1 Analisi univariata

Dataset per gli esempi: `laureati` Dati di 1363 laureati di economia di Trento:

- `corso`: `EC`=economia commercio, `EP`=economia politica
- `anno`: anno di laurea
- `media`: media esami
- `voto`: voto di laurea

```
library(lbdatasets)
head(laureati)
```

```
##   corso anno    media voto
## 1    EC 2000 20.96182   77
## 2    EP 2000 23.03727   89
## 3    EP 2000 27.29182  110
## 4    EC 2000 22.87091   86
## 5    EC 2000 23.43545   89
## 6    EP 2000 23.61000   89
```

1.1 Stima di modello e riproduzione componenti

Partiamo da una prima stima del dataset laureati con il voto di laurea studiato in funzione della media voti: la seguente stima suggerisce come all'aumentare di un punto di media, il voto di laurea aumenti mediamente di 4.29 punti

```
mod_l <- lm(voto ~ media, data = laureati)
summary(mod_l)
```

```
##
## Call:
## lm(formula = voto ~ media, data = laureati)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.1076 -1.9126 -0.2796  1.5666 15.4043
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -10.51411    0.96972  -10.84   <2e-16 ***
## media         4.29213    0.03938  108.99   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.542 on 1361 degrees of freedom
## Multiple R-squared:  0.8972,Adjusted R-squared:  0.8971
## F-statistic: 1.188e+04 on 1 and 1361 DF,  p-value: < 2.2e-16
```

1.2 Estrazione componenti

Una volta stimato il modello possiamo accedere a diversi valori generati dalla stima

```
## Coefficienti stimati
coefficients(mod_l)

## (Intercept)       media
##   -10.51411     4.29213

## Intervallo di confidenza per i coefficienti stimati
confint(mod_l)

##                   2.5 %    97.5 %
## (Intercept) -12.416406 -8.611809
## media         4.214878  4.369382

## Fitted values: solo i primi stampati
head(fitted(mod_l))

##         1         2         3         4         5         6
##  79.45674  88.36486 106.62592  87.65081  90.07391  90.82308

## Residui della regressione: solo i primi stampati
head(residuals(mod_l))

##          1          2          3          4          5          6
## -2.4567411  0.6351375  3.3740773 -1.6508074 -1.0739092 -1.8230806

## Errori standard dei coefficienti: ottenuti da vcov che restituisce
## matrice di varianza/covarianza per i parametri del modelli
(seb <- sqrt(diag(vcov(mod_l))))

## (Intercept)       media
##  0.96971506  0.03937989
```

Alcune di queste funzioni ci tornano utili per riprodurre alcune statistiche fornite nel `summary`; posticipiamo l'ultima riga (test F sul modello) alla sezione sull'analisi multivariata;

```r
## R^2
(TSS_1 <- sum((laureati$voto - mean(laureati$voto))^2))
```

```
## [1] 85558.26
```

```r
(ESS_1 <- sum((fitted(mod_1) - mean(laureati$voto))^2))
```

```
## [1] 76763.64
```

```r
(RSS_1 <- sum((residuals(mod_1))^2))
```

```
## [1] 8794.617
```

```r
(r2 <- ESS_1 / TSS_1)
```

```
## [1] 0.897209
```

```r
## Errore standard dei residui
(resid_df <- nrow(laureati) - 1 - 1)
```

```
## [1] 1361
```

```r
(rse <- sqrt( sum((residuals(mod_1)^2)) / (resid_df) ))
```

```
## [1] 2.542023
```

```r
## test T
t <- coefficients(mod_1) / seb
t_df <- nrow(laureati) - 1 - 1
## p-value per media: specifichiamo la parte a destra della distribuzione
## mediante lower.tail = FALSE dato che il coefficiente della media è positivo;
## moltiplichiamo per due per avere la probabilità che
## il test sia più estremo in valore assoluto
pt(q = t["media"], df = t_df, lower.tail = FALSE) * 2
```

```
## media
##     0
```

```r
## Intervalli di confidenza
df <- nrow(laureati) - 1 - 1
(upper <- coefficients(mod_1) + qt(0.975, df = df) * seb)
```

```
## (Intercept)       media
##   -8.611809    4.369382
```

```r
(lower <- coefficients(mod_1) + qt(0.025, df = df) * seb)
```

```
## (Intercept)       media
##  -12.416406    4.214878
```

1.3 Centratura, scaling

Entrambi si utilizzano a fini interpretativi:

- centrare significa analizzarne il suo scarto rispetto ad un valore di riferimento (spesso ma non necessariamente la media) e serve per ottenere stime nell'intercetta di particolari popolazioni di riferimento;
- effettuare lo scaling significa dividerla per la sua deviazione standard e serve per avere questa come unità di misura nell'interpretazione.

1.3.1 Centratura della indipendente

Centriamo nel modello il voto al 18 ossia stimiamo il modello

$$y_i = b_0 + b_1(x_i - 18)$$

Possiamo farlo senza modificare i dati compattamente attraverso:

```
fit <- lm(voto ~ I(media - 18), data = laureati)
summary(fit)

##
## Call:
## lm(formula = voto ~ I(media - 18), data = laureati)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.1076 -1.9126 -0.2796  1.5666 15.4043
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)   66.74423    0.26744   249.6   <2e-16 ***
## I(media - 18)  4.29213    0.03938   109.0   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.542 on 1361 degrees of freedom
## Multiple R-squared:  0.8972,Adjusted R-squared:  0.8971
## F-statistic: 1.188e+04 on 1 and 1361 DF,  p-value: < 2.2e-16
```

Rispetto ai risultati visti in precedenza l'unica cosa che cambia è il valore dell'intercetta, che rappresenterà sempre il valore medio della laurea per coloro che hanno 0 come voto centrato (ossia 18 di voto); pertanto la stima del voto medio per coloro che hanno 18 di media è 66.

1.3.2 Scaling della indipendente

Scaliamo il voto medio stimando il modello

$$y_i = b_0 + b_1 \frac{(x_i - \mu_x)}{\sigma_x}$$

La funzione `scale` sottrae la media e divide per la deviazione standard; si ha pertanto che 94 è la stima del voto medio di laurea per un soggetto che ha una media voti media; 7.5 è l'incremento del voto di laurea se la media voti aumenta di una deviazione standard

```
fit <- lm(voto ~ scale(media), data = laureati)
summary(fit)

##
## Call:
## lm(formula = voto ~ scale(media), data = laureati)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.1076 -1.9126 -0.2796  1.5666 15.4043
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  94.91123    0.06885    1378   <2e-16 ***
## scale(media)  7.50739    0.06888     109   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.542 on 1361 degrees of freedom
## Multiple R-squared:  0.8972,Adjusted R-squared:  0.8971
## F-statistic: 1.188e+04 on 1 and 1361 DF,  p-value: < 2.2e-16
```

1.3.3 Centratura doppia

Nel caso si centri sia indipendente che dipendente rispetto alla media,

$$(y_i - \mu_y) = b_0 + b_1(x_i - \mu_x)$$

l'intercetta diviene lo scarto dalla media di y quando x assume la propria media; la pendenza diviene la variazione dello scarto di y dalla propria media all'aumentare di un punto di quello di x

```
f <- I(voto - mean(voto)) ~ I(media - mean(media))
fit <- lm(formula = f, data = laureati)
summary(fit)
```

```
##
## Call:
## lm(formula = f, data = laureati)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.1076 -1.9126 -0.2796  1.5666 15.4043
##
## Coefficients:
##                           Estimate Std. Error t value Pr(>|t|)
## (Intercept)             -1.220e-15  6.885e-02       0        1
## I(media - mean(media))  4.292e+00  3.938e-02     109   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.542 on 1361 degrees of freedom
## Multiple R-squared:  0.8972,Adjusted R-squared:  0.8971
## F-statistic: 1.188e+04 on 1 and 1361 DF,  p-value: < 2.2e-16
```

1.3.4 Scaling doppio

Nel caso si centri sia indipendente che dipendente rispetto alla media,

$$\frac{(y_i - \mu_y)}{\sigma_y} = b_0 + b_1 \frac{(x_i - \mu_x)}{\sigma_x}$$

si ottiene un modello simile alla doppia centratura ma dove le unità di misura non son più le originali bensì le deviazioni standard dei fenomeni analizzati

```
fit <- lm(scale(voto) ~ scale(media), data = laureati)
summary(fit)
```

```
##
## Call:
## lm(formula = scale(voto) ~ scale(media), data = laureati)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.77060 -0.24132 -0.03528  0.19765  1.94357
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.562e-16  8.687e-03       0        1
## scale(media)  9.472e-01  8.691e-03     109   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3207 on 1361 degrees of freedom
## Multiple R-squared:  0.8972,Adjusted R-squared:  0.8971
## F-statistic: 1.188e+04 on 1 and 1361 DF,  p-value: < 2.2e-16
```

L'intercetta è 0 come nella doppia centratura ma la pendenza è 0.95: quando x incrementa di una deviazione standard, y incrementa di 0.95 deviazioni standard (ossia circa 7.5 punti.

2 Modello multivariato

2.1 Dataset per gli esempi: `mental`

Un dataset dell'Agresti riguardante 40 soggetti con 3 variabili:

- `impair`: indice di disagio mentale (range 17-41), variabile dipendente
- `stress`: punteggio eventi stressanti della vita (3-97)
- `status`: socio economico (0-100)

```
head(mental)

##   impair stress status
## 1     17     46     84
## 2     19     39     97
## 3     20     27     24
## 4     20      3     85
## 5     20     10     15
## 6     21     44     55
```

2.2 Alcune stime

```
# modelli con covariate separatamente
mod_stress <- lm(impair ~ stress, data = mental)
mod_status <- lm(impair ~ status, data = mental)
# modelli con entrambe, senza e con interazione
mod_noint  <- lm(impair ~ stress + status, data = mental)
mod_int    <- lm(impair ~ stress * status, data = mental)
```

Partiamo considerando i modelli con le covariate separatamente e il modello con entrambe ma senza interazione: si nota che essendo bassa la correlazione tra le misure, i coefficienti di `stress` e `status` non subiscono grossi scossoni nel

passare da stima univariata a multivariata. Per entrambi l'aumento dello stress e la diminuzione dello status sono associati ad un incremento del punteggio di disagio mentale

```
##  Vediamo la correlazione tra le misure del dataset: si nota che non è
## altissima tra le covariate
round(cor(mental), 2)

##        impair stress status
## impair   1.00   0.37  -0.40
## stress   0.37   1.00   0.12
## status  -0.40   0.12   1.00

mod_stress

##
## Call:
## lm(formula = impair ~ stress, data = mental)
##
## Coefficients:
## (Intercept)       stress
##    23.30949      0.08983

mod_status

##
## Call:
## lm(formula = impair ~ status, data = mental)
##
## Coefficients:
## (Intercept)       status
##    32.17201     -0.08608

mod_noint

##
## Call:
## lm(formula = impair ~ stress + status, data = mental)
##
## Coefficients:
## (Intercept)       stress       status
##    28.22981      0.10326     -0.09748
```

2.3 Modello nullo, test F globale

Poniamo attenzione al modello con entrambe le covariate e calcoliamo alcune statistiche

```r
summary(mod_noint)
```

```
##
## Call:
## lm(formula = impair ~ stress + status, data = mental)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.678 -2.494 -0.336  2.886 10.891
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 28.22981    2.17422  12.984 2.38e-15 ***
## stress       0.10326    0.03250   3.177  0.00300 **
## status      -0.09748    0.02908  -3.351  0.00186 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.556 on 37 degrees of freedom
## Multiple R-squared:  0.3392,Adjusted R-squared:  0.3034
## F-statistic: 9.495 on 2 and 37 DF,  p-value: 0.0004697
```

```r
(TSS_noint <- sum((mental$impair - mean(mental$impair))^2))
```

```
## [1] 1162.4
```

```r
(ESS_noint <- sum((fitted(mod_noint) - mean(mental$impair))^2))
```

```
## [1] 394.2384
```

```r
(RSS_noint <- sum((residuals(mod_noint))^2))
```

```
## [1] 768.1616
```

Il test finale riportato nell'output di `lm` saggia l'ipotesi che tutti i coefficienti siano congiuntamente nulli. Per farlo confronta il modello stimato con quello nullo (ossia senza covariate); il coefficiente dell'intercetta, migliore approssimazione dei dati che possiamo dare, è la media dei voti:

```r
## Test F sul modello completo
summary(mod_null <- lm(impair ~ 1, data = mental))
```

```
##
## Call:
## lm(formula = impair ~ 1, data = mental)
##
## Residuals:
```

```
##    Min     1Q Median     3Q    Max
## -10.30  -3.55  -0.30   3.70  13.70
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  27.3000     0.8632   31.63   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 5.459 on 39 degrees of freedom
```

```
mean(mental$impair)
```

```
## [1] 27.3
```

```
## Nel modello nullo TSS e RSS coincidono (TSS è sempre il solito),
## mentre ESS è 0
(TSS_null <- sum((mental$impair - mean(mental$impair))^2))
```

```
## [1] 1162.4
```

```
(ESS_null <- sum((fitted(mod_null) - mean(mental$impair))^2))
```

```
## [1] 2.966117e-27
```

```
(RSS_null <- sum((residuals(mod_null))^2))
```

```
## [1] 1162.4
```

TSS è sempre 1162.4: se si passa dal modello nullo al modello con la media voti ESS passa da 2.966117×10^{-27} a 394.2383986 e specularmente RSS scende da 1162.4 a 768.1616014.
Possiamo confrontare i due modelli, prestando attenzione alle devianze residue RSS:

- la devianza residua dal modello nullo rappresenta la variabilità non spiegata da parte di un modello che predice la dipendente mediante la sua media (per il *mental impairment*, 1162 punti)
- la devianza residua del modello analizzato rappresenta la quota di variabilità non spiegata dopo aver introdotto le variabili del modello (768)
- la differenza tra le due (394) è interpretabile come la quota di variabilità della dipendente spiegata dall'introduzione delle due variabili indipendenti (quei punti potrebbero esser legati a x_1, o x_2, non importa).

Per la spiegazione di una maggiore variabilità, c'è un prezzo da pagare e le devianze debbono esser pesate dai gradi di libertà del modello (solitamente riportati alla destra della devianza):

Fonte variazione	Devianza	DF	Varianza
Regressione	394.2	2	$394.2/2 = 197.1$
Residua	768.2	37	20.8
Totale	1162.4	39	

Tabella 1: Analisi della varianza per l'introduzione di due variabili nel modello di *mental impairment*

- i gdl del modello nullo sono $n - 1$ (39 in questo dataset);
- nel modello analizzato i gdl sono $n - k - 1$ (37);
- la differenza (2) sono i gradi di libertà impiegati per migliorare la previsione (ovvero il numero di parametri stimati).

Una volta ottenute devianza e relativi gradi di libertà possiamo calcolare le varianze effettuando il rapporto. Riportiamo innanzitutto i dati sulla tabella classica dell'analisi della varianza (tab 1). Ci si può chiedere: guadagnare 394 unità di devianza, compensa due gradi di libertà persi? Per farlo si utilizza il test F, che è un rapporto tra varianze, calcolato nel nostro caso come segue:

$$F_{2,37} = \frac{197.1}{20.8} = 9.4 \tag{1}$$

Se fosse vera l'ipotesi nulla (coefficienti del modello tutti nulli), la variabilità spiegata dalla regressione (ovvero dall'introduzione delle due variabili) dovrebbe essere 0[1], e con esso dovrebbe esser nullo la statistica test F nel suo complesso. Per dire che tale affermazione è falsa, abbiamo bisogno della distribuzione del test: che si distribuisce come una F con n e d gradi di libertà (ovvero gradi di libertà del numeratore e del denominatore), una distribuzione con supporto strettamente positivo e con coda a destra. Il nostro problema è di determinare il valore critico:

- considerando un α di 0.05, esso può esser determinato in R mediante `qf`:

```
qf(0.95, 2, 37)

## [1] 3.251924
```

- essendo 9.48 ben più grande di 3.25, rifiutiamo l'ipotesi nulla, concludendo che almeno una delle variabili introdotte nel modello è correlata con la dipendente;

[1] Difficilmente sarà 0, a meno che in toto le variabili introdotte siano perfettamente incorrelate con la dipendente

Per ripetere i calcoli mediante R (a parte guardarlo nel `summary`) si debbono confrontare mediante `anova` il modello nullo e il modello analizzato

```
anova(mod_null, mod_noint)

## Analysis of Variance Table
##
## Model 1: impair ~ 1
## Model 2: impair ~ stress + status
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1     39 1162.40
## 2     37  768.16  2    394.24 9.4946 0.0004697 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

## oppure per il calcolo del p-value
(num <- ESS_noint / 2)

## [1] 197.1192

(den <- RSS_noint / 37)

## [1] 20.76112

(f <- num/den)

## [1] 9.49463

pf(q = f, df1 = 2, df2 = 37, lower.tail = FALSE)

## [1] 0.0004696717
```

2.4 Strumenti per analisi della devianza di un modello

- se si fornisce solo un modello di stima, R spacchetta il contributo delle singole variabili alla spiegazione della devianza totale; in questo schema, l'ordine di specificazione delle indipendenti nel modello è rilevante al fine dei calcoli (ovvero nella riga di *status* si ha la varianza spiegata da *status* nell'aggiungerlo al modello avente *stress*):

```
anova(mod_noint)

## Analysis of Variance Table
##
## Response: impair
##           Df Sum Sq Mean Sq F value   Pr(>F)
```

```
## stress     1 161.05 161.048  7.7572 0.008385 **
## status     1 233.19 233.190 11.2320 0.001863 **
## Residuals 37 768.16  20.761
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Quindi questi sono i contributi sequenziali. Se cambiamo l'ordine di introduzione delle variabili, i risultati verosimilmente cambiano; in un unico caso ciò non avviene, e questo è qualora se le variabili indipendenti siano perfettamente incorrelate fra loro. In tale situazione quello che spiega un regressore non lo spiega l'altro e alterare la sequenza di inserimento non ha effetto.

- il confronto con un modello nested valuta l'incremento della devianza spiegata:

```
anova(mod_stress, mod_noint)

## Analysis of Variance Table
##
## Model 1: impair ~ stress
## Model 2: impair ~ stress + status
##   Res.Df     RSS Df Sum of Sq      F   Pr(>F)
## 1     38 1001.35
## 2     37  768.16  1    233.19 11.232 0.001863 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Una considerazione sul test F. Il test F, rispetto al test t, ha il *vantaggio* che può esser utilizzato per testare più coefficienti contemporaneamente; se il test F testa però solo una stima le implicazioni che ne derivano sono le medesime (F è il quadrato di t), anche se il test rimane più generale

- se viceversa si utilizza `drop1` viene effettuato il test di confronto con il modello non avente la variabile in riga: se questa occupa un solo grado di libertà le conclusioni inferenziali sono le stesse di quelle ottenute col test t del `summary`; viceversa diviene di interesse per le variabili categoriche, nel qual caso fornisce un unico test legato a tutti i coefficienti

```
drop1(mod_noint, test = "F")

## Single term deletions
##
## Model:
## impair ~ stress + status
```

```
##        Df Sum of Sq     RSS    AIC F value   Pr(>F)
## <none>               768.16 124.20
## stress  1    209.58  977.75 131.85  10.095 0.002998 **
## status  1    233.19 1001.35 132.81  11.232 0.001863 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

- `add1` effettua i test di comparazione tra il modello attuale e quello che si ottiene aggiungendo 1 variabile tra quelle specificate in `scope`:

```
add1(mod_null, scope = ~ . + stress + status, test = "F")

## Single term additions
##
## Model:
## impair ~ 1
##        Df Sum of Sq     RSS    AIC F value  Pr(>F)
## <none>              1162.40 136.78
## stress  1    161.05 1001.35 132.81  6.1116 0.01802 *
## status  1    184.65  977.75 131.85  7.1766 0.01085 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(mod_null, mod_stress)

## Analysis of Variance Table
##
## Model 1: impair ~ 1
## Model 2: impair ~ stress
##   Res.Df    RSS Df Sum of Sq      F  Pr(>F)
## 1     39 1162.4
## 2     38 1001.4  1    161.05 6.1116 0.01802 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

anova(mod_null, mod_status)

## Analysis of Variance Table
##
## Model 1: impair ~ 1
## Model 2: impair ~ status
##   Res.Df     RSS Df Sum of Sq      F  Pr(>F)
## 1     39 1162.40
## 2     38  977.75  1    184.65 7.1766 0.01085 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

2.5 Interazione ed inferenza

Focalizzandoci ora sul confronto tra i modelli con entrambe le covariate senza e con interazione.

```
summary(mod_noint)
```

```
##
## Call:
## lm(formula = impair ~ stress + status, data = mental)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -8.678 -2.494 -0.336  2.886 10.891
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 28.22981    2.17422  12.984 2.38e-15 ***
## stress       0.10326    0.03250   3.177  0.00300 **
## status      -0.09748    0.02908  -3.351  0.00186 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.556 on 37 degrees of freedom
## Multiple R-squared:  0.3392,Adjusted R-squared:  0.3034
## F-statistic: 9.495 on 2 and 37 DF,  p-value: 0.0004697
```

```
summary(mod_int)
```

```
##
## Call:
## lm(formula = impair ~ stress * status, data = mental)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.5117 -2.2323 -0.3881  2.9763 11.4088
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   26.036648   3.948826   6.594 1.13e-07 ***
## stress         0.155865   0.085338   1.826   0.0761 .
## status        -0.060493   0.062674  -0.965   0.3409
## stress:status -0.000866   0.001297  -0.668   0.5087
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.591 on 36 degrees of freedom
```

```
## Multiple R-squared:  0.3472,Adjusted R-squared:  0.2928
## F-statistic: 6.383 on 3 and 36 DF,  p-value: 0.001396
```

L'introduzione di una interazione è l'introduzione di una variabile molto correlata alle già presenti (facile pensarlo qualora entrambe le variabili siano quantitative), quindi dal punto di vista inferenziale questo può provocare problemi al modello.
Il fatto di aver introdotto una variabile molto correlata ha apparentemente "scassato" tutto (t non significativi, ma test F è significativo, quindi almeno uno dei coefficienti è diverso da 0); in questo caso che fare?

Abbiamo due modelli, quello con e quello senza interazione: come scegliere tra i due? Effettuiamo il test F (che in questo semplice caso corrisponde al t), non possiamo rifiutare la nulla. La conclusione pertanto è che il modello di interazione è eccessivamente complicato e non vale la pena tenere quest'ultima dentro.

```
anova(mod_noint, mod_int)

## Analysis of Variance Table
##
## Model 1: impair ~ stress + status
## Model 2: impair ~ stress * status
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     37 768.16
## 2     36 758.77  1    9.3921 0.4456 0.5087
```

Considerando che in un modello vi siano alcune variabili che costituiscono gli *effetti principali* sulla dipendente (nel caso di sopra `stress` e `status`) ed altre *l'interazione*, possiamo avere alcune situazioni dubbie:

- interazione non significativa, effetti principali significativi;
- interazione significativa ed uno o più effetti principali di cui è composta non significativi: questo non è un caso inverosimile, dato che stiamo cercando di stimare il contributo netto di alcune variabili solitamente correlate fra loro (soprattutto se le due variabili degli effetti principali lo sono già).

Ipotizzando di venire da modelli con soli effetti principali come comportarsi?:

- nel primo caso si può procedere nel rispecificare il modello senza interazione.
- nel secondo posto che facendo l'anova per l'introduzione dell'interazione non si sbaglia mai, una strategia alternativa è la seguente. Se vi è evidenza forte di interazione non ha più interesse testare altre ipotesi (es che gli effetti principali siano nulli) e possiamo tenere il modello così specificato (non dobbiamo guardare gli altri test t, quelli degli effetti principali).

Infatti questi ultimi testano, separatamente, ipotesi non più interessanti, ovvero che il contributo parziale di una data variabile sia nullo: anche se vero, sappiamo che sebbene non in maniera diretta ma mediante un'altra variabile, la variabile il cui coefficiente non è statisticamente significativo *agisce* sulla dipendente. In altre parole se vi è interazione, l'effetto di ogni variabile esiste, in quanto differisce a seconda del valore assunto dall'altra variabile.
Quindi la regola è: non interessarsi della significatività degli effetti principali delle variabili che partecipano ad interazione. Ad esempio attenzione che nel modello

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 (x_1 \cdot x_2) + \beta_4 x_3 \tag{2}$$

il t di x_3 lo devo guardare perchè esso non partecipa all'interazione (che riguarda x_1 e x_2).
Messaggio: se l'interazione è significativa non ha senso parlare di effetti principali.

2.6 Scelta del modello

Possiamo stimare un modello per due ragioni, sostanzialmente:

- può essere che il modello da stimare sia **noto a priori** (es dettato dalla teoria o altro); se la teoria alla base è buona sono i dati ad adattarsi bene ad un modello prespecificato;

- può altrettanto essere che ci troviamo in una situazione con tanti dati raccolti e una variabile di cui desideriamo studiare la dipendenza dalle altre, ma non siamo sicuri di quali variabili includere o meno nel modello. Si rende necessaria dunque una **scelta del modello** e pertanto ci troviamo in una fase esplorativa. In funzione di ciò, sovrainterpretare le significatività dei coefficienti è un errore: il fatto che un modello scelto si adatti bene ai dati non è sorprendente, è stato "creato" sequenzialmente a partire da essi. Il rischio cui si va incontro è che il modello vada in *overfitting*: il problema è che essendo troppo il modello adattato ai dati, non è utile come generalizzazione.

Le volte che bisogna scegliere il modello è importante capire **l'obiettivo** del nostro studio:

- se l'obiettivo è la **previsione** occorre privilegiare modelli *parsimoniosi*; in questi casi vogliamo un modello che permetta a chiunque di prevedere cosa accadrà a soggetti che non hanno partecipato allo studio, che però sono simili o in situazioni analoghe a quelle studiate;

- se l'obiettivo è **esplorativo**, mettiamo dentro *tanti regressori*, se vengono non significativi commentiamo che non sono significativi, non è che lo ristimiamo togliendo le variabili non significative.

Dataset esempio: `fitness` Si vuole valutare quanto rapidamente il corpo può assorbire e usare l'ossigeno (cosa costosa/difficile da fare normalmente). Si conduce un esperimento in cui si desidera sviluppare una equazione di previsione per il consumo di ossigeno basato sulla misura di certi parametri (di facile quantificazione/ottenimento) raccolti durante esercizi fisici effettuati dai soggetti in studio (tutte variabili quantitative continue):

- `oxy` consumo di ossigeno (ml per kg di peso al minuto), la variabile da spiegare;
- `age`: età in anni;
- `weight`: peso in kg;
- `runtime`: minuti necessari per correre 1.5 miglia;
- `rstpulse`: battito cardiaco a riposo;
- `runpulse`: battito cardiaco terminata la corsa;
- `maxpulse`: battito cardiaco massimo (durante la corsa).

```
head(fitness)
```

```
##      oxy age weight runtime rstpulse runpulse maxpulse
## 1 44.609  44  89.47   11.37       62      178      182
## 2 45.313  40  75.07   10.07       62      185      185
## 3 54.297  44  85.84    8.65       45      156      168
## 4 59.571  42  68.15    8.17       40      166      172
## 5 49.874  38  89.02    9.22       55      178      180
## 6 44.811  47  77.45   11.63       58      176      176
```

2.6.1 Analisi di correlazione, multicollinearità e collinearità perfetta

```
round(cor(fitness[, -1]), 2)
```

```
##            age weight runtime rstpulse runpulse maxpulse
## age       1.00  -0.23    0.19    -0.16    -0.34    -0.43
## weight   -0.23   1.00    0.14     0.04     0.18     0.25
## runtime   0.19   0.14    1.00     0.45     0.31     0.23
## rstpulse -0.16   0.04    0.45     1.00     0.35     0.31
## runpulse -0.34   0.18    0.31     0.35     1.00     0.93
## maxpulse -0.43   0.25    0.23     0.31     0.93     1.00
```

`maxpulse` e `runpulse` sono molto correlate (multicollinearità: ossia alta correlazione di almeno due covariate); persino con correlazioni di 0.30 ci possono esser problemi nel modello di regressione multipla. Ad includerle entrambe potremmo avere stranezze del tipo:

- coefficienti che dovrebbero risultare positivi ci vengono negativi;
- nessuno dei due regressori correlati venga significativo e se ne lascio uno viene tutto significativo.

Strategie possibili:

- stimare un modello con tutti i regressori: dopodichè provo a togliere una delle variabili molto correlate e rifitto per vedere se va meglio. Nel caso occorra scegliere tra due variabili si potrà privilegiarne una se:
  - è piu facile rilevarla
  - sia meno prona ad errori la sua rilevazione rispetto a quella dell'altra
  - magari è più "importante" da un punto di vista teorico
- componenti principali
- regressione penalizzata (ridge, lasso etc)

Se infine nel modello vi è una **dipendenza lineare esatta** tra covariate (es sulla base di una o più è possibile determinarne un'altra) la matrice $\mathbf{X}^T\mathbf{X}$ è singolare e dunque non c'è una soluzione unica per le stime dei coefficienti con il metodo dei minimi quadrati (problema della *collinearità perfetta*): vediamone un esempio con R

```
y <- rnorm(100)
x1 <- 1:100
x2 <- x1 * 2
summary(lm(y ~ x1 + x2))

##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.54373 -0.44506  0.06091  0.66515  1.89507
##
## Coefficients: (1 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.060608   0.177686   0.341    0.734
## x1          -0.000655   0.003055  -0.214    0.831
## x2                NA         NA      NA       NA
##
## Residual standard error: 0.8818 on 98 degrees of freedom
## Multiple R-squared:  0.0004689,Adjusted R-squared:  -0.00973
## F-statistic: 0.04598 on 1 and 98 DF,  p-value: 0.8307
```

2.6.2 Stima complessiva

```
mod_oxy <- lm(oxy ~ ., data = fitness)
summary(mod_oxy)

##
## Call:
## lm(formula = oxy ~ ., data = fitness)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.4026 -0.8991  0.0706  1.0496  5.3847
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 102.93448   12.40326   8.299 1.64e-08 ***
## age          -0.22697    0.09984  -2.273  0.03224 *
## weight       -0.07418    0.05459  -1.359  0.18687
## runtime      -2.62865    0.38456  -6.835 4.54e-07 ***
## rstpulse     -0.02153    0.06605  -0.326  0.74725
## runpulse     -0.36963    0.11985  -3.084  0.00508 **
## maxpulse      0.30322    0.13650   2.221  0.03601 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.317 on 24 degrees of freedom
## Multiple R-squared:  0.8487,Adjusted R-squared:  0.8108
## F-statistic: 22.43 on 6 and 24 DF,  p-value: 9.715e-09
```

È buona norma non partire dal guardare i p-values bensì dai coefficienti, per vedere se ci sono cose strampalate (es segni di effetto contrario); tutte le stime sono negative ecetto quella di `maxpulse` e nello specifico:

- il consumo dell'ossigeno qui diminuisce all'aumentare dell'età: questa cosa è ragionevole;
- peso stessa cosa (anche se non significativa);
- quando il tempo di percorrenza `runtime` aumenta il consumo di ossigeno diminuisce: torna anche questo;
- tornano meno le variabili sulle pulsazioni.

L'effetto di `weight` e `rstpulse` non è significativo.

2.6.3 Valtutazione della multicollinearità

Oltre alla matrice di correlazione, la collinearità nel modello viene attraverso i VIF, ossia

$$VIF = \frac{1}{1 - R^2}$$

dove R^2 è quello del modello in cui si pone a turno una variabile indipendente del modello regredita sulle rimanenti covariate. Tanto più R^2 si avvicina a 1, tanto più la variabile considerata sarà spiegabile (e correlata) con le variabili rimanenti, e tanto più aumenterà il vif. A differenza della matrice di correlazione qui consideriamo la correlazione congiunta di tutte le variabili dipendenti (non solo la correlazione a coppie).

```
car::vif(mod_oxy)

##      age   weight  runtime rstpulse runpulse maxpulse
## 1.512836 1.155329 1.590868 1.415589 8.437274 8.743848
```

Ad esempio il vif dell'età (1.51) è calcolato regredendo l'età sulle rimanenti covariate del modello.
Anche qui si notano le problematicità di `runpulse` o `maxpulse`. VIF e matrice di correlazione vengono utilizzati assieme:

- dai vif può essere che non si capisca cosa è correlato con cosa (a parte questo caso dove sono le due variabili che hanno un vif elevato);
- tuttavia potrebbe evidenziare una variabile con vif elevato perchè correlata a diverse (es 0.4 con tutti i regressori).

2.6.4 Eliminazione variabile

Togliamo una tra maxpulse e runpulse per vedere cosa accade

```
no_max <- update(mod_oxy, . ~ . - maxpulse)
no_run <- update(mod_oxy, . ~ . - runpulse)
summary(no_max)

##
## Call:
## lm(formula = oxy ~ age + weight + runtime + rstpulse + runpulse,
##     data = fitness)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -5.125 -1.268  0.217  1.030  5.331
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 116.48761   11.61779  10.027 3.04e-10 ***
## age          -0.28528    0.10363  -2.753   0.0108 *
## weight       -0.05184    0.05773  -0.898   0.3777
## runtime      -2.70392    0.41211  -6.561 7.13e-07 ***
## rstpulse     -0.02711    0.07101  -0.382   0.7059
## runpulse     -0.12628    0.05231  -2.414   0.0234 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.493 on 25 degrees of freedom
## Multiple R-squared:  0.8176,Adjusted R-squared:  0.7811
## F-statistic: 22.41 on 5 and 25 DF,  p-value: 1.688e-08
```

```
summary(no_run)
```

```
##
## Call:
## lm(formula = oxy ~ age + weight + runtime + rstpulse + maxpulse,
##     data = fitness)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.0257 -1.5091  0.1441  1.3080  5.5761
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 110.15983   14.10169   7.812 3.62e-08 ***
## age          -0.25795    0.11500  -2.243    0.034 *
## weight       -0.04936    0.06252  -0.790    0.437
## runtime      -2.86147    0.43657  -6.554 7.25e-07 ***
## rstpulse     -0.04121    0.07612  -0.541    0.593
## maxpulse     -0.08153    0.06412  -1.272    0.215
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.683 on 25 degrees of freedom
## Multiple R-squared:  0.7887,Adjusted R-squared:  0.7464
## F-statistic: 18.66 on 5 and 25 DF,  p-value: 1.007e-07
```

In sintesi:

- togliendo `maxpulse`, lo standard error di `runpulse` si restringe parecchio;
- se facciamo la stessa cosa senza `runpulse`, `maxpulse` non è più positivo (ora è un coefficiente negativo). È un indice ancora più forte che quel-

le variabili erano correlate: cambia radicalmente l'interpretazione della variabile.

Non è immediata in questo ambito la scelta tra i due (che non sia motivata dalla significatività).

2.6.5 Scelta mediante procedura stepwise

```
mod_AIC <- MASS::stepAIC(mod)

## Error:  oggetto 'mod' non trovato
```

Con l'invocazione di sopra viene stampata la procedura stepwise che utiliza AIC come criterio decisionale; il modello risultante viene salvato in `mod_AIC`:

- La procedura parte dal modello completo, caratterizzato da un AIC di 58.16 e toglie a turno una variabile (la riga `none` è il caso in cui non si tolga niente). Prestando attenzione alla prima tabella, ad esempio se tolgo `weight` l'AIC del modello sarà 58.46, `runtime` 89.66.
- Visto che il criterio AIC migliora con un AIC inferiore, scelgo di togliere la variabile che mi conduce ad un AIC inferiore, ovvero tolgo `rstpulse`.
- Si giunge alla seconda iterazione (e seconda tabella); in questa qualsiasi mossa faccio (ad eccezione del non far nulla) ho un AIC peggiore. Quindi ho raggiunto il modello ottimale.

Questi criteri possono esser molto pericolosi se nel modello completo si hanno variabili molto correlate (in tal caso fanno disastri).

2.7 Amenità varie

2.7.1 Confondimento e paradosso di Simpson

Il dataset crimeFL di Agresti (esempio 11.1) contiene 67 contee della Florida

```
head(crimeFL)

##     county   c    i   hs    u
## 1  ALACHUA 104 22.1 82.7 73.2
## 2    BAKER  20 25.8 64.1 21.5
## 3      BAY  64 24.7 74.7 85.0
## 4 BRADFORD  50 24.6 65.0 23.2
## 5  BREVARD  64 30.5 82.3 91.9
## 6  BROWARD  94 30.6 76.8 98.9
```

Il nostro focus qui è su:

- `c` crime rate

- `hs`: percentuale di adulti con diploma di high school
- `u`: percentuale di urbanizzazione (persone che vivon in ambiente urbano)

siamo interessati alla spiegazione del tasso di criminalità in funzione delle rimanenti due

```
crimeFL <- crimeFL[c("c", "hs", "u")]
summary(crimeFL)
```

```
##        c               hs              u
##  Min.   :  0.0   Min.   :54.50   Min.   : 0.00
##  1st Qu.: 35.5   1st Qu.:62.45   1st Qu.:21.60
##  Median : 52.0   Median :69.00   Median :44.60
##  Mean   : 52.4   Mean   :69.49   Mean   :49.56
##  3rd Qu.: 69.0   3rd Qu.:76.90   3rd Qu.:83.55
##  Max.   :128.0   Max.   :84.90   Max.   :99.60
```

```
cor(crimeFL)
```

```
##            c        hs         u
## c  1.0000000 0.4669119 0.6773678
## hs 0.4669119 1.0000000 0.7907190
## u  0.6773678 0.7907190 1.0000000
```

```
## Partiamo da stime univariate
summary(mod_uni <- lm(c ~ hs, data = crimeFL))
```

```
##
## Call:
## lm(formula = c ~ hs, data = crimeFL)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -43.74 -21.36  -4.82  17.42  82.27
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -50.8569    24.4507  -2.080   0.0415 *
## hs            1.4860     0.3491   4.257 6.81e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25.12 on 65 degrees of freedom
## Multiple R-squared:  0.218,Adjusted R-squared:  0.206
## F-statistic: 18.12 on 1 and 65 DF,  p-value: 6.806e-05
```

Guardando ai coefficienti stimati Sembrerebbe che le contee che hanno maggior istruzione hanno anche tasso di criminalità più elevato: come dobbiamo interpretare? che studiare fa male?
Nella matrice con correlazione si nota come la scolarità sia molto correlato con l'urbanizzazione. Può essere allora che sia questo che l'analisi univariata ci stia nascondendo? Per farlo effettuiamo la stima multivariata

```
summary(mod_mult <- lm(c ~ hs + u, data = crimeFL))

##
## Call:
## lm(formula = c ~ hs + u, data = crimeFL)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -34.693 -15.742  -6.226  15.812  50.678
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  59.1181    28.3653   2.084   0.0411 *
## hs           -0.5834     0.4725  -1.235   0.2214
## u             0.6825     0.1232   5.539 6.11e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 20.82 on 64 degrees of freedom
## Multiple R-squared:  0.4714,Adjusted R-squared:  0.4549
## F-statistic: 28.54 on 2 and 64 DF,  p-value: 1.379e-09
```

L'interpretazione cambia radicalmente: controllando per urbanizzazione, l'effetto di scolarizzazione su crimine è negativa, mentre se non si fa ciò è positiva (dato che scolarizzazione e urbanizzazione sono correlati. Per vederlo graficamente plottiamo la relazione

```
## classi di urbanizzazione e colori associati
crimeFL$u_cl <- factor(ifelse(crimeFL$u < 33.3, 'low',
                      ifelse(crimeFL$u < 66.6, 'mid', 'high')),
                      levels = c('low', 'mid', 'high'))
crimeFL$col <- lbmisc::recode(as.character(crimeFL$u_cl),
                              c('low', 'green',
                                'mid', 'yellow',
                                'high', 'red'))

## plottiamo la relazione tra le due
plot(y = crimeFL$c, x = crimeFL$hs, col = crimeFL$col,
     xlab = 'High school %', ylab = 'Crime rate')
```

```
legend('topleft, legend = c('low urb', 'mid urb', 'high urb'),
       col = c('green', 'yellow', 'red'), pch = 1)

## plot della stima univariata, scorretta
abline(mod_uni)

## plot delle stime negli strati
crime_spl <- split(crimeFL, crimeFL$u_cl)
strata_mods <- lapply(crime_spl, function(x){
    lm(c ~ hs, data = x)
})
tmp <- Map(function(m, c) abline(m, col = c),
           strata_mods,
           list('green', 'yellow', 'red'))

## Error in parse(text = input):  <text>:14:30:  simbolo inatteso
## 13:       xlab = 'High school %', ylab = 'Crime rate')
## 14:  legend('topleft, legend = c('low
##                                  ^
```

Negli strati a media e alta urbanizzazione si nota una relazione negativa tra crime rate e high school, in quelli a bassa urbanizzazione una positiva; questo fa si che a livello complessivo, nella stima multivariata (dove si fa una media), la relazione sia negativa (anche se non statisticamente significativa).

Questo è un esempio (non perfetto) di paradosso di Simpson: se si considerano i dati nel loro complesso la relazione di due variabili è di un certo tipo, se però si stratifica all'interno degli strati è di tipo opposto (ma uguale nei due gruppi).

Il grafico suggerisce di tentare l'interazione per lasciare il coefficiente di high school variare in relazione ad urbanizzazione, ma non vi sono elementi per concludere che vi sia.

```
mod_int <- lm(c ~ hs * u, data = crimeFL)
anova(mod_mult, mod_int)

## Analysis of Variance Table
##
## Model 1: c ~ hs + u
## Model 2: c ~ hs * u
##   Res.Df   RSS Df Sum of Sq      F Pr(>F)
## 1     64 27730
## 2     63 27324  1    406.41 0.9371 0.3367
```

2.7.2 Partial regression stuff

Supponiamo ancora di voler analizzare la correlazione tra criminalità e istruzione, tenendo però in considerazione una terza variabile (ovvero al netto della sua influenza).
Potremmo plottare criminalità vs istruzione, per ogni valore (se discreto, range di valori omogenei se continuo) della terza variabile:

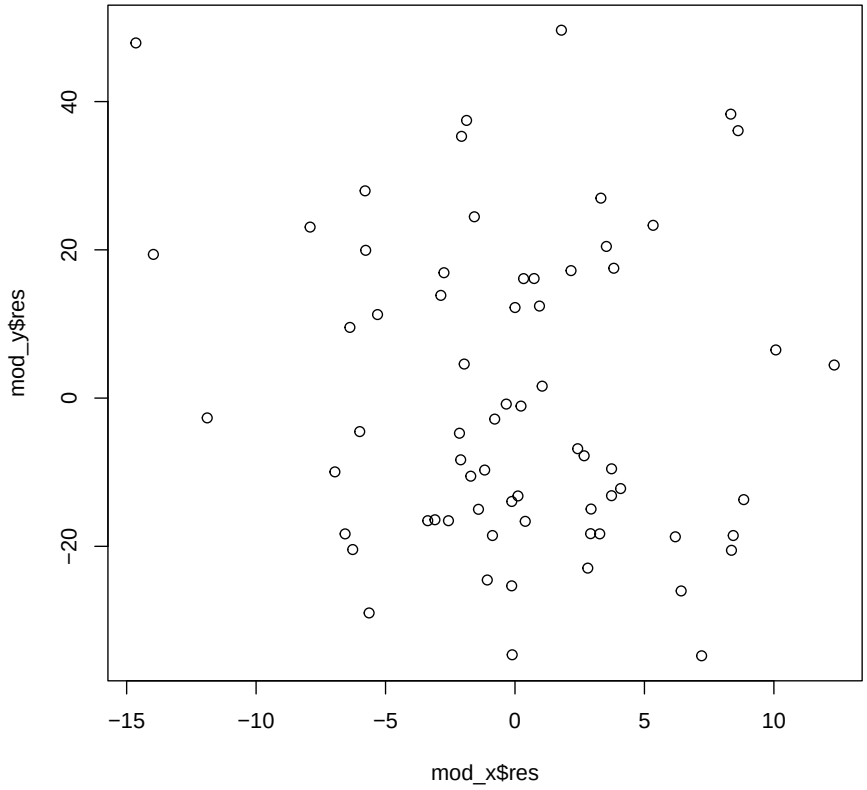
- se la terza variabile è quantitativa la stratificazione è discrezionale
- se volessimo aumentare il numero di altre variabili che vogliamo tener costanti, lo stratificare ad oltranza potrebbe esser scomodo e ridurre il confronto a poche unità, che effettivamente sono uguali su tutte le covariate.

Un modo alternativo e più pratico è esemplificato utilizzando urbanizzazione stimare i modelli:

- $c = f(u)$: il residuo di questo modello è quanto rimane da spiegare della criminalità una volta che si controlla per urbanizzazione
- $hs = f(u)$: il residuo di questo modello è la variabilità rimane di istruzione, oltre a quella spiegata da urbanizzazione
- nel caso di molteplici variabili di controllo $(x_2, \ldots, x_n)$, i modelli diventano semplicemente $y = f(x_2, \ldots, x_n)$ e $x_1 = f(x_2, \ldots, x_n)$

Possiamo calcolare plot di questi residui (partial regression plot) che mostra la relazione tra criminalità e istruzione depurato da urbanizzazione. Si vede come il coefficiente della regressione multipla è il coefficiente della semplice, depurato dall'effetto delle altre variabili dipendenti.

```
mod_y <- lm(c ~ u, data = crimeFL)
mod_x <- lm(hs ~ u, data = crimeFL)
# Partial regression plot, correlazione semplice vs correlazione parziale
plot(y = mod_y$res, x = mod_x$res)
```




```
cor(crimeFL$c, crimeFL$hs)

## [1] 0.4669119

cor(mod_y$res, mod_x$res)

## [1] -0.1525397

# Il coefficiente di regressione corrisponde a quello ottenuto nella stima
# multivariata che controlla per u
lm(mod_y$res ~ mod_x$res)

##
## Call:
## lm(formula = mod_y$res ~ mod_x$res)
##
## Coefficients:
```

```
## (Intercept)    mod_x$res
##  -9.947e-16   -5.834e-01

lm(c ~ hs + u, data = crimeFL)

##
## Call:
## lm(formula = c ~ hs + u, data = crimeFL)
##
## Coefficients:
## (Intercept)           hs            u
##     59.1181      -0.5834       0.6825
```

2.7.3 Coefficienti standardizzati e loro confronto

Possiamo essere interessati a valuare se l'influsso di due covariate quantitative sulla spiegata sia uguale o differente; per farlo occorre stimare un modello con coefficienti standardizzati (al fine di evitare che le unità di misura originali influiscano) e confrontare gli stessi con un test sulle restrizioni lineari, effettuato mediante `car::linearHypothesis`.

```
library(car)
mod_duncan_sc <- lm(prestige ~ scale(income) + scale(education), data = Duncan)
summary(mod_duncan_sc)

##
## Call:
## lm(formula = prestige ~ scale(income) + scale(education), data = Duncan)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.538  -6.417   0.655   6.605  34.641
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)        47.689      1.993  23.929  < 2e-16 ***
## scale(income)      14.630      2.924   5.003 1.05e-05 ***
## scale(education)   16.244      2.924   5.555 1.73e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.37 on 42 degrees of freedom
## Multiple R-squared:  0.8282,Adjusted R-squared:   0.82
## F-statistic: 101.2 on 2 and 42 DF,  p-value: < 2.2e-16

car::linearHypothesis(mod_duncan_sc, "scale(income) = scale(education)")
```

```
##
## Linear hypothesis test:
## scale(income) - scale(education) = 0
##
## Model 1: restricted model
## Model 2: prestige ~ scale(income) + scale(education)
##
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     43 7522.5
## 2     42 7506.7  1    15.796 0.0884 0.7677
```

Le restrizioni vengono utilizzate come ipotesi nulla (per creare il modello ristretto): in questo esempio il modello non viene rifiutato dal test, quindi reddito ed educazione hanno approssimativamente lo stesso influsso sul prestigio. Volendo provare a riprodurre il calcolo (seguendo ad esempio Wooldridge), il modello ristretto è:

$$prestige = \beta_0 + \beta_1(scale(income) + scale(education))$$

mentre il test si implementa come

$$F = \frac{(RSS_r - RSS_{ur})/q}{RSS_{ur}/(n - k_{ur} - 1)} \sim F_{q,n-k_{ur}-1}$$

dove:

- il pedice r fa riferimento al modello *restricted*
- il pedice ur fa riferimento al modello *unrestricted* (ossia l'originale)
- q è il numero di restrizioni lineari (in questo caso 1)

```
mod_restr <- lm(prestige ~ I(scale(income) + scale(education)), data = Duncan)
RSS_r <- sum(residuals(mod_restr)^2)
RSS_ur <- sum(residuals(mod_duncan_sc)^2)
df1 <- q <- 1 # una restrizione lineare
df2 <- nrow(Duncan) - 2 - 1
f <- ((RSS_r - RSS_ur)/df1) /  (RSS_ur/df2)
pf(q = f, df1 = df1, df2 = df2, lower.tail = FALSE)

## [1] 0.7677152
```

3 Diagnostica

Complessivamente

- condizionatamente ai valori di x (stratificando) il termine di *errore* deve:
  1. avere varianza omogenea (omoschedasticità)
  2. esser normale
  3. esser indipendente

  L'analisi dei *residui* della regressione può aiutare soprattutto sui primi due
- oltre a questo vanno cercate osservazioni inusuali/strane;

```
par(mfrow = c(2,2))
plot(mod_oxy)
```

3.1 Analisi residui

Procediamo sul modello di ossigeno/fitness con i quattro grafici di default forniti da R per l'analisi dei residui in figura 1. Nell'ordine:

1. il primo grafico mostra i residui sull'asse delle ordinate contro i valori fittati; i fa residui verso fittati e non residui verso variabile dipendente perché i residui sono per costruzione non correlati con la previsione mentre potrebbero invece esser correlati con la variabile dipendente.
   Desidero che se faccio una regressione in questa nuvola risulti una retta con pendenza nulla e che in particolare all'aumentare della x l'altezza dello scatter non aumenti, perché in tal caso ci si troverebbe in una situazione di eteroschedasticità. In tal caso i p-value delle stime che avremmo non sarebbero corretti.
   Un test quick'n dirty proposto da Faraway: se una retta di regressione dei residui assoluti in funzione dei valori fittati restituisce un beta1 nullo, allora si può confermare non vi sia eteroschedasticità. Un modello quadratico può esser utile per trovare degli yoyo o delle trottole. Bisogna in entrambi i casi guardare all'F-test e in questo caso non sembrano esserci grossi problemi

   ```
   r <- abs(residuals(mod_oxy))
   f <- fitted(mod_oxy)
   res_n <- lm(r ~ 1)
   res_f <- lm(r ~ f + I(f^2))
   anova(res_n, res_f)

   ## Analysis of Variance Table
   ##
   ## Model 1: r ~ 1
   ## Model 2: r ~ f + I(f^2)
   ```

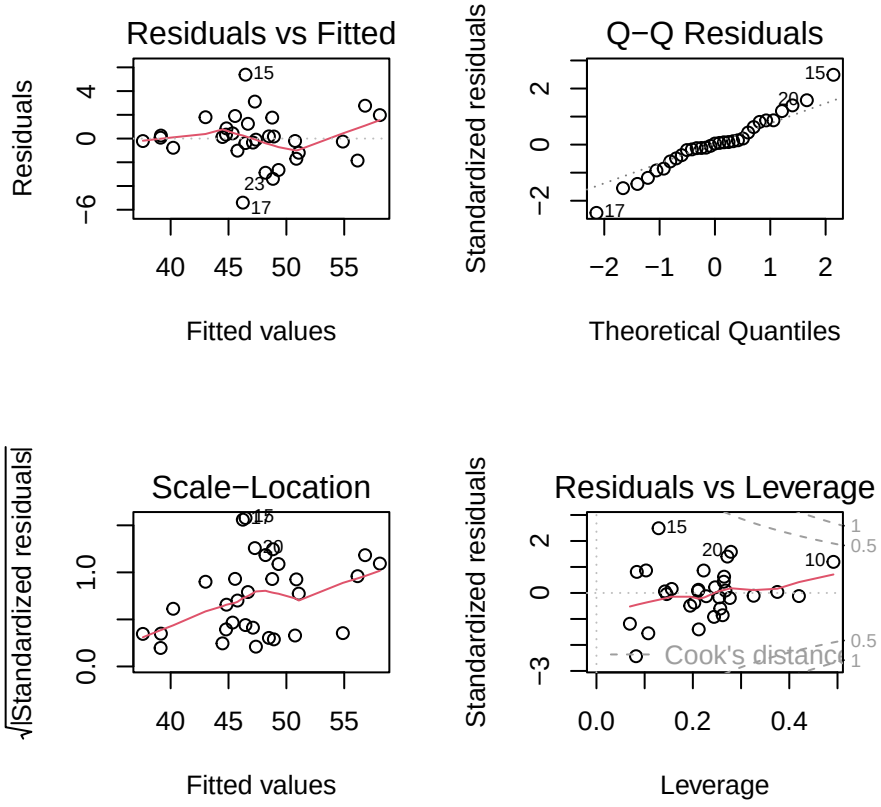


Figura 1: Plot residui

```
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1     30 64.243
## 2     28 58.642  2    5.6012 1.3372 0.2788
```

2. il normal qqplot (grafico in alto a destra) serve per valutare che i residui siano normalmente distribuiti; tanto più i residui stanno sulla retta tratteggiata, tanto più l'assunto è verificato.
   Il grafico sembra rappresentare all'incirca una retta supportando anche se nelle code sembrano concentrarsi i problemi maggiori. Un pattern a forma di S o di banana avrebbe fatto pensare ad una forma distributiva diversa da quella normale ipotizzata.
   Il normal qqplot viene guardato assieme al test di Shapiro-Wilk

   ```
   shapiro.test(residuals(mod_oxy))

   ##
   ##  Shapiro-Wilk normality test
   ##
   ## data:  residuals(mod_oxy)
   ## W = 0.96779, p-value = 0.4603
   ```

   il test si basa basa sull'ipotesi nulla che i residui abbiano una distribuzione normale (e in questo caso non la rifiutiamo). Va guardato assieme al qqplot:per campioni grandi potrebbe essere troppo sensibile (significativo sempre) e per campioni piccoli potrebbe aver poca potenza.
   Se l'errore non è normale test e intervalli di confidenza potrebbero non esser corretti: le distribuzioni con lunghe code nello specifico causano inaccuratezze, mentre la non normalità intermedia può essere ignorata. In caso di non normalità dell'errore può essere utile provare trasformazioni o adottare stimatori robusti.

3. I residui standardizzati (divisi per la loro deviazione standard) possono esser utilizzati per valutare la presenza di *outlier* (grafico in basso a sinistra). Se sono normalmente distribuiti, mi piacerebbe che stessero nel range $(-2, 2)$. In questo caso poche cose superano il $|2|$: attenzione che nel grafico viene plottato $\sqrt{|residuo|}$, quindi non deve andare oltre 1.4. In questa situazione ci sono solo 2 osservazioni, il che va bene, (corrispondono circa al 6%).

4. Il quarto ed ultimo grafico mostra i residui standardizzati come funzione del leverage (in asse x) e la distanza di Cook (curve di livello tratteggiate). Lo scopo di questo grafico è quello di evidenziare le osservazioni molto diversa dalle altre che hanno un'influenza maggiore nel calcolo delle stime.

Possiamo provare a togliere dal dataframe le osservazioni 10, 15 e 20: si osserva che sia stime che relativi standard error rimangono pressoché gli stessi (ovvia-

mente abbiamo perso tre gradi di libertà). Quindi non sono molto sensibili ai valori (pseudo) outlier.

```
# Per confronto il vecchio modello ..
summary(mod_oxy)

##
## Call:
## lm(formula = oxy ~ ., data = fitness)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.4026 -0.8991  0.0706  1.0496  5.3847
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 102.93448   12.40326   8.299 1.64e-08 ***
## age          -0.22697    0.09984  -2.273  0.03224 *
## weight       -0.07418    0.05459  -1.359  0.18687
## runtime      -2.62865    0.38456  -6.835 4.54e-07 ***
## rstpulse     -0.02153    0.06605  -0.326  0.74725
## runpulse     -0.36963    0.11985  -3.084  0.00508 **
## maxpulse      0.30322    0.13650   2.221  0.03601 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.317 on 24 degrees of freedom
## Multiple R-squared:  0.8487,Adjusted R-squared:  0.8108
## F-statistic: 22.43 on 6 and 24 DF,  p-value: 9.715e-09

# e il nuovo
fitness2 <- fitness[-c(10, 15, 20), ]
summary(mod_oxy2 <- lm(oxy ~ . , data = fitness2))

##
## Call:
## lm(formula = oxy ~ ., data = fitness2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.9407 -0.9089  0.3082  0.9311  2.7368
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 107.69268   10.47799  10.278 1.20e-09 ***
## age          -0.29586    0.08758  -3.378  0.00284 **
```

```
## weight       -0.10266    0.04627  -2.219  0.03766 *
## runtime      -2.29987    0.33530  -6.859 8.84e-07 ***
## rstpulse     -0.06778    0.06204  -1.093  0.28698
## runpulse     -0.35335    0.12717  -2.779  0.01126 *
## maxpulse      0.28332    0.14998   1.889  0.07278 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.913 on 21 degrees of freedom
## Multiple R-squared:  0.8812,Adjusted R-squared:  0.8473
## F-statistic: 25.97 on 6 and 21 DF,  p-value: 1.099e-08

car::vif(mod_oxy2)

##       age    weight   runtime  rstpulse  runpulse  maxpulse
##  1.424018  1.177050  1.638979  1.594397 13.863361 14.277482
```

3.2 Residui vs covariate

In figura 2 può essere utile per individuare le forme funzionali (es quadratiche etc) con le quali inserire una covariata

```
par(mfrow = c(3, 2))
fitness$res <- residuals(mod_oxy)
ylab <- 'Residui'
with(fitness, {
    plot(age, res, ylab = ylab, xlab = 'age')
    plot(weight, res, ylab = ylab, xlab = 'weight')
    plot(runtime, res, ylab = ylab, xlab = 'runtime')
    plot(rstpulse, res, ylab = ylab, xlab = 'rstpulse')
    plot(runpulse, res, ylab = ylab, xlab = 'runpulse')
    plot(maxpulse, res, ylab = ylab, xlab = 'maxpulse')
})
```

TODO: influence(mod) e spiegazione

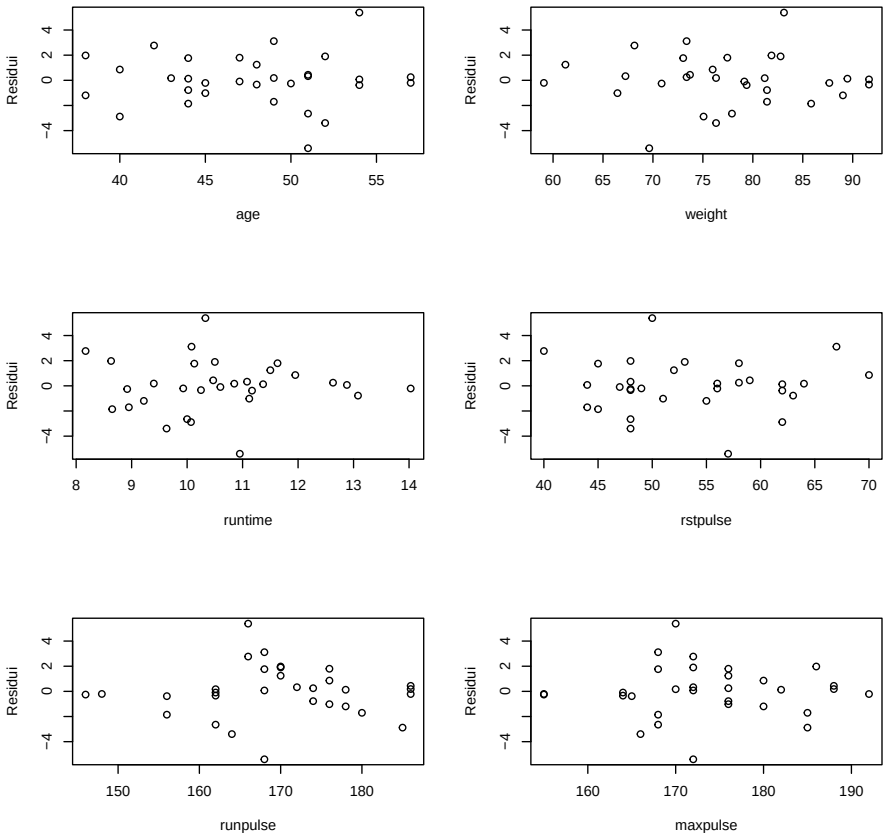


Figura 2: Residui vs covariate