

CORREZIONE ESERCIZIO 2 (15/10/09)

ESERCIZIO 1

Per verificare l'indipendenza dei due eventi A e B bisogna confrontare le quantità $P(B|A)$ e $P(B)$.

$P(B|A)$ = probabilità che si verifica l'evento B una volta che si è verificato l'evento A

$P(B)$ = probabilità che si verifica l'evento B

Per quanto riguarda $P(B|A)$:

Una volta che si è verificato A, le parti totali diventano 849 mentre quelle difettose saranno 49. Quindi la probabilità che la seconda parte è difettosa è: $P(B|A) = 49/849$

Per quanto riguarda $P(B)$:

A_1, A_2, \dots, A_n sono eventi mutuamente esclusivi ed esaustivi se $\bigcup_{i=1}^n A_i = A_1 \cup A_2 \dots \cup A_n = \Omega$ e

$A_i \cap A_j = \emptyset \quad i \neq j$. Sia B un evento in Ω . Allora $P(B) = \sum_{i=1}^n P(B \cap A_i) = \sum_{i=1}^n P(B|A_i)P(A_i)$.

Quindi possiamo valutare la probabilità che si verifichi B senza condizionamenti usando come eventi mutuamente esclusivi ed esaustivi A ed A^C .

In definitiva:

$$P(B) = P(A) \cdot P(B|A) + P(A^c) \cdot P(B|A^c) = 50/850 \cdot 49/849 + 800/850 \cdot 50/849 = 50/850$$

Quindi, essendo $P(B|A)$ diverso da $P(B)$, i due eventi sono indipendenti.

ESERCIZIO 2

Caricamento dei dati in Matlab:

```
>> dati = load('dataEserc1.txt')
```

Calcolo del numero ottimale di intervalli (per esempio con la regola di Larson):

```
>> N = length(dati)
>> r = 1 + 2.2*log10(N)
```

Definizione degli intervalli (classi):

```
>> edges = min(dati):(max(dati) - min(dati))/round(r):max(dati)
```

NB: siccome r deve essere un intero, è necessario arrotondarlo con il comando round

Calcolo della frequenza assoluta e relativa:

```
>> freqass = histc(dati, edges)
>> freqrel = freqass/N
```

Diagramma della frequenza assoluta e relativa:

```
>> bar(edges, freqass)
>> bar(edges, freqrel)
```

Calcolo delle frequenze cumulative assolute e relative:

```
>> freqcumass = cumsum(freqass)
>> freqcumrel = freqcumass/N
```

Diagramma delle frequenze cumulative assolute e relative:

```
>> bar(edges, freqcumass)
>> bar(edges, freqcumrel)
```

Calcolo di media, mediana, moda, deviazione standard e varianza:

```
>> mean(dati)
>> median(dati)
>> mode(dati)
>> std(dati)
>> var(dati)
```

NB: il comando mode (che nelle versioni meno recenti di Matlab non è predefinito) restituisce il valore tra i dati che è ripetuto più volte nel campione. Tuttavia non tiene conto della suddivisione in classi. Una valutazione più rigorosa della moda andrebbe effettuata individuando la classe che ha frequenza massima.