

***Cluster Analysis:***  
**Metodi non gerarchici**  
**Analisi Esplorativa**

Aldo Solari



① Cluster Analysis

② Metodo delle  $K$ -medie



# Outline

## ① Cluster Analysis

## ② Metodo delle $K$ -medie



# Perchè raggruppare

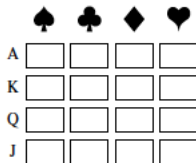
- Suddividere le unità in gruppi è un modo naturale e, si può dire, imprescindibile, di ragionare per comprendere i fenomeni
- Si ragiona per gruppi perchè è più facile dominare mentalmente pochi gruppi che tante unità
- Uno stesso insieme di unità consente diversi raggruppamenti, nessuno è 'giusto', semmai può essere utile (o inutile o anche dannoso)
- Raggruppare utilmente: mettere insieme unità simili e separare unità dissimili, in altre parole creare gruppi
  - omogenei al loro interno (*internal cohesion*)
  - disomogenei tra di loro (*external isolation*)



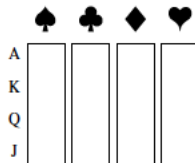
# Cluster analysis

- Nell'analisi di raggruppamento (o *cluster analysis*) la domanda cui si vuol rispondere è se esistono e quanti sono dei gruppi sensati (naturali) in cui suddividere le unità sulla base delle variabili osservate
- Se si ha una conoscenza approfondita del fenomeno in esame, si è in grado di distinguere tra 'buoni' raggruppamenti e 'cattivi' raggruppamenti
- Perchè non semplicemente considerare tutti i possibili raggruppamenti (*partizioni* di  $n$  unità in  $K$  gruppi) e sceglierne il 'migliore'?

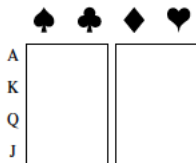




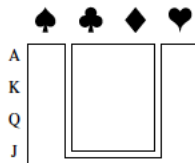
(a) Individual cards



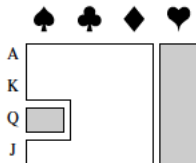
(b) Individual suits



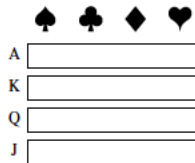
(c) Black and red suits



(d) Major and minor suits (bridge)



(e) Hearts plus queen of spades and other suits (hearts)



(f) Like face cards



# Numero di partizioni possibili

Per l'esempio delle  $n = 16$  carte:

- 1 modo di formare 1 singolo gruppo  $K = 1$
- 32767 modi di formare 2 gruppi  $K = 2$
- 7141686 modi di formare 3 gruppi  $K = 3$
- etc.
- 1 modo di formare  $n$  gruppi  $K = n = 16$

per un totale di 10480142147 partizioni possibili



# Numero di partizioni possibili

Il numero di tutte le possibili partizioni di  $n$  unità in  $K$  gruppi è

$$S(n, K) = \frac{1}{K!} \sum_{k=0}^K (-1)^{K-k} \binom{K}{k} k^n$$

dove  $S(n, K)$  è il numero di Stirling (di seconda specie), quindi si dovrebbero considerare un numero di possibilità pari all'  $n$ -esimo numero di Bell

$$B_n = \sum_{K=1}^n S(n, K)$$

Data la scarsa percorribilità dell'esplorazione di tutte le possibili partizioni, si procede usando dei metodi (algoritmi) che non esplorano l'intero spazio di tutte le possibili partizioni ma solo una parte di esse: non v'è perciò garanzia di ottenere la soluzione ottima in senso assoluto.



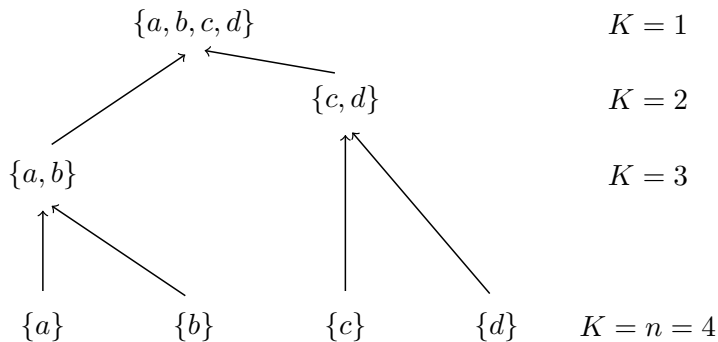


# Metodi (algoritmi) gerarchici e non

- Nei *metodi gerarchici* si individua una sequenza di partizioni nidificate: la partizione in  $K + 1$  gruppi si ottiene dalla partizione in  $K$  gruppi facendo di due degli elementi di questa un elemento di quella (AGNES), o viceversa (DIANA)
  - Algoritmo Agglomerativo (AGNES, AGGlomerative NESTing)
  - Algoritmo Scissorio (DIANA, DIvisive ANAlysis)
- Nei *metodi non gerarchici*: il numero di gruppi  $K$  è deciso a priori
  - Metodo delle  $K$ -medie



# AGNES



# Outline

① Cluster Analysis

② Metodo delle  $K$ -medie



# Scomposizione della distanza totale

- Fissato a priori il numero dei gruppi  $K$ , come possiamo scegliere i gruppi

$$G_1, \dots, G_K$$

in maniera 'ottimale'?

- La *distanza totale* risulta

$$T = \frac{1}{2n} \sum_{i=1}^n \sum_{l=1}^n d^2(u_i, u_l)$$

dove  $d^2(u_i, u_l)$  è la distanza Euclidea al quadrato tra due unità

- Possiamo scomporre la distanza totale  $T$  in

$$T = W + B$$

dove

- $B$  è la distanza tra i gruppi (*between*)
- $W$  è la distanza entro i gruppi (*within*)



# Distanza entro i gruppi

- La distanza entro i gruppi si può esprimere come

$$W = \sum_{k=1}^K W(G_k)$$

dove

$$W(G_k) = \frac{1}{2n_k} \sum_{i: u_i \in G_k} \sum_{l: u_l \in G_k} d^2(u_i, u_l)$$

è la distanza entro il  $k$ -simo gruppo, e  $n_k$  è la numerosità del gruppo  $G_k$



# Problema di minimo

- Vogliamo determinare i gruppi  $G_1^*, \dots, G_K^*$  tali che

$$W^* \leq W$$

ovvero risolvere il problema di minimo

$$G_1^*, \dots, G_K^* = \arg \min_{G_1, \dots, G_K} W$$

- Si noti che determinare  $G_1^*, \dots, G_K^*$  che minimizza  $W$  comporta anche la massimizzazione di  $B$  poichè  $T$  è costante (non dipende dai  $G_1, \dots, G_K$ )

$$T = W^* + B^*$$



## Esempio con $K = 2$ , $n = 3$ e $p = 2$

$$u'_1 = (0, 0), u'_2 = (0, 3), u_3 = (4, 3)$$

$$\text{Baricentro } \bar{x}'_{1 \times 2} = (4/3, 2)$$

$$d^2(u_1, u_2) = 9, d^2(u_1, u_3) = 25, d^2(u_2, u_3) = 16$$

$G_1, G_2$	$W$	$B$	$T$
$\{1\}, \{2,3\}$	8	8.6	16.6
<b><math>\{1,2\}, \{3\}</math></b>	4.5	12.1	16.6
$\{1,3\}, \{2\}$	12.5	4.1	16.6

Tuttavia, in generale, data la scarsa percorribilità dell'esplorazione di tutte le possibili partizioni, l'algoritmo delle  $K$ -medie non esplora l'intero spazio di tutte le possibili partizioni ma solo una parte di esse



# Centroidi

- La distanza totale (in R, totss) può essere espressa come

$$T = \sum_{i=1}^n \sum_{j=1}^p (x_{ij} - \bar{x}_j)^2$$

dove  $\bar{x}_j$  è il  $j$ -simo elemento del vettore delle medie  $\bar{x}$ .

- La distanza entro il  $k$ -simo gruppo può essere espressa come

$$W(G_k) = \sum_{i:u_i \in G_k} d^2(u_i, \bar{x}_k) = \left[ \sum_{i:u_i \in G_k} \sum_{j=1}^p (x_{ij} - \bar{x}_{kj})^2 \right]$$

dove  $\bar{x}_{kj}$  è il  $j$ -simo elemento del  $k$ -simo *centroide* (vettore delle medie del gruppo  $G_k$ ).

$$\bar{x}_k = \begin{bmatrix} \bar{x}_{k1} \\ \vdots \\ \bar{x}_{kp} \end{bmatrix} = \begin{bmatrix} \frac{1}{n_k} \sum_{i:u_i \in G_k} x_{i1} \\ \vdots \\ \frac{1}{n_k} \sum_{i:u_i \in G_k} x_{ip} \end{bmatrix}$$





# Minimo locale

Per minimizzare la distanza entro i gruppi (in R, `tot.withinss`)

$$W = \sum_{k=1}^K W(C_k) = \sum_{k=1}^K \left[ \sum_{i: u_i \in G_k} \sum_{j=1}^p (x_{ij} - \bar{x}_{kj})^2 \right]$$

bisogna minimizzare *congiuntamente*

- rispetto ai gruppi  $G_1, \dots, G_K$
- ai centri  $\bar{x}_1, \dots, \bar{x}_K$

L'algoritmo delle  $K$  medie minimizza *localmente* la quantità sopra indicata (non è garantito il minimo globale) minimizzando *in alternanza* rispetto ai gruppi e rispetto ai centri



# Algoritmo delle $K$ -medie

- ① Si parte con una attribuzione iniziale per  $\bar{x}_1, \dots, \bar{x}_k$  (e.g. considerando  $K$  unità statistiche).

Si procede iterando ② e ③ fino alla convergenza:

- ② Minimizzazione rispetto ai gruppi:

per  $i = 1, \dots, n$ , si individua il centroide più vicino (secondo  $d^2$ ) all'unità  $u_i$  e la si attribuisce al gruppo corrispondente  $G_k$

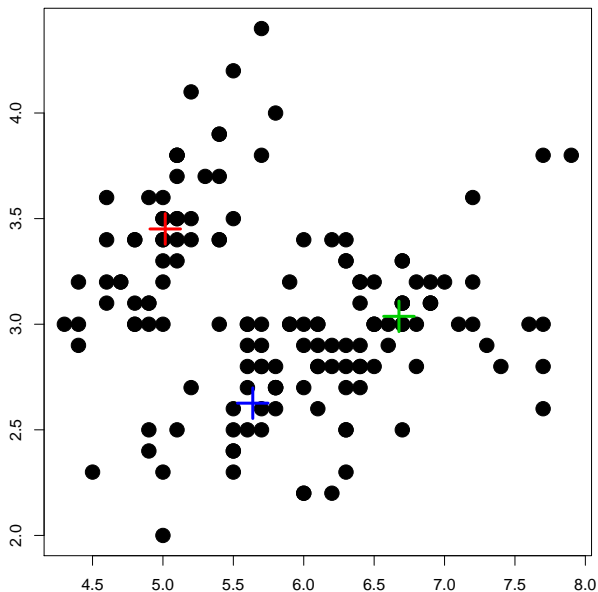
- ③ Minimizzazione rispetto ai centroidi:

per  $k = 1, \dots, K$ , si aggiorna il valore del  $k$ -simo centroide con la media delle unità del gruppo  $G_k$ . Calcolare  $W$

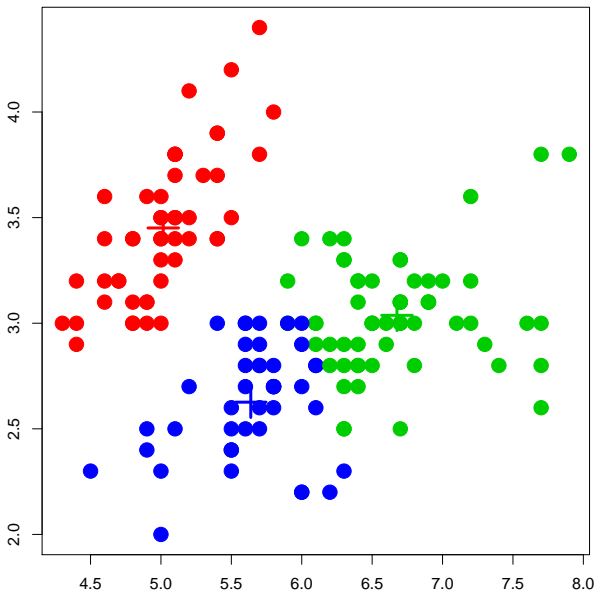
Si arresta l'algoritmo quando  $W$  non cambia rispetto al passo precedente (convergenza)



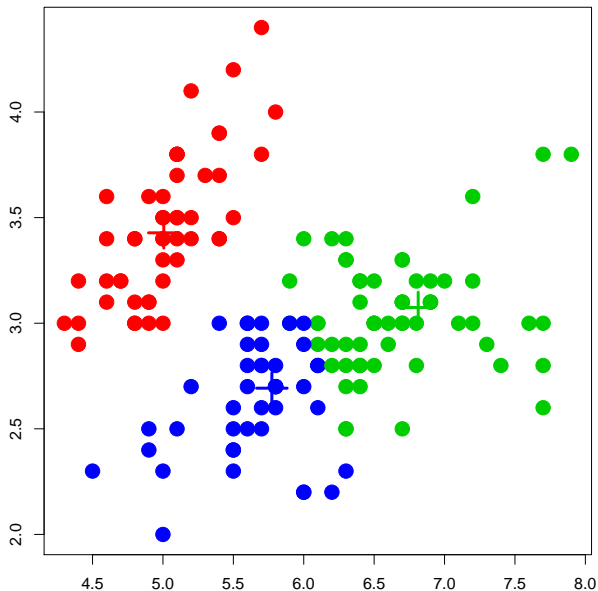
# ① Inizializzo i centri di



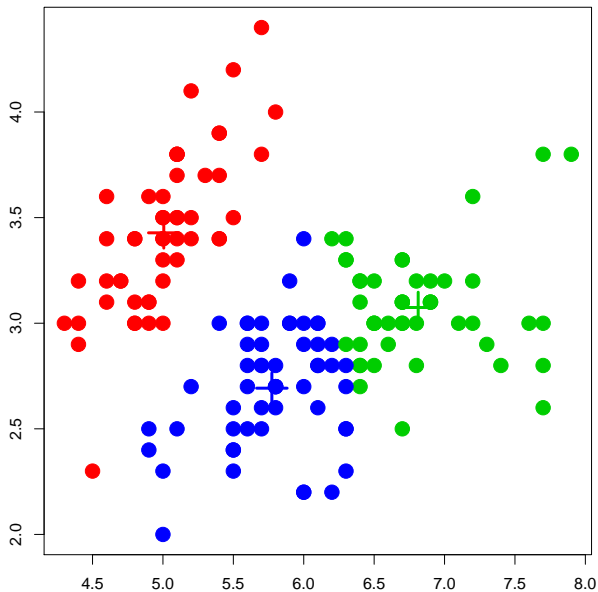
## ② Attribuzione unità ai gruppi



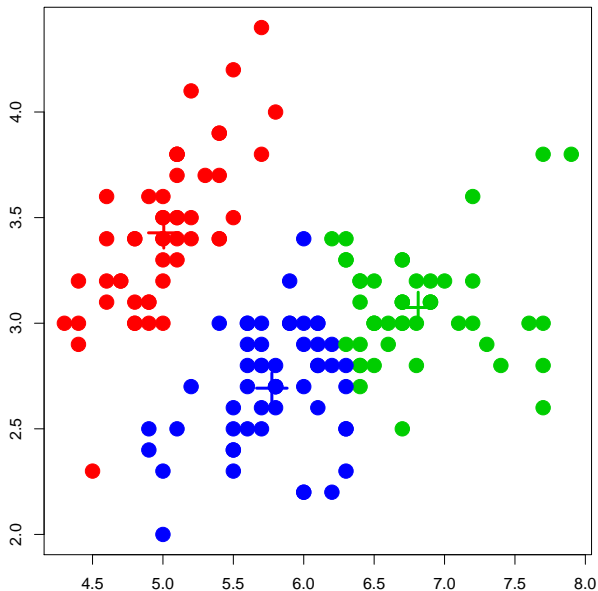
### ③ Aggiorno i centroidi



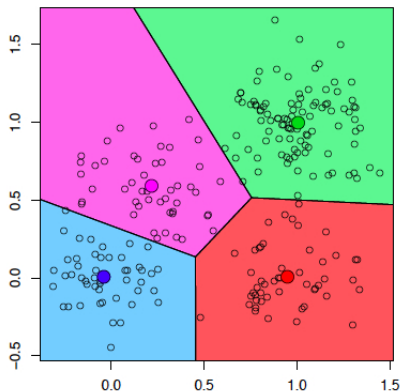
## ② iterazione 1



### ③ iterazione 1: STOP



# Tassellazione di Voronoi



L'algoritmo definisce una tassellazione di Voronoi in  $\mathbb{R}^p$

$$V_k = \{x \in \mathbb{R}^p : d^2(x - \bar{x}_k) \leq d^2(x - \bar{x}_h), h = 1, \dots, K\}$$

che sono poliedri convessi





# Proprietà dell'algoritmo delle K-medie

- $W$  decresce ad ogni iterazione dell'algoritmo:  $W_{i+1} \leq W_i$ , dove  $W_i$  è  $W$  all'iterazione  $i$ -sima
- L'algoritmo converge sempre, indipendentemente dall'attribuzione iniziale dei centroidi.  
Ci mette  $\leq K^n$  iterazioni
- I gruppi finali dipendono dall'attribuzione iniziale dei centroidi.  
Tipicamente si fa girare l'algoritmo più volte inizializzando i centroidi casualmente, e si sceglie il risultato con  $W$  minimo
- L'algoritmo non garantisce di minimizzare globalmente  $W$



# Ripetere più volte K-medie



# Indicazioni sul metodo delle K-medie

- Adatto a scoprire gruppi di forma convessa
- Inadatto per gruppi di forma concava;
- Il risultato è sensibile alla presenza di valori anomali
- Non è invariante a trasformazioni di scala



# Algoritmo di Lloyd

L'algoritmo K-medie è spesso chiamato algoritmo di Lloyd in computer science e ingegneria, e viene utilizzato per la compressione di immagini (*vector quantization*)



Immagine originale, compressione 23.9%, compressione 6.25%



# Quanti gruppi? L'indice CH

- Determinare il numero  $K$  di gruppi è un problema molto difficile!
- Una possibilità è calcolare l'indice CH (Calinski and Harabasz, 1974)

$$\text{CH}(K) = \frac{B(K)/(K-1)}{W(K)/(n-K)}$$

per  $K$  che va da 2 a un pre-fissato  $K_{\max}$  e si sceglie

$$\hat{K} = \arg \max_{K \in \{2, \dots, K_{\max}\}} \text{CH}(K)$$



# La *silhouette*

- Determinato, in qualunque modo (non solo con il metodo delle  $K$ -medie), un raggruppamento di  $n$  unità in  $K$  gruppi  $G_1, \dots, G_K$  la *silhouette* è uno strumento per verificare la 'bontà' (coesione interna e separazione esterna) di tale raggruppamento
- Si confronta, per ciascuna osservazione, quanto essa sia vicina al suo gruppo e agli altri.
- La distanza dell'osservazione  $u_i^*$  dal gruppo  $G_k$  è definita come

$$d(u_i^*, G_k) = \frac{1}{n_k} \sum_{l: u_l \in G_k} d(u_i^*, u_l).$$



# La silhouette

- Sia poi  $G_{k^*}$  il gruppo in cui è inclusa l'osservazione  $u_i^*$  e sia

$$d_0 = \min_{k \neq k^*} d(u_i^*, G_k),$$

$d_0$  è la distanza di  $u_i^*$  dal gruppo più vicino diverso da quello cui appartiene

- Si confronta  $d_0$  con la distanza dal suo gruppo mediante

$$S(u_{i^*}) = \frac{d_0 - d(u_{i^*}, G_{k^*})}{\max\{d_0, d(u_{i^*}, G_{k^*})\}}.$$

- $S(u_{i^*}) \leq 1$
- $S(u_{i^*})$  è tanto più grande quanto più  $u_{i^*}$  è vicino al suo gruppo e distante dagli altri gruppi.
- $S(u_{i^*}) < 0$  indica che  $u_{i^*}$  è più vicino a un altro gruppo che non al suo.



# Forza lavoro nei paesi europei

Si considerano le composizioni della forza lavoro per settore produttivo negli stati europei nel 1970

Country	Agr	Min	Man	Pow	Con	Ser	Fin	SPS	TC	blocco
Belgium	3.3	0.9	27.6	0.9	8.2	19.1	6.2	26.6	7.2	w
Denmark	9.2	0.1	21.8	0.6	8.3	14.6	6.5	32.2	7.1	w
France	10.8	0.8	27.5	0.9	8.9	16.8	6.0	22.6	5.7	w
W. Germany	6.7	1.3	35.8	0.9	7.3	14.4	5.0	22.3	6.1	w
Ireland	23.2	1.0	20.7	1.3	7.5	16.8	2.8	20.8	6.1	w
Italy	15.9	0.6	27.6	0.5	10.0	18.1	1.6	20.1	5.7	w
Luxembourg	7.7	3.1	30.8	0.8	9.2	18.5	4.6	19.2	6.2	w
Netherlands	6.3	0.1	22.5	1.0	9.9	18.0	6.8	28.5	6.8	w
United Kingdom	2.7	1.4	30.2	1.4	6.9	16.9	5.7	28.3	6.4	w
Austria	12.7	1.1	30.2	1.4	9.0	16.8	4.9	16.8	7.0	w
Finland	13.0	0.4	25.9	1.3	7.4	14.7	5.5	24.3	7.6	w
Greece	41.4	0.6	17.6	0.6	8.1	11.5	2.4	11.0	6.7	w
Norway	9.0	0.5	22.4	0.8	8.6	16.9	4.7	27.6	9.4	w
Portugal	27.8	0.3	24.5	0.6	8.4	13.3	2.7	16.7	5.7	w
Spain	22.9	0.8	28.5	0.7	11.5	9.7	8.5	11.8	5.5	w
Sweden	6.1	0.4	25.9	0.8	7.2	14.4	6.0	32.4	6.8	w
Switzerland	7.7	0.2	37.8	0.8	9.5	17.5	5.3	15.4	5.7	n
Turkey	66.8	0.7	7.9	0.1	2.8	5.2	1.1	11.9	3.2	n
Bulgaria	23.6	1.9	32.3	0.6	7.9	8.0	0.7	18.2	6.7	e
Czechoslovakia	16.5	2.9	35.5	1.2	8.7	9.2	0.9	17.9	7.0	e
E. Germany	4.2	2.9	41.2	1.3	7.6	11.2	1.2	22.1	8.4	e
Hungary	21.7	3.1	29.6	1.9	8.2	9.4	0.9	17.2	8.0	e
Poland	31.1	2.5	25.7	0.9	8.4	7.5	0.9	16.1	6.9	e
Rumania	34.7	2.1	30.1	0.6	8.7	5.9	1.3	11.7	5.0	e
USSR	23.7	1.4	25.8	0.6	9.2	6.1	0.5	23.6	9.3	e
Yugoslavia	48.7	1.5	16.8	1.1	4.9	6.4	11.3	5.3	4.0	n





# Forza lavoro nei paesi europei

Consideriamo una suddivisione in 3 gruppi, basata sul metodo delle  $K$ -medie

I tre gruppi che si ottengono hanno centroidi

	Agr	Min	Man	Pow	Con	Ser	Fin	SPS	TC
1	52.30	0.93	14.10	0.60	5.27	7.70	4.93	9.40	4.63
2	25.02	1.78	28.08	0.93	8.72	9.54	2.13	17.11	6.69
3	8.24	0.99	29.09	0.96	8.43	16.28	5.00	24.17	6.86



# Forza lavoro nei paesi europei

	Agr	Min	Man	Pow	Con	Ser	Fin	SPS	TC	blocco
Greece	41.40	0.60	17.60	0.60	8.10	11.50	2.40	11.00	6.70	w
Turkey	66.80	0.70	7.90	0.10	2.80	5.20	1.10	11.90	3.20	n
Yugoslavia	48.70	1.50	16.80	1.10	4.90	6.40	11.30	5.30	4.00	n
Ireland	23.20	1.00	20.70	1.30	7.50	16.80	2.80	20.80	6.10	w
Portugal	27.80	0.30	24.50	0.60	8.40	13.30	2.70	16.70	5.70	w
Spain	22.90	0.80	28.50	0.70	11.50	9.70	8.50	11.80	5.50	w
Bulgaria	23.60	1.90	32.30	0.60	7.90	8.00	0.70	18.20	6.70	e
Czechoslovakia	16.50	2.90	35.50	1.20	8.70	9.20	0.90	17.90	7.00	e
Hungary	21.70	3.10	29.60	1.90	8.20	9.40	0.90	17.20	8.00	e
Poland	31.10	2.50	25.70	0.90	8.40	7.50	0.90	16.10	6.90	e
Rumania	34.70	2.10	30.10	0.60	8.70	5.90	1.30	11.70	5.00	e
USSR	23.70	1.40	25.80	0.60	9.20	6.10	0.50	23.60	9.30	e
Belgium	3.30	0.90	27.60	0.90	8.20	19.10	6.20	26.60	7.20	w
Denmark	9.20	0.10	21.80	0.60	8.30	14.60	6.50	32.20	7.10	w
France	10.80	0.80	27.50	0.90	8.90	16.80	6.00	22.60	5.70	w
W. Germany	6.70	1.30	35.80	0.90	7.30	14.40	5.00	22.30	6.10	w
Italy	15.90	0.60	27.60	0.50	10.00	18.10	1.60	20.10	5.70	w
Luxembourg	7.70	3.10	30.80	0.80	9.20	18.50	4.60	19.20	6.20	w
Netherlands	6.30	0.10	22.50	1.00	9.90	18.00	6.80	28.50	6.80	w
United Kingdom	2.70	1.40	30.20	1.40	6.90	16.90	5.70	28.30	6.40	w
Austria	12.70	1.10	30.20	1.40	9.00	16.80	4.90	16.80	7.00	w
Finland	13.00	0.40	25.90	1.30	7.40	14.70	5.50	24.30	7.60	w
Norway	9.00	0.50	22.40	0.80	8.60	16.90	4.70	27.60	9.40	w
Sweden	6.10	0.40	25.90	0.80	7.20	14.40	6.00	32.40	6.80	w
Switzerland	7.70	0.20	37.80	0.80	9.50	17.50	5.30	15.40	5.70	n
E. Germany	4.20	2.90	41.20	1.30	7.60	11.20	1.20	22.10	8.40	e



# Forza lavoro nei paesi europei: la *silhouette*

