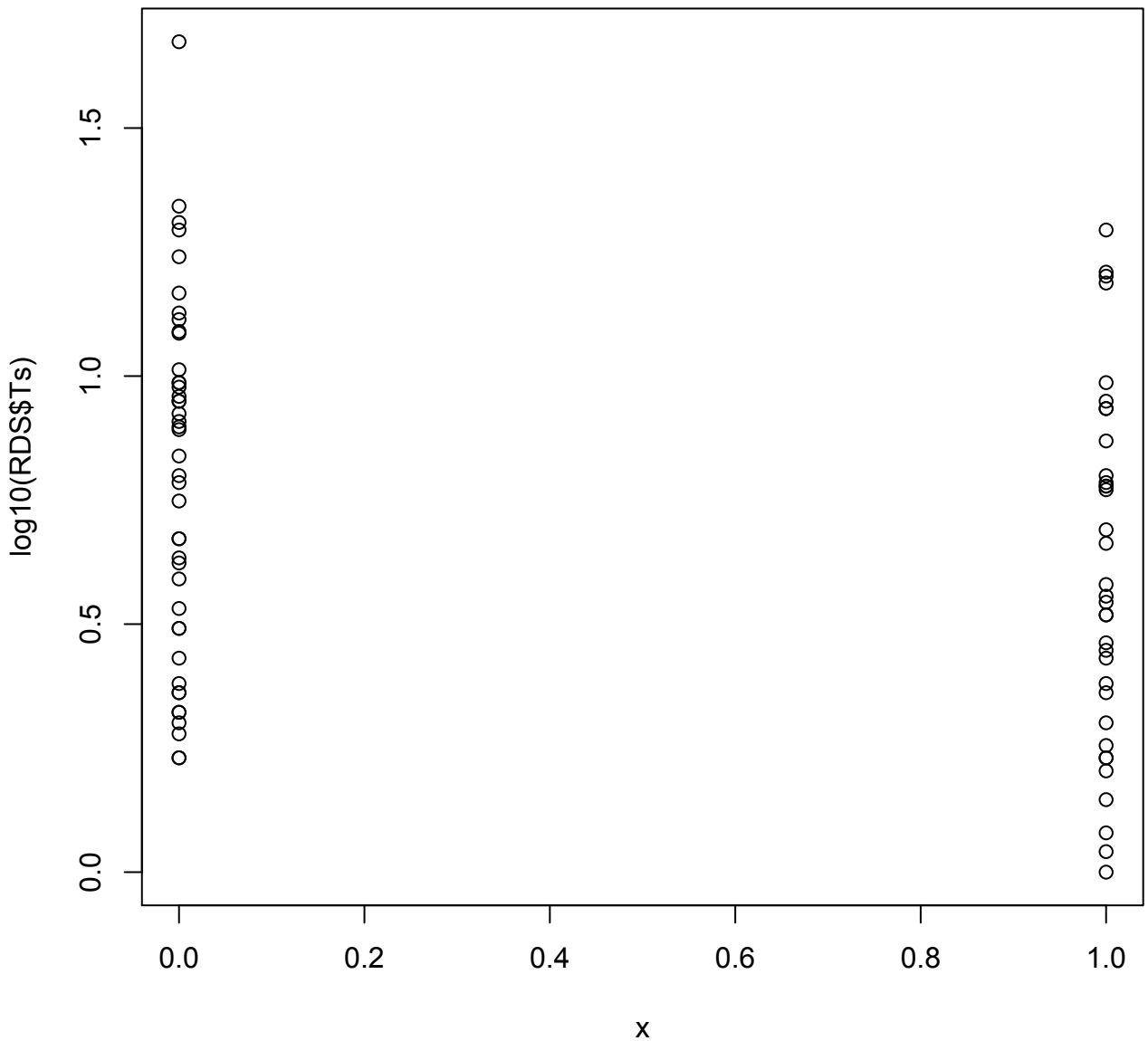
```
> boxplot(log10(RDS$Ts) ~ RDS$group, xlab = "gruppo",
ylab = "log(tempo)")
```




no preview
Density
0.0
0.4
0.8
1.2
0.5
1.0
1.5
log(time))
preview
Density
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
log(time))
Normal Q-Q Plot
Sample Quantiles
0.5
1.0
1.5
-2
-1
0
1
2
Theoretical Quantiles
Normal Q-Q Plot
Sample Quantiles
0.0
0.4
0.8
1.2
-2
-1
0
1
2
Theoretical Quantiles

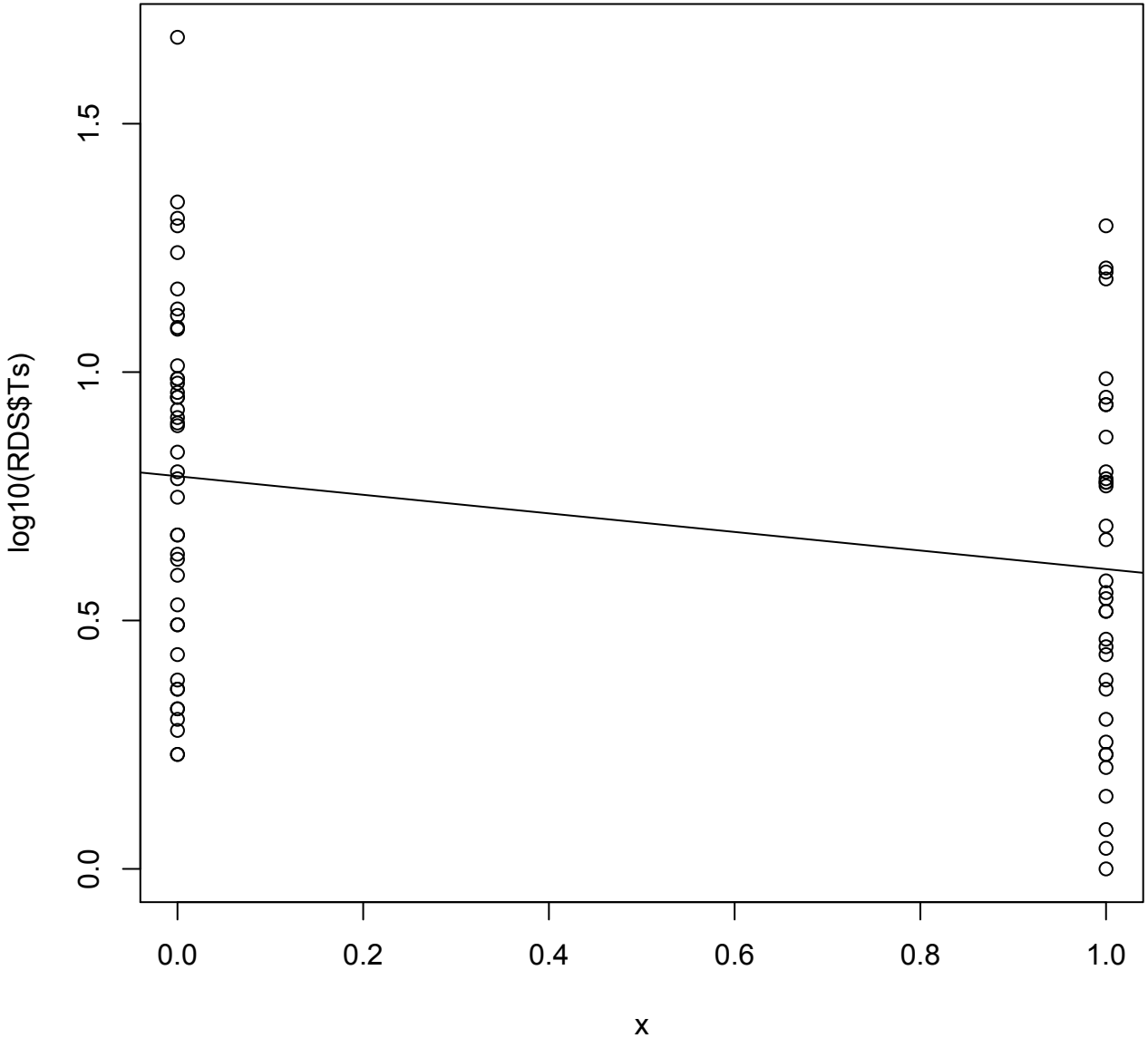
```
> RDS$group

 [1] NV NV NV NV NV NV NV NV NV NV NV NV NV NV NV NV NV NV NV NV
[22] NV NV NV NV NV NV NV NV NV NV NV NV NV NV NV NV NV NV NV NV
[43] NV VV VV VV VV VV VV VV VV VV VV VV VV VV VV VV VV VV VV VV
[64] VV VV VV VV VV VV VV VV VV VV VV VV VV VV VV
Levels: NV VV

> as.numeric(RDS$group)

 [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
[32] 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
[63] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

> x <- as.numeric(RDS$group) - 1
```

```
> plot(log10(RDS$Ts) ~ x)
```



```
> RDS.lm <- lm(log10(RDS$Ts) ~ x)

> coef(RDS.lm)
(Intercept)           x
  0.7904207  -0.1869884

> mVV <- mean(log10(RDS$Ts[RDS$group ==
"VV"]))
> mNV <- mean(log10(RDS$Ts[RDS$group ==
"NV"]))

> dm <- mVV - mNV
> dm
[1] -0.1869884

> mNV
[1] 0.7904207
```

```
> abline(RDS.lm)
```



misurare l'associazione

l'associazione fra x (categoria 0, 1) e y (tempo) può essere misurata da r

```
> cor(log10(RDS$Ts), x)
[1] -0.2570671
```

misurare l'associazione

ma r è anche data dalla pendenza, moltiplicata per il rapporto DSx/DSy

```
> lcf <- coef(RDS.lm)
>  (lcf[[2]] * sd(RDS$dc))/sd(log10(RDS$Ts))
[1] -0.2570671
```

d di Cohen

associazione (nel senso di *effect size)* fra una variabile categoriale dicotomica e un variabile numerica

differenza fra le medie, standardizzata

$$d = \frac{\bar{x}_1 - \bar{x}_2}{s}.$$

d di Cohen

***effect size* di una variabile categoriale dicotomica su una variabile numerica**

pendenza

$$d = \frac{\bar{x}_1 - \bar{x}_2}{s}.$$

DSy

```
> lcf[[2]]/sd(log10(RDS$Ts))
[1]  -0.513536
```

stima congiunta della *DS*

d **è una differenza fra medie, standardizzata (cioè divisa per una DS)**

ma è giusto calcolare così la DS?

in realtà, una stima migliore si ottiene con una media pesata delle DS prese separatamente

pesare in base alla precisione

se DS1 è basata su n1 casi, e DS2 è basata su n2 casi

la stima ottimale è una media pesata usando

$w1 = n1/(n1 + n2)$

$w2 = n2/(n1 + n2)$

ma partendo dalla varianza!

$$s = \sqrt{\frac{n1 - 1}{n1 + n2 - 2} \, var1 + \frac{n2 - 1}{n1 + n2 - 2} \, var2}$$

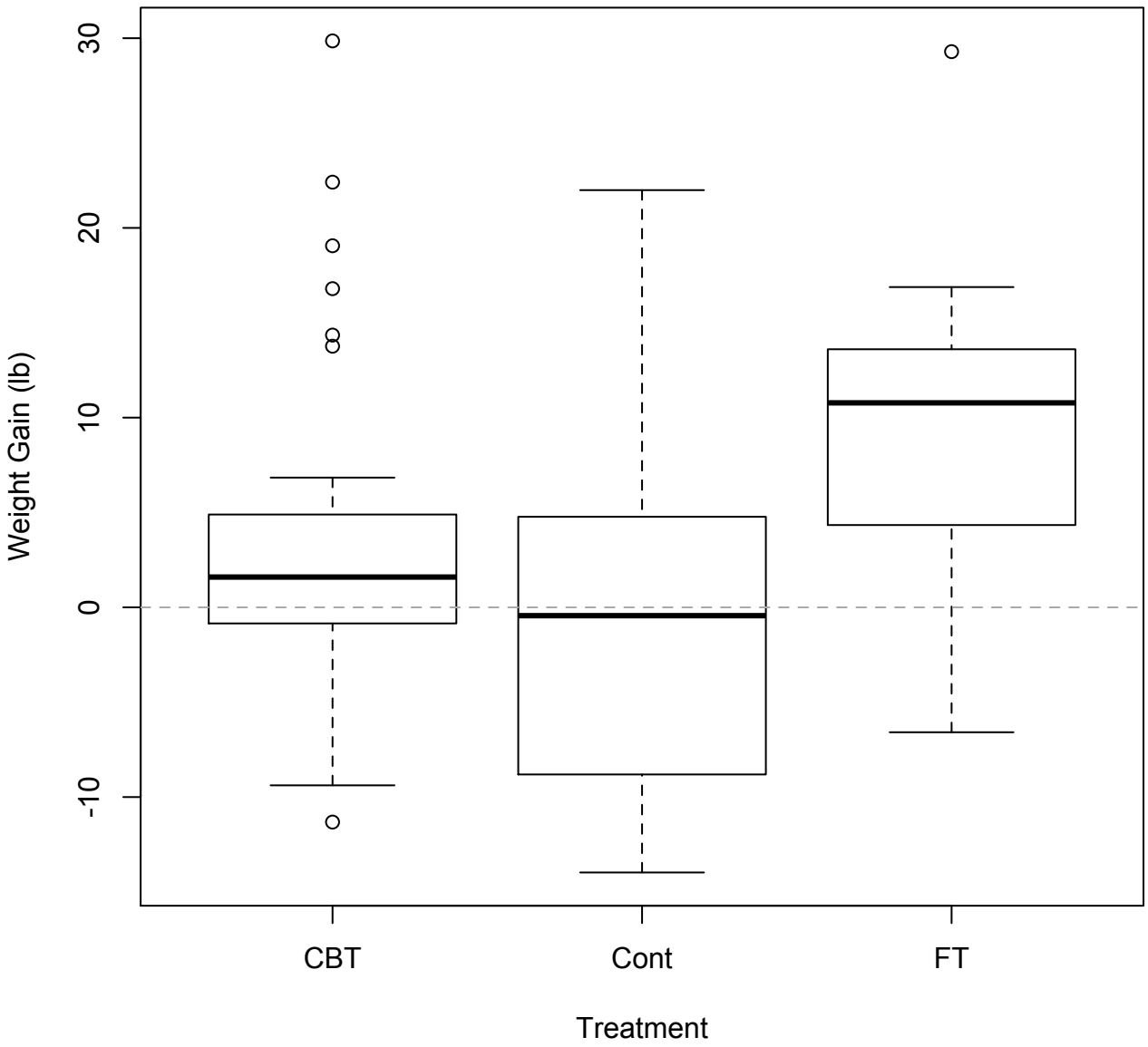
$$s = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}}$$

d di Cohen

```
>lt1 <- log10(RDS$Ts)[RDS$group == "NV"]
>lt2 <- log10(RDS$Ts)[RDS$group == "VV"]
>n1 <- length(t1)
>n2 <- length(t2)
>s<- sqrt(((n1-1)*var(lt1)+(n2-1)*var(lt2))/
(n1+n2-2))
>d <-  lcf[[2]]/s
>
>d
[1] -0.5279324
```

tre categorie

```r
library(MASS)
data(anorexia)
anorexia <- within(anorexia, difw <- 100 *
(Postwt-Prewt)/Prewt)
with(anorexia, boxplot(difw ~ Treat,
          ylim = c(min(difw), max(difw)),
          ylab = "Weight Gain (lb)",
          xlab = "Treatment"))
with(anorexia, abline(h=0,col="dark grey",
lty = "dashed"))
```


30
20
10
0
-10
Weight Gain (lb)
CBT
Cont
FT
Treatment

```
> str(anorexia)

'data.frame': 72 obs. of  4 variables:
 $ Treat : Factor w/ 3 levels "CBT","Cont","FT": 2 2
2 2 2 2 2 2 2 2 ...
 $ Prewt : num  80.7 89.4 91.8 74 78.1 88.3 87.3
75.1 80.6 78.4 ...
 $ Postwt: num  80.2 80.1 86.4 86.3 76.1 78.1 75.1
86.7 73.5 84.6 ...
 $ difw  : num  -0.62 -10.4 -5.88 16.62 -2.56 ...
```

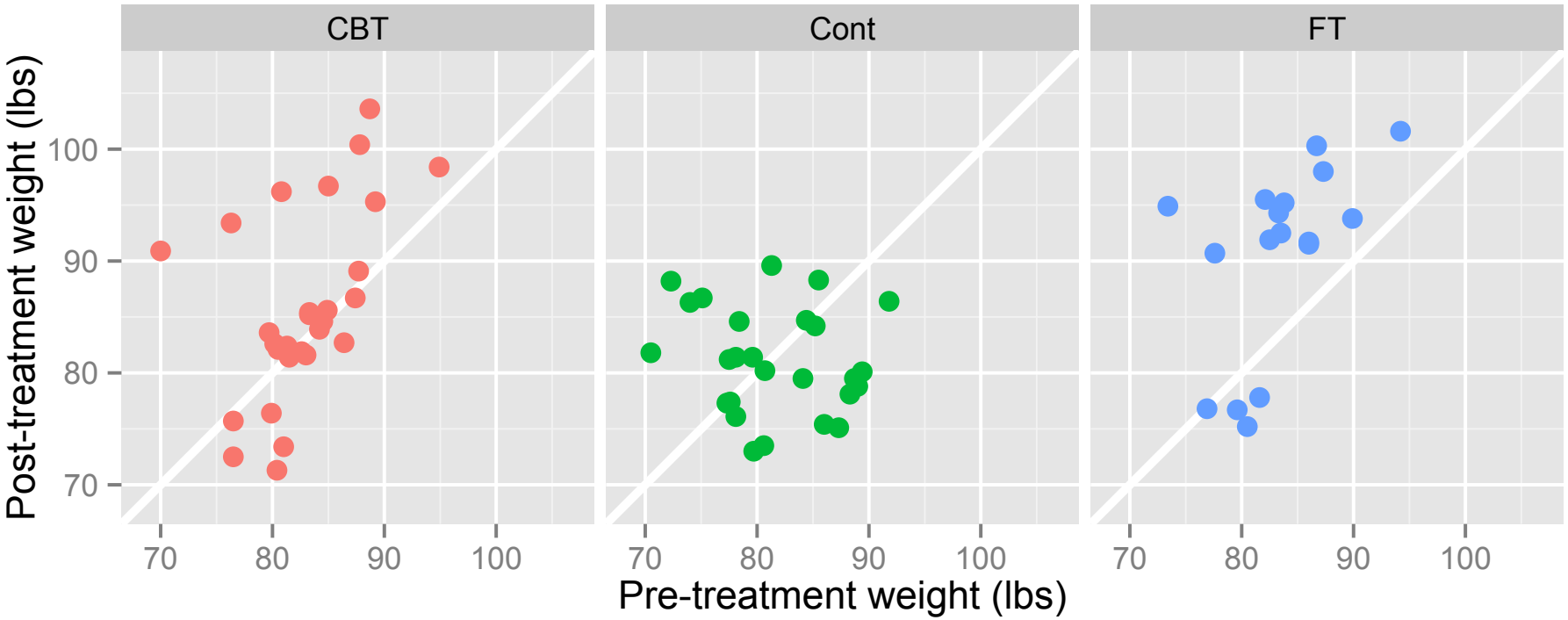
```
> str(anorexia)

'data.frame': 72 obs. of  4 variables:
 $ Treat : Factor w/ 3 levels "CBT","Cont","FT": 2 2 2 2 2 2 2 2 2
2 ...
 $ Prewt : num  80.7 89.4 91.8 74 78.1 88.3 87.3 75.1 80.6 78.4
...
 $ Postwt: num  80.2 80.1 86.4 86.3 76.1 78.1 75.1 86.7 73.5
84.6 ...
 $ difw  : num  -0.62 -10.4 -5.88 16.62 -2.56 ...

> anorexia$Treat

 [1] Cont Cont Cont Cont Cont Cont Cont Cont Cont Cont Cont Cont
[13] Cont Cont Cont Cont Cont Cont Cont Cont Cont Cont Cont Cont
[25] Cont Cont CBT  CBT  CBT  CBT  CBT  CBT  CBT  CBT  CBT  CBT
[37] CBT  CBT  CBT  CBT  CBT  CBT  CBT  CBT  CBT  CBT  CBT  CBT
[49] CBT  CBT  CBT  CBT  CBT  CBT  CBT  FT   FT   FT   FT   FT
[61] FT   FT   FT   FT   FT   FT   FT   FT   FT   FT   FT   FT
Levels: CBT Cont FT
```

```
> anorexia$Treat

 [1] Cont Cont Cont Cont Cont Cont Cont Cont Cont Cont Cont Cont
[13] Cont Cont Cont Cont Cont Cont Cont Cont Cont Cont Cont Cont
[25] Cont Cont CBT  CBT  CBT  CBT  CBT  CBT  CBT  CBT  CBT  CBT
[37] CBT  CBT  CBT  CBT  CBT  CBT  CBT  CBT  CBT  CBT  CBT  CBT
[49] CBT  CBT  CBT  CBT  CBT  CBT  CBT  FT   FT   FT   FT   FT
[61] FT   FT   FT   FT   FT   FT   FT   FT   FT   FT   FT   FT
Levels: CBT Cont FT

> anorexia$dum1 <- c(rep(0, 26), rep(1, 29),
rep(0, 17))
> anorexia$dum2 <- c(rep(0, 26), rep(0, 29),
rep(1, 17))
```

```
> head(anorexia)
  Treat Prewt Postwt       difw    dum1 dum2
1  Cont  80.7   80.2  -0.6195787     0    0
2  Cont  89.4   80.1 -10.4026846     0    0
3  Cont  91.8   86.4  -5.8823529     0    0
4  Cont  74.0   86.3  16.6216216     0    0
5  Cont  78.1   76.1  -2.5608195     0    0
6  Cont  88.3   78.1 -11.5515289     0    0

> tail(anorexia)
   Treat Prewt Postwt      difw    dum1 dum2
67    FT  82.1   95.5 16.321559     0    1
68    FT  77.6   90.7 16.881443     0    1
69    FT  83.5   92.5 10.778443     0    1
70    FT  89.9   93.8  4.338154     0    1
71    FT  86.0   91.7  6.627907     0    1
72    FT  87.3   98.0 12.256586     0    1
```

```
> anor.lm <- with(anorexia, lm(difw ~ dum1 + dum2))
> anor.lm

Call:
lm(formula = difw ~ dum1 + dum2)

Coefficients:
(Intercept)         dum1         dum2
  -0.006556     3.730528     8.807848

> tapply(anorexia$difw, anorexia$Treat, mean)

       CBT        Cont                FT
 3.72397187 -0.00655586  8.80129215
```

```
> anor.lm <- with(anorexia, lm(difw ~ dum1 + dum2))
> anor.lm

Call:
lm(formula = difw ~ dum1 + dum2)

Coefficients:
(Intercept)         dum1         dum2
  -0.006556     3.730528     8.807848

> tapply(anorexia$difw, anorexia$Treat, mean)

       CBT        Cont                FT
 3.72397187 -0.00655586  8.80129215
```


CBT
Cont
FT
Post-treatment weight (lbs)
100
90
80
70
70
80
90
100
Pre-treatment weight (lbs)