

Università degli Studi di Pisa

Corso Intensivo Sperimentale per la Preparazione di Laureati all'Insegnamento nelle Scuole Secondarie

Classe Disciplinare di Matematica per la Scuola Media Superiore

Anno Accademico 1994/1995

Una proposta per il Laboratorio di Matematica e Informatica nell'insegnamento del Calcolo delle Probabilità e della Statistica

Giorgio Meini

0. Introduzione.

I nuovi programmi di matematica per la scuola media superiore (Piano Nazionale Informatica e Brocca) prevedono sia la trattazione di argomenti di calcolo delle probabilità e statistica che la frequentazione del laboratorio di informatica. E' senz'altro opportuno distinguere l'uso del *computer* come "strumento" didattico per l'insegnamento e l'apprendimento della matematica dall'uso del *computer* come "oggetto" di conoscenza specifica [1], ciò nonostante nei nuovi programmi di insegnamento questi due diversi aspetti sono ricondotti nell'ambito della medesima disciplina denominata "Matematica e Informatica". Nel concreto della pratica didattica il *computer* viene effettivamente utilizzato con modalità differenziate:

- come esecutore di algoritmi codificati con uno specifico linguaggio di programmazione (per esempio: Pascal, ...);
- come piattaforma per un *software* applicativo che costituisce un ambiente "foglio di dati e di calcolo" interattivo e generico (per esempio: Lotus123, Excel, ...);
- come piattaforma per *software* applicativi di tipo specificamente matematico (per esempio: Derive, Mathematica, MapleV, ...);

Apparentemente la prima di queste modalità - la programmazione - non vede il *computer* come "strumento", ma come "oggetto": la codifica di algoritmi con un linguaggio di programmazione è certamente una conoscenza di informatica e non di matematica¹, ma una volta acquisita - come è d'altra parte previsto dai nuovi programmi di insegnamento - costituisce uno strumento disponibile anche in un contesto didatticamente diverso. Si noti per inciso che anche un uso meramente didattico dei linguaggi di programmazione per applicazioni di calcolo numerico richiede una adeguata consapevolezza critica rispetto alla valutazione degli errori numerici in cui è possibile incorrere (errori di rappresentazione, errori analitici, errori algoritmici, ... cfr. [3]).

La presente proposta consiste in un itinerario didattico - impostato nella sua generalità, ma non compiutamente sviluppato - per l'uso del *computer* come strumento nell'insegnamento e nell'apprendimento di alcuni argomenti di calcolo delle probabilità e di statistica sia nel biennio che nel triennio della scuola media superiore².

1. La simulazione di esperimenti aleatori consistenti in prove ripetute.

La linea didattica tracciata dal prof. Prodi per l'introduzione di elementi di calcolo delle probabilità nel biennio della scuola media superiore prevede di "compiere materialmente una serie di estrazioni casuali" per "stabilire il significato empirico della probabilità" [4]. Solo dopo avere interpretato - per mezzo della legge dei "grandi numeri" di Bernoulli che resta comunque "un enunciato interno alla teoria della probabilità" e che "non è un enunciato empirico" [4] - l'esperienza della realtà "si può fare ricorso al calcolatore ... [che] permette di costruire in concreto degli interessanti spazi di probabilità e di compiere rapidamente interessanti simulazioni" [4].

Pianificando opportunamente la scansione temporale degli argomenti oggetto di insegnamento è possibile disporre, già in questa prima fase di trattazione del calcolo delle probabilità, di alcune conoscenze minimali di programmazione, per esempio in linguaggio Pascal. A mio avviso è questo il caso in cui una conoscenza relativa ad una disciplina distinta - l'informatica - diviene uno "strumento" didattico il cui uso è possibile anche in ambito diverso. Si può così realizzare e

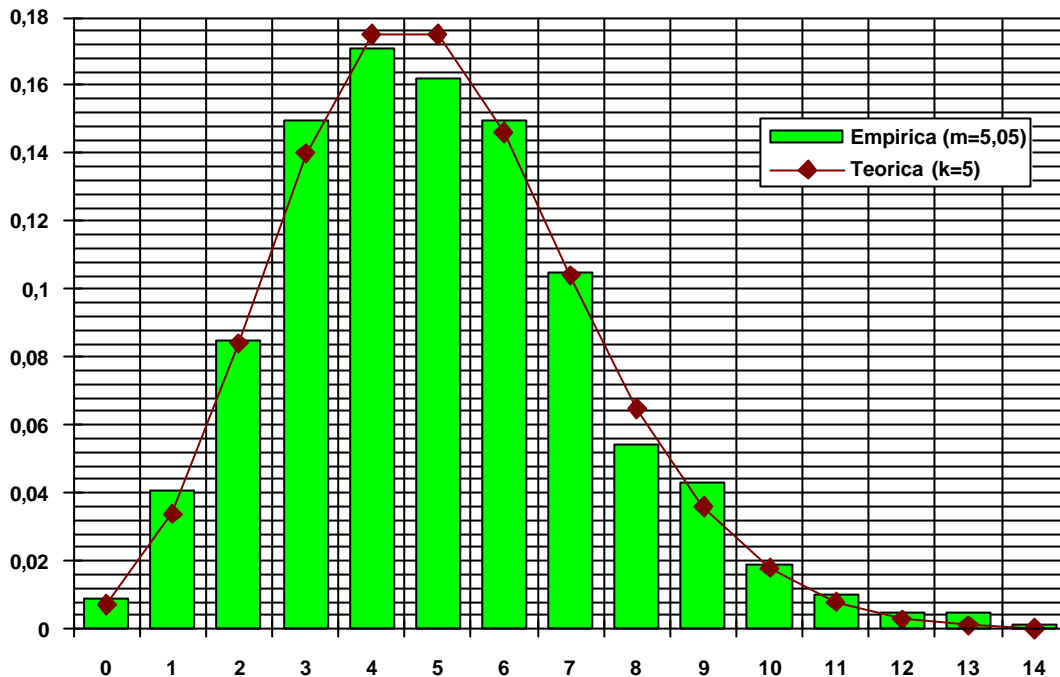
sperimentare - dopo avere introdotto le funzionalità del generatore di numeri casuali di cui dispone il linguaggio di programmazione³ - un primo e semplicissimo programma che permetta di simulare il lancio ripetuto di una moneta, contando il verificarsi degli eventi "testa" e "croce" (cfr. Appendice: programma 1).

Anche solo con conoscenze minime, sia nel campo della programmazione che in quello della probabilità, è già possibile costruire programmi di simulazione che rappresentino esperimenti per cui, a partire dalla generazione di più variabili aleatorie uniformi, "emerge" un comportamento non uniforme delle frequenze relative: un classico esempio consiste nel lancio ripetuto e contemporaneo di due dadi (cfr. Appendice: programma 2). Un programma di questo tipo fornisce come risultato le frequenze relative degli eventi simulati, da confrontarsi "a posteriori" con i valori teorici delle probabilità calcolati "a priori" ricorrendo ad uno schema combinatorio. Data l'importanza che riveste la rappresentazione grafica delle probabilità e delle frequenze relative - per esempio sotto forma di istogramma -, è utile ricorrere a questo scopo alle funzionalità integrate di un "foglio di dati e di calcolo" come può essere Lotus123 o Excel (è questa infatti un'altra delle attività previste dai nuovi programmi di insegnamento). Inoltre tale programma è facilmente estendibile e generalizzabile al lancio di 3 o più dadi: essendo gli eventi presi in considerazione per il calcolo delle frequenze relative - il punteggio complessivo ottenuto dal lancio contemporaneo di più dadi - la variabile aleatoria "somma" delle variabili aleatorie uniformi ed indipendenti associate al punteggio realizzato con ogni singolo dado, si dispone di una via "sperimentale" per fornire una prima intuizione del significato del teorema limite centrale (se si è optato per l'uso in laboratorio di un *package software* di tipo matematico - come Mathematica della Wolfram Research - è anche possibile sovrapporre alla rappresentazione grafica degli istogrammi delle probabilità o delle frequenze relative la curva normale di Gauss opportunamente parametrizzata).

2. La simulazione stocastica come metodologia didattica.

A mio parere riveste grande importanza didattica la possibilità di confrontare uno schema teorico non deterministico - per esempio la distribuzione di probabilità di una variabile aleatoria - con il sistema, reale o artificiale, che modella. In assenza di dati statistici è sempre possibile simulare - con un programma per calcolatore - il sistema originale: questa metodologia computazionale (simulazione stocastica) è ormai diffusa in molti campi di ricerca della Fisica e della Matematica [6] e, almeno in alcuni casi che sono comunque interessanti, non è di difficile implementazione anche disponendo di limitate conoscenze di programmazione⁴. Ad esempio, per introdurre la distribuzione di probabilità di Poisson per una variabile aleatoria discreta⁵ è interessante valutarne la possibilità di "adattamento" parametrico ai dati ottenuti da un esperimento di decadimento atomico per disintegrazione radioattiva. Partendo dal classico modello empirico e deterministico (ricavabile dai dati osservati sperimentalmente) secondo il quale il numero N di atomi non ancora decaduti all'istante t è $N(t) = N_0 e^{-kt}$ [8] si ha che il numero di atomi decaduti nell'intervallo di tempo $[t_0, t_1]$ è dato da $N(t_0) - N(t_1) = N_0(e^{-kt_0} - e^{-kt_1})$: applicando la "definizione frequentista" di probabilità si può quindi affermare [9] che la probabilità che un singolo atomo decada nell'intervallo di tempo $[t_0, t_1]$ è $P = [N(t_0) - N(t_1)] / N(t_0) = [N_0 / N(t_0)] (e^{-kt_0} - e^{-kt_1})$.

Non è complicato realizzare un programma che simuli il trascorrere successivo di intervalli discreti di tempo e l'eventuale disintegrazione dei singoli "atomi" - rappresentati come un *array* di valori binari aventi il significato di "vivo o morto"⁶ - con la stessa tecnica già illustrata per il lancio di dadi e monete (cfr. Appendice: programma 3). Nella mia esperienza di insegnamento la progettazione e l'implementazione di programmi per calcolatore come attività di *problem solving* ha sempre ottenuto una risposta positiva da parte degli studenti, almeno nel triennio della scuola media superiore quando le più elementari conoscenze di programmazione sono ormai consolidate. Nel caso particolare dello sviluppo di programmi di simulazione ho sempre constatato un grado di comprensione del sistema oggetto di studio superiore a quello riscontrato in situazioni di apprendimento diverse: ritengo che ciò sia dovuto al fatto che la stesura di un programma di simulazione, per quanto semplice, impone uno studio "analitico" del sistema in esame, normalmente presentato invece sotto la forma di un modello matematico "sintetico" (nella pratica della ricerca scientifica è proprio la non esistenza o conoscenza di un soddisfacente modello matematico che costringe alla simulazione computazionale del sistema da studiare). Il programma di simulazione della disintegrazione atomica radioattiva si comporta come un contatore *Geiger* virtuale fornendo, per ciascuno degli esperimenti simulati, il numero di disintegrazioni atomiche registrate per ogni singolo intervallo di tempo. I dati finali - valore, frequenza assoluta e frequenza relativa - sono invece visualizzati in modo conforme alla variabile aleatoria "numero di disintegrazioni per unità di tempo" e sono di conseguenza direttamente confrontabili con il modello teorico non deterministico costituito dalla distribuzione di probabilità di Poisson: si noti comunque che il comportamento "poissoniano" del modello "emerge" dall'interazione complessa di fenomeni aleatori elementari aventi una diversa distribuzione di probabilità. Il grafico seguente (realizzato con Excel della Microsoft) permette di confrontare i dati ottenuti con il programma di simulazione dell'esperimento di disintegrazione atomica con la distribuzione di probabilità teorica di Poisson parametrizzata in modo da avere valore atteso uguale alla media dei dati empirici⁷.



3. Le catene di Markov come esempio di processo stocastico.

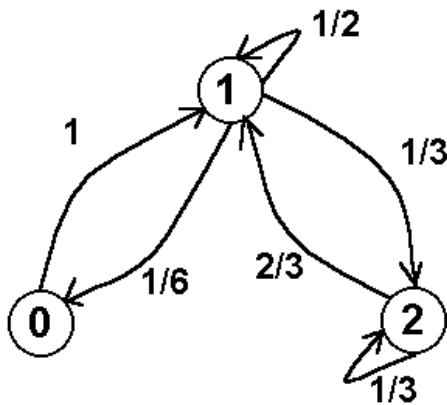
Anche se non esplicitamente citato nei nuovi programmi di insegnamento, il modello non deterministico delle "catene di Markov" - si tratta di una generalizzazione dello schema di prove ripetute in cui, secondo De Finetti, "il futuro, dato il presente, è indipendente dal passato" (si veda il capitolo III di [10] per una esposizione della teoria) - presenta alcune caratteristiche che lo rendono interessante sotto l'aspetto didattico:

- è un modello realistico di prototipizzazione di alcuni fenomeni fisici notevoli (moto browniano come passeggiata casuale e modello delle urne per la cinetica dei gas), ma può anche essere più semplicemente riferito ad un labirinto dal quale si tenta di uscire scegliendo la strada a caso lanciando un dado o una moneta;
- è una naturale estensione degli schemi probabilistici usualmente rappresentati con grafi etichettati in cui viene rilasciato il requisito di a-ciclicità (cfr. [4]);
- le proprietà matematiche del modello matriciale hanno importanti corrispondenze nelle caratteristiche del sistema (per esempio: esistenza del vettore fisso ed ergodicità);
- il modello matriciale è intrinsecamente interessante ed è facilmente "trattabile" sotto l'aspetto numerico ricorrendo alle funzionalità di calcolo matriciale di un *package software* come Mathematica della Wolfram Research o MapleV della Waterloo Maple Software.

Vediamo come modellizzare un semplice processo:

Inizialmente l'urna a sinistra contiene 2 palline bianche, mentre l'urna a destra contiene 3 palline nere: in ogni transizione si estrae casualmente una pallina da ciascuna urna e le due palline estratte vengono reintrodotte nelle urne scambiandole tra di loro. Lo stato del processo è dato dal numero di palline nere nell'urna di sinistra (0, 1 o 2).

A partire da semplici considerazioni combinatorie si può facilmente arrivare a costruire il seguente "grafo delle probabilità di transizione":



da cui si ottiene la "matrice di transizione" della catena di Markov relativa al processo illustrato:

$$\begin{bmatrix}
 0 & 1 & 0 \\
 1/6 & 1/2 & 1/3 \\
 0 & 2/3 & 1/3
 \end{bmatrix}$$

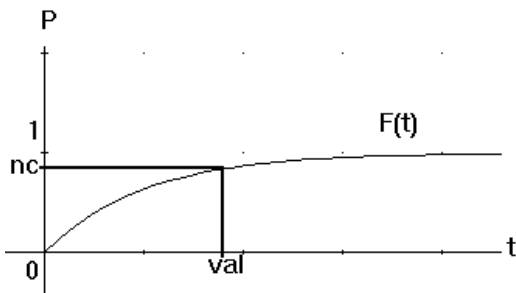
Ricorrendo alla matrice di transizione come "struttura di memorizzazione" dei valori di probabilità da cui dipende il processo è immediato realizzare un programma che, ripetendo più volte la simulazione del processo stesso, calcoli la frequenza relativa associata allo stato finale dopo un numero prefissato di transizioni (cfr. Appendice: programma 4). I risultati ottenuti dalla simulazione sono da confrontarsi con le previsioni del modello teorico matriciale: la teoria delle catene di Markov afferma che la probabilità del sistema di trovarsi in un determinato stato dopo N transizioni è data dalle componenti del vettore ottenuto come prodotto tra il vettore di probabilità iniziale (nel nostro caso [1,0,0]: lo stato iniziale è sempre 0) e la matrice di transizione elevata all'N-esima potenza.

Non è difficile svolgere velocemente e con precisione calcoli numerici di questo tipo con un *software* come Derive, Mathematica o MapleV (cfr. Appendice: sessione MapleV). Si può ricorrere allo strumento *software* anche per risolvere il sistema lineare che permette di trovare l'eventuale vettore fisso di probabilità v (tale che $vT=v$, dove T è la matrice di transizione) descrivente la distribuzione di probabilità degli stati del sistema al crescere indefinito del numero di transizioni.

4. La generazione di variabili aleatorie continue: una applicazione del metodo di Monte Carlo allo studio di una fila di attesa.

Come in molte applicazioni della matematica anche nello studio delle situazioni di incertezza non tutte le grandezze sono discrete o finite: una variabile aleatoria può assumere valori in un sottoinsieme continuo, limitato o illimitato, dei numeri reali. Per esempio, prendendo nuovamente in considerazione l'esperimento relativo al decadimento atomico radioattivo, se il numero di disintegrazioni osservate nel periodo di tempo t è una variabile aleatoria discreta - teoricamente illimitata - avente distribuzione di probabilità data dalla legge di Poisson $P_k(n)=(kt)^n e^{-kt}/n!$ allora gli inter-tempi osservati tra una disintegrazione e la successiva costituiscono una variabile aleatoria continua T che assume valori nell'intervallo reale illimitato $(0,+\infty)$: la relativa funzione di distribuzione della probabilità è $f_T(t)=ke^{-kt}$ (distribuzione esponenziale negativa⁸), mentre la funzione di ripartizione della probabilità - l'analogo continuo della probabilità cumulata del caso discreto - è $F_T(t)=1-e^{-kt}$.

Per generare in un programma di simulazione il valore di una variabile aleatoria continua si generalizza la tecnica già impiegata nel caso discreto usando i valori della probabilità cumulata (un esempio è il programma 4 riportato in appendice): il grafico che segue illustra la cosiddetta tecnica di Monte Carlo



Il valore generato (*val*) è ottenuto come ascissa corrispondente, secondo la funzione di ripartizione $F(t)$, all'ordinata del numero casuale (*nc*) fornito dal generatore con distribuzione uniforme nell'intervallo $[0,1]$ ⁹. Ai fini della computazione è ovviamente più semplice calcolare $val=F^{-1}(nc)$: nel caso di una distribuzione di probabilità esponenziale negativa si ha che $F(t)=1-e^{-kt}$ e, di conseguenza, che $F^{-1}(p)=[-\ln(1-p)]/k$; codificando in Pascal $val=F^{-1}(p)$ con $p=nc$ si ha

```
FUNCTION val:REAL;
  CONST k = ...;
  VAR nc: REAL;
  BEGIN
    nc:=RANDOM;
    val:=-LN(1-nc)/k
  END;
```

Un esempio per il quale è facile realizzare un breve programma di simulazione (cfr. Appendice: programma 5), ma che non è facilmente trattabile sotto l'aspetto teorico (almeno con gli strumenti matematici disponibili ad uno studente di scuola media superiore) è un modello di fila di attesa tecnicamente definito coda di tipo $M|M|1$. Si tratta di un particolare sistema di coda (per una introduzione generale ai sistemi di code si veda [11], [10] per una trattazione teorica approfondita) che ha le seguenti caratteristiche:

1. gli arrivi sono distribuiti nel tempo secondo la legge di Poisson di parametro kt con k numero medio di arrivi nell'unità di tempo (di conseguenza i tempi tra un arrivo e il successivo sono distribuiti con legge esponenziale negativa di parametro k e tempo medio tra un arrivo e il successivo $1/k$),
2. la coda è disciplinata secondo l'ordine di arrivo (FIFO: First In First Out),
3. nel sistema esiste un solo punto di servizio: il tempo di durata del servizio è una variabile aleatoria distribuita con legge esponenziale negativa di parametro c , essendo $1/c$ il tempo medio di servizio.

In teoria si dimostra¹⁰ che sotto tali ipotesi la lunghezza media della coda (inclusa l'unità attualmente servita) risulta essere $k/(c-k)$. È interessante confrontare questo risultato con i valori forniti dal programma di simulazione riportato in appendice: l'impossibilità di accedere ad un modello teorico soddisfacente permette di ricreare nella didattica una situazione tipica della modellizzazione matematica delle scienze sperimentali.

5. Conclusioni (più un semplice esempio di induzione statistica bayesiana).

Il percorso sopra delineato prevede di svolgere in laboratorio alcune unità didattiche molto diverse tra loro - anche sotto l'aspetto della complessità -, ma aventi tutte una impostazione comune: confrontare i dati forniti dall'osservazione di un esperimento virtuale - simulato per mezzo di un programma per *computer* - con le previsioni quantitative fornite da un modello teorico non deterministico. Il laboratorio di informatica diviene così il luogo dove si pratica una forma sperimentale di matematica che può seguire, o precedere, la presentazione di un modello più formalizzato - in questo caso probabilistico - della realtà. La realizzazione di semplici programmi di simulazione coinvolge le capacità analitiche degli studenti obbligandoli al tempo stesso all'uso dello strumento formale del linguaggio di programmazione; ma l'attività di programmazione non esaurisce la pratica di laboratorio che viene a comprendere anche l'uso di strumenti *software* appropriati per la risoluzione di problemi specifici come la rappresentazione di grafici o il calcolo numerico e simbolico. Questa impostazione didattica si applica facilmente a numerosi argomenti e concetti del calcolo della probabilità e della statistica: la legge dei grandi numeri, il teorema limite centrale, le distribuzioni notevoli di probabilità (normale di Gauss, binomiale, di Poisson, ...), le catene di Markov e molti altri ancora.

Il primo dei suoi limiti è certamente la supremazia che viene attribuita ad una interpretazione "frequentista" del concetto di probabilità rispetto ad una presentazione assiomatica o "soggettiva" ("cognitiva" secondo il prof. Prodi, cfr. [4]): questa

interpretazione è tuttavia in accordo con la prassi di molte scienze sperimentali e, nella pratica didattica, non è difficile rimarcare continuamente l'ambito di una sua corretta applicazione, anche ricorrendo esplicitamente a modelli alternativi. Una seconda limitazione potrebbe essere vista nell'obbligo di "concretizzare" un esempio particolare da simulare a fronte di un modello teorico avente caratteristiche proprie e valenza generale, ma la proposta didattica illustrata è relativa alla sola pratica di laboratorio e l'insegnante dispone di molti modi per integrarla con l'attività teorica da svolgersi in classe. Vediamo, come ultimo esempio e per testimoniare la generalità di questa pratica didattica, un rudimentale schema inferenziale (cfr. [12]) basato sulla formula di Bayes

$$P(B|A) = \frac{P(B) \cdot P(A|B)}{P(B) \cdot P(A|B) + P(\bar{B}) \cdot P(A|\bar{B})}$$

dove, per definizione di probabilità condizionata, $P(X|Y) = P(X \cap Y) / P(Y)$.

L'esperimento da simulare consiste nello scegliere a caso un dado da una coppia in cui uno è normale e l'altro è stato truccato trasformando il valore '1' in un secondo valore '6': osservando i risultati del lancio ripetuto del dado prescelto si intende calcolare, di volta in volta, la probabilità - a "posteriori" della conoscenza che via via si acquisisce - di avere scelto il dado truccato¹¹. Dato che nel corso nell'esperimento risulterà determinante, almeno ai nostri fini, contare il numero di '6' ottenuti in N lanci del dado, stabiliamo la seguente denominazione per gli eventi¹²:

A: "in N lanci si è ottenuto il valore '6' esattamente K volte",

B: "il dado è truccato",

da cui si ottiene che:

\bar{B} : "il dado NON è truccato",

$A \cap \bar{B}$: "in N lanci il valore '6' si è presentato esattamente K volte e il dado NON è truccato"

e

$A \cap B$: "in N lanci il valore '6' si è presentato esattamente K volte e il dado è truccato".

Ora la "probabilità a priori" - calcolata cioè senza disporre di informazioni derivate dall'esperienza - di avere scelto il dado truccato è $P(B) = 1/2$ e, di conseguenza, $P(\bar{B}) = 1 - P(B) = 1/2$ è la probabilità di non avere scelto il dado truccato; dopo N lanci in cui si ottiene esattamente K volte il valore '6' è possibile calcolare con la formula di Bayes $P(B|A)$ che rappresenta la "probabilità a posteriori" - calcolata cioè disponendo delle informazioni fornite dall'esperimento - di avere scelto il dado truccato: è interessante vedere come varia la "stima" della probabilità al crescere della "quantità" di informazione disponibile (inferenza statistica bayesiana).

Ricorrendo allo schema di Bernoulli per le prove ripetute si possono calcolare le probabilità condizionate $P(A|B)$ e $P(A|\bar{B})$ come applicazione della distribuzione binomiale (si ricordi che la probabilità di ottenere il valore '6' con il dado normale vale $1/6$, mentre la probabilità di ottenere il valore '6' con il dado truccato vale $2/6 = 1/3$):

$$P(A|\bar{B}) = \frac{P(A \cap \bar{B})}{P(\bar{B})} = \frac{N!}{K!(N-K)!} \left(\frac{1}{6}\right)^K \cdot \left(1 - \frac{1}{6}\right)^{N-K}$$

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{N!}{K!(N-K)!} \left(\frac{1}{3}\right)^K \cdot \left(1 - \frac{1}{3}\right)^{N-K}$$

da cui, applicando la formula di Bayes, si ha:

$$\begin{aligned}
 P(B|A) &= \frac{\frac{1}{2} \cdot 2 \cdot \frac{N!}{K!(N-K)!} \cdot (1/3)^K (2/3)^{N-K}}{\frac{1}{2} \cdot 2 \cdot \frac{N!}{K!(N-K)!} (1/3)^K (2/3)^{N-K} + \frac{1}{2} \cdot 2 \cdot \frac{N!}{K!(N-K)!} (1/6)^K (5/6)^{N-K}} = \\
 &= \frac{(1/3)^K (2/3)^{N-K}}{(1/3)^K (2/3)^{N-K} + (1/6)^K (5/6)^{N-K}}
 \end{aligned}$$

Il programma 6 riportato in appendice effettua la scelta, casuale ed equiprobabile, tra dado normale e dado truccato e simula la ripetizione di un numero specificato di lanci e visualizzando per ciascuno di essi il valore $P(B|A)$ calcolato con la formula ricavata sopra e corrispondente alla stima corrente della probabilità di avere prescelto il dado truccato¹³.

Note.

(1) Il prof. Bernardi ha comunque ricordato [2]: "Nei programmi del biennio la logica è vista come un possibile punto di collegamento fra matematica e informatica. L'idea di base è che la pratica informatica metta in evidenza, dia concretezza a concetti logici: ad esempio, sono citate nei programmi le differenze fra linguaggio e metalinguaggio, e fra sintassi e semantica. Anche questa idea risponde all'esigenza di trovare modelli concreti ed operativi per la logica, evitandone una presentazione astratta. [...]. (Naturalmente, il discorso è valido se l'insegnamento dell'informatica non consiste esclusivamente nell'uso di programmi già confezionati)."

(2) L'itinerario didattico proposto è stato parzialmente sperimentato presso l'ITIS di Siena dove - negli anni scolastici dal 1984/1985 al 1988/1989 - ho insegnato come assistente di Laboratorio di Matematica e di Calcolo delle Probabilità e Statistica agli studenti del triennio di specializzazione in Informatica Industriale e presso l'ITIS di Livorno dove - nell'anno scolastico 1993/1994 - ho insegnato come docente di Calcolo delle Probabilità e Statistica al corso serale di specializzazione in Informatica Industriale. È stato inoltre oggetto di un corso di aggiornamento che ho tenuto nell'anno scolastico 1992/1993 ai docenti di Matematica dell'ITG di Livorno.

(3) Nel linguaggio di programmazione Pascal la funzione predefinita RANDOM permette di generare il valore di una variabile casuale avente distribuzione di probabilità uniforme: in molte implementazioni è possibile parametrizzare la funzione RANDOM per ottenere numeri casuali di tipo REAL - che, è bene ricordarlo, costituiscono comunque un insieme di valori limitato, discreto e finito, cfr. [3] - appartenenti all'intervallo $[0,1)$ o numeri casuali di tipo INTEGER compresi tra 0 incluso ed un intero positivo N specificato escluso. È senz'altro una scelta corretta nascondere in prima istanza agli studenti "il problema - veramente delicato - di come faccia un irreprensibile calcolatore deterministico a fornire numeri a caso" [4]: nella mia esperienza di insegnamento ho ottenuto un valido riscontro da parte degli studenti discutendo la struttura interna - finita e deterministica! - di un semplice generatore di numeri casuali basato sul metodo congruenziale lineare [5] nell'ultimo anno del triennio della scuola secondaria superiore. Una esperienza sempre istruttiva per gli studenti è la codifica di un semplice programma basato sul metodo di Monte Carlo per la stima statistica del valore di p (cfr. [3]). Alcune scuole ad indirizzo tecnico prevedono tra le discipline oggetto di studio nel triennio la statistica: in questo caso è interessante sottoporre il generatore di numeri casuali del linguaggio di programmazione ad un test statistico di validità, per esempio il test chi-quadro con stimatore di Pearson (cfr. [5]). È comunque didatticamente utile che lo studente del triennio della scuola media superiore utilizzi un generatore di numeri casuali minimale (limitato cioè alla generazione uniforme di numeri pseudo-casuali e pseudo-reali compresi tra 0 ed 1) realizzando in proprio i passi di programma necessari alla simulazione di una variabile aleatoria avente le caratteristiche desiderate: nel biennio - avendo a che fare principalmente con variabili aleatorie discrete, finite ed uniformemente distribuite - è sufficiente ipotizzare una corrispondenza biunivoca tra i valori interi ed appartenenti ad un dato intervallo forniti con distribuzione uniforme dal generatore di numeri casuali e gli eventi dell'esperimento che si intende simulare rappresentanti i possibili "valori" della variabile aleatoria.

(4) Credo non sia dissimile da questo l'approccio suggerito dal prof. Bini [7]: "In questo corso vogliamo portare all'attenzione dei docenti delle scuole superiori gli aspetti algoritmici e costruttivi della matematica che vengono generalmente trascurati. Riteniamo che la loro valenza culturale e didattica sia molto importante per vari motivi.

- [...].

- La capacità di modellizzare anche semplici problemi del mondo reale permette di acquisire pienamente lo strumento matematico che a questo punto non rimane più un oggetto astratto difficilmente raggiungibile. La matematica, nei suoi aspetti applicativi, appare anche come disciplina concreta, tangibile, utile e potente.

- La matematica algoritmica permette un uso attivo e costruttivo del calcolatore che diventa uno strumento per poter risolvere i problemi in modo creativo. L'alunno può raggiungere un ruolo attivo nel risolvere autonomamente semplici problemi del mondo reale 'facendo' matematica algoritmica col calcolatore.

- L'utente del calcolatore è costretto ad acquisire un linguaggio formalmente corretto (linguaggio di programmazione) cosa che raramente accade col solo uso del linguaggio matematico. [...].

- Gli aspetti modellistici hanno caratteristiche interdisciplinari e sono più facilmente trasferibili per la loro concretezza.

- Una presentazione della matematica fatta per problemi, dove l'aspetto algoritmico e computazionale permette una maggiore concretezza espositiva, fornisce anche un legame naturale fra matematica e calcolatore nell'insegnamento della matematica e dell'informatica nelle scuole superiori. Questo tipo di presentazione può ripercorrere schemi analoghi a quelli con cui si svolge gran parte della ricerca scientifica.

[...].

Occorre inoltre sottolineare che l'uso del calcolatore che tende ad essere proposto nelle scuole è talvolta distorto e comunque generalmente limitativo, passivo ed acritico. Il calcolatore viene spesso proposto, in gran parte dei prodotti software commercializzati, come surrogato dell'insegnante o del laboratorio fisico; vengono presentate delle simulazioni a calcolatore preconfezionate, 'a scatola chiusa', che sottraggono allo studente i processi di modellizzazione, di risoluzione ed

analisi algoritmica, e di programmazione."

(5) $P_k(n) = k^n e^{-k} / n!$ si ottiene dalla distribuzione di probabilità binomiale $P_{N,p}(n) = \{N! / [n!(N-n)!]\} p^n (1-p)^{N-n}$ che caratterizza lo schema delle prove indipendenti di Bernoulli per $n \rightarrow \infty$, $p \rightarrow 0$ e $k = Np$: la dimostrazione è fondata su alcuni limiti notevoli generalmente conosciuti dagli studenti della scuola media superiore. Per il calcolo del valore atteso e della varianza, che risultano essere entrambi uguali al parametro k , è necessario invece conoscere lo sviluppo in serie di potenze (serie di Taylor) della funzione $f(x) = e^x$.

(6) La terminologia è quella classica relativa ai processi stocastici di "nascita e morte" di cui il processo di Poisson è un esempio particolare.

(7) La "bontà" dell'adattamento del modello al comportamento del sistema simulato può essere eventualmente quantificata per mezzo del test statistico chi-quadro con l'ausilio dello stimatore di Pearson.

(8) La funzione di ripartizione della probabilità di una variabile aleatoria continua T ha il significato $P(T \leq t)$. Nel nostro caso se si verificano due eventi aventi distanza nel tempo non superiore a t , allora si verifica almento un evento in un intervallo di tempo t : la probabilità che l'intervallo di tempo tra due eventi successivi non sia superiore a t è cioè uguale alla probabilità che si verifichi almento un evento in un intervallo di tempo t

$$F_T(t) = P(T \leq t) = \sum_{n=1}^{\infty} P_{kt}(n) = 1 - P_{kt}(0) = 1 - e^{-kt}.$$

Questa breve dimostrazione - semplice, ma dall'esito non scontato a priori - si basa sul teorema che stabilisce la relazione tra le misure di probabilità di 2 eventi complementari: è un'occasione didattica per sottolineare il ruolo che assume l'assiomatica di Kolmogorov nel calcolo delle probabilità. La funzione di distribuzione di probabilità è la derivata della funzione di ripartizione, in questo caso: $f_T(t) = ke^{-kt}$. È immediato calcolare, applicando la definizione integrale per v. a. continue, il valore atteso $1/k$ e la varianza $1/k^2$.

(9) A rigore è necessario dimostrare che ogni valore appartenente ad un qualsiasi intervallo (t_1, t_2) viene generato con probabilità $P(t_1 < T < t_2) = F(t_2) - F(t_1)$: l'intervallo $[F(t_1), F(t_2)]$ - immagine di (t_1, t_2) per mezzo della funzione $F(t)$ - ha infatti probabilità uguale alla sua ampiezza di contenere un numero casuale generato uniformemente nell'intervallo $[0, 1)$.

(10) Nonostante l'apparente semplicità del risultato per la dimostrazione si ricorre ad un sistema di equazioni differenziali probabilistiche definite ricorsivamente.

(11) Un problema di questo tipo permette di svolgere una riflessione sul concetto di probabilità come "grado di fiducia" (De Finetti) nel verificarsi di un evento in base alle informazioni di cui si dispone.

(12) Nella prassi didattica è importante sottolineare la corrispondenza tra il concetto di "evento" e i concetti di "enunciato" e di "insieme": gli studenti, nel contesto del calcolo delle probabilità, dovrebbero in qualche modo unificare la logica delle proposizioni e la teoria degli insiemi secondo il seguente schema:

EVENTI – ENUNCIATI	INSIEMI
certo	universo
impossibile	vuoto
implicazione	inclusione
negazione	complementare
disgiunzione logica (OR)	unione
coniunzione logica (AND)	intersezione
incompatibili	disgiunti

(13) È possibile aumentare la "velocità di convergenza" della successione dei valori di probabilità sostituendo, nella formula, i valori di probabilità costanti degli eventi B ("dado truccato") e $\neg B$ ("dado non truccato") con il valore di probabilità ottenuto dall'applicazione del teorema di Bayes nell'iterazione immediatamente precedente ed il suo complementare rispetto a 1.

Riferimenti Bibliografici.

[1] M. A. Mariotti, "Tecnologie informatiche come supporto alla didattica della matematica ...", Nota per i partecipanti al Corso Intensivo Sperimentale ..., Università di Pisa, Dipartimento di Matematica, Primavera 1995
 [2] C. Bernardi, "La logica nella Scuola Secondaria", in "L'insegnamento della Matematica e delle Scienze integrate", n. 11-12 vol. 16, Novembre-Dicembre 1993

- [3] R. Bevilacqua, D. Bini, M. Capovani, O. Menchi, "Introduzione alla Matematica Computazionale", Zanichelli, 1987
- [4] G. Prodi, "La didattica della probabilità", Nota per i partecipanti al Corso Intensivo Sperimentale ..., Università di Pisa, Dipartimento di Matematica, Primavera 1995
- [5] D. Knuth, "The Art of Computer Programming", Vol. 2, "Seminumerical Algorithms", Addison-Wesley, 1969
- [6] S. Wolfram, "Software nella scienza e nella matematica", Le Scienze (ed. it. di *Scientific American*), Novembre 1985
- [7] D. Bini, "Matematica, Mondo Reale e Calcolatore", Nota introduttiva per i partecipanti al Corso Intensivo Sperimentale ..., Università di Pisa, Dipartimento di Matematica, Primavera 1995
- [8] B. Javorskij, A. Detlaf, "Manuale di Fisica", ed. MIR, Mosca 1977
- [9] P. Hoel, S. Port, C. Stone, "Introduction to Probability Theory", Houghton Mifflin, 1971
- [10] B. Gnedenko, "Teoria della probabilità", Editori Riuniti, 1979
- [11] Martin A. Leibowitz, "Le Code", Le Scienze (ed. it. di *Scientific American*), Gennaio 1969
- [12] V. Colagrande, G. Di Biase, "Simulazione su calcolatore di alcuni esempi di applicazione del teorema di Bayes", Periodico di Matematiche, ?, 1990

APPENDICE.

Tutti i programmi codificati in linguaggio PASCAL sono stati compilati ed eseguiti con BORLAND PASCAL 6.0 per DOS. L'esempio di sessione MAPLE V è stato prodotto con MapleV 3.0 per Windows della Waterloo Maple Software.

Programma 1.

```
PROGRAM TestaCroce(INPUT,OUTPUT);
VAR
  ConteggioProve,NumeroProve: INTEGER;
  NumeroTeste,NumeroCroci: INTEGER;
BEGIN
  RANDOMIZE;
  NumeroTeste:=0;
  NumeroCroci:=0;
  WRITE('Numero di prove da ripetere? ');
  READLN(NumeroProve);
  FOR ConteggioProve:=1 TO NumeroProve DO
    IF RANDOM(2)=0 THEN {testa}
      BEGIN
        WRITE('T');
        NumeroTeste:=NumeroTeste+1
      END
    ELSE {croce}
      BEGIN
        WRITE('C');
        NumeroCroci:=NumeroCroci+1
      END;
  WRITELN; WRITE(NumeroProve,' lanci: ');
  WRITELN(NumeroTeste,' teste e ',NumeroCroci,' croci')
END.
```

Programma 2.

```
PROGRAM DueDadi(INPUT,OUTPUT);
VAR
  ConteggioProve,NumeroProve: INTEGER;
  PrimoDado, SecondoDado: INTEGER;
  Punteggio: INTEGER;
  FrequenzaPunteggio: ARRAY[2..12] OF INTEGER;
BEGIN
  RANDOMIZE;
  FOR Punteggio:=2 TO 12 DO FrequenzaPunteggio[Punteggio]:=0;
  WRITE('Numero di prove da ripetere? ');
  READLN(NumeroProve);
  FOR ConteggioProve:=1 TO NumeroProve DO
    BEGIN
      PrimoDado:=TRUNC(RANDOM*6+1);
      SecondoDado:=TRUNC(RANDOM*6+1);
      Punteggio:=PrimoDado+SecondoDado;
      WRITELN(PrimoDado,'+',SecondoDado,'=',Punteggio);
      FrequenzaPunteggio[Punteggio]:=FrequenzaPunteggio[Punteggio]+1
    END;
  WRITELN(NumeroProve,' lanci');
  WRITELN('Frequenze assolute punteggi:');
```

```

FOR Punteggio:=2 TO 12 DO
  WRITELN(Punteggio,') ',FrequenzaPunteggio[Punteggio]);
END.

```

Programma 3.

```

PROGRAM AtomiRadioattivi(OUTPUT);
CONST
  N0=10000; {numero iniziale di atomi}
  k=0.0005; {costante di decadimento radioattivo}
  Ti=0; {tempo iniziale esperimento simulato}
  Tf=100; {tempo finale esperimento simulato}
  dt=1; {intervallo di tempo per la simulazione discreta}
  NumEsper=10; {numero di esperimenti completi da ripetere}
VAR
  Atomi: ARRAY[1..N0] OF BOOLEAN;
  Freq: ARRAY[0..N0] OF INTEGER;
  t,t0,t1: REAL;
  NumAtomi,NumDis,Atomo,i,Esper,NumMaxDis: INTEGER;
FUNCTION Prob(t0,t1: REAL; N: INTEGER):REAL;
  BEGIN
    Prob:=(N0/N)*(EXP(-k*t0)-EXP(-k*t1))
  END;
BEGIN
  FOR i:=0 TO N0 DO Freq[i]:=0;
  NumMaxDis:=0;
  FOR Esper:=1 TO NumEsper DO
    BEGIN
      WRITELN('Esperimento ',Esper);
      RANDOMIZE;
      FOR Atomo:=1 TO N0 DO Atomi[Atomo]:=TRUE; {"vivo"}
      t:=Ti;
      NumAtomi:=N0;
      WHILE (t<=Tf) DO
        BEGIN {simulazione intervallo di tempo [t,t+dt]}
          t0:=t;
          t1:=t+dt;
          WRITE(t0,'-',t1,':');
          NumDis:=0;
          FOR Atomo:=1 to N0 DO
            BEGIN
              IF Atomi[Atomo] THEN
                IF RANDOM<Prob(t0,t1,NumAtomi) THEN
                  BEGIN {disintegrazione}
                    Atomi[Atomo]:=FALSE; {"morto"}
                    NumDis:=NumDis+1;
                  END
            END;
          WRITELN(' ',NumDis);
          Freq[NumDis]:=Freq[NumDis]+1;
          IF NumDis>NumMaxDis THEN NumMaxDis:=NumDis;
          NumAtomi:=NumAtomi-NumDis;
          t:=t+dt;
        END
      END;
    END;
  END;

```

```

WRITELN('Frequenze assolute e relative');
WRITELN('del numero di disintegrazioni per unita` di tempo:');
FOR i:=0 TO NumMaxDis DO
  WRITELN(i,' : ',Freq[i],' - ',Freq[i]/(NumEsper*((Tf-Ti)/dt)))
END.

```

Programma 4.

```

PROGRAM Markov (INPUT,OUTPUT);
CONST
  NumStati = 3;
  MatTrans: ARRAY [0..(NumStati-1),0..(NumStati-1)] OF REAL =
    ((0,1,0),
     (1/6,1/2,1/3),
     (0,2/3,1/3));
  StatoIniziale = 0;
VAR
  Esperimento, NumEsperimenti: INTEGER;
  Trans, NumTrans: INTEGER;
  Stato, ProssimoStato: 0..(NumStati-1);
  FreqStatoFinale: ARRAY [0..(NumStati-1)] OF INTEGER;
  NumCas, ProbCumulat: REAL;
BEGIN
  FOR Stato:=0 TO (NumStati-1) DO FreqStatoFinale[Stato]:=0;
  WRITE('Numero Transizioni? ');
  READLN(NumTrans);
  WRITE('Numero Esperimenti? ');
  READLN(NumEsperimenti);
  FOR Esperimento:=1 TO NumEsperimenti DO
    BEGIN
      WRITELN('Esperimento ',Esperimento);
      Stato:=StatoIniziale;
      RANDOMIZE;
      WRITE(Stato);
      FOR Trans:=1 TO NumTrans DO
        BEGIN
          NumCas:=RANDOM;
          ProbCumulat:=MatTrans[Stato,0];
          ProssimoStato:=0;
          WHILE NumCas>ProbCumulat DO {determinazione stato successivo}
            BEGIN
              ProssimoStato:=ProssimoStato+1;
              ProbCumulat:=ProbCumulat+MatTrans[Stato,ProssimoStato]
            END;
          Stato:=ProssimoStato;
          WRITE('->',Stato)
        END;
        FreqStatoFinale[Stato]:=FreqStatoFinale[Stato]+1;
        WRITELN;
      END;
    WRITELN('Frequenze assolute e relative stato finale');
    WRITE('in ',NumEsperimenti,' esperimenti');
    WRITELN(' di ',NumTrans,' transizioni ciascuno:');
    FOR Stato:=0 TO (NumStati-1) DO
      BEGIN

```

```

WRITE(Stato,') ',FreqStatoFinale[Stato]);
WRITELN(' - ',FreqStatoFinale[Stato]/NumEsperimenti)
END
END.

```

Sessione MapleV.

```

> with(linalg);
> T:=matrix([[0,1,0],[1/6,1/2,1/3],[0,2/3,1/3]]);

```

$$T := \begin{bmatrix} 0 & 1 & 0 \\ \frac{1}{6} & \frac{1}{2} & \frac{1}{3} \\ 0 & \frac{2}{3} & \frac{1}{3} \end{bmatrix}$$

```

# T è la matrice di transizione
> T10:=evalm(T^10);

```

$$T^{10} := \begin{bmatrix} \frac{1007855}{10077696} & \frac{6046447}{10077696} & \frac{503899}{1679616} \\ \frac{6046447}{60466176} & \frac{12093349}{20155392} & \frac{9069841}{30233088} \\ \frac{503899}{5038848} & \frac{9069841}{15116544} & \frac{2267503}{7558272} \end{bmatrix}$$

```

> for i to 3 do for j to 3 do T10[i,j]:=evalf(T10[i,j],3) od od;
> evalm(T10);

```

$$\begin{bmatrix} .100 & .600 & .300 \\ .1000 & .600 & .300 \\ .100 & .600 & .300 \end{bmatrix}$$

```

# T10 è la matrice di transizione elevata alla 10ª potenza (gli elementi di T10 sono stati trasformati da valori esatti a valori approssimati)

```

```

> V0:=matrix([[1,0,0]]);

```

$$V0 := [1 \quad 0 \quad 0]$$

```

# V0 è il vettore di probabilità iniziale
> evalm(multiply(V0,T));

```

$$\begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$

#V0*T rappresenta la distribuzione di probabilità dei singoli stati dopo una transizione
 > evalm(multiply(V0,T^2));

$$\begin{bmatrix} \frac{1}{6} & \frac{1}{2} & \frac{1}{3} \end{bmatrix}$$

#V0*(T^2)=V0*(T*T)=(V0*T)*T rappresenta la distribuzione di probabilità dopo 2 transizioni
 > evalm(multiply(V0,T10));

$$\begin{bmatrix} .100 & .600 & .300 \end{bmatrix}$$

#V0*T10=V0*(T^10) rappresenta la distribuzione di probabilità dopo 10 transizioni ed è il vettore fisso della catena

Programma 5.

```
PROGRAM MM1 (OUTPUT);
CONST
  NumArrivi=1000;
  IntertempoMedioArrivi=3;
  k=1/IntertempoMedioArrivi;
  TempoMedioServizio=2;
  c=1/TempoMedioServizio;
TYPE
  Tempo=REAL;
VAR
  LunMedia: LONGINT;
  TempoSimulato: Tempo;
  Prossimo: Tempo;
  Delta: Tempo;
  Coda: ARRAY [0..(NumArrivi-1)] OF Tempo;
  {per ogni elemento in coda memorizza il relativo tempo di servizio}
  Primo, Ultimo: 0..(NumArrivi-1);
  LunCoda: 0..NumArrivi;
  Arrivi: 1..NumArrivi;
FUNCTION ProssimoArrivo (t: Tempo): Tempo;
  VAR NumeroCas: REAL;
  BEGIN
    {metodo di Monte Carlo e della funzione di ripartizione inversa per
    generare una v.a. continua con distribuzione esponenziale negativa}
    NumeroCas:=RANDOM;
    ProssimoArrivo:=t+(-LN(1-NumeroCas))/k
  END;
FUNCTION Durata: Tempo;
  VAR NumeroCas: REAL;
  BEGIN
    {metodo di Monte Carlo e della funzione di ripartizione inversa per
    generare una v.a. continua con distribuzione esponenziale negativa}
```

```

    NumeroCas:=RANDOM;
    Durata:=-LN(1-NumeroCas)/c
END;
BEGIN
RANDOMIZE;
LunMedia:=0;
Primo:=0; Ultimo:=0;
LunCoda:=0;
TempoSimulato:=0;
FOR Arrivi:= 1 TO NumArrivi DO
    BEGIN {simulazione intervallo di tempo tra un arrivo e il successivo}
        Prossimo:=ProssimoArrivo(TempoSimulato);
        Delta:=Prossimo-TempoSimulato;
        WHILE ((LunCoda > 0) AND (Delta > 0)) DO
            BEGIN {simulazione servizio: estrazioni da coda}
                IF Delta >= Coda[Primo] THEN
                    BEGIN
                        Delta:=Delta-Coda[Primo];
                        Primo:=(Primo+1) MOD NumArrivi;
                        LunCoda:=LunCoda-1
                    END
                ELSE
                    BEGIN
                        Coda[Primo]:=Coda[Primo]-Delta;
                        Delta:=0
                    END
                END;
            END;
            WRITELN(LunCoda);
            LunMedia:=LunMedia+LunCoda;
            Coda[Ultimo]:=Durata; {memorizzazione tempo di servizio nuovo arrivo}
            Ultimo:=(Ultimo+1) MOD NumArrivi; {simulazione nuovo arrivo}
            LunCoda:=LunCoda+1;
            TempoSimulato:=Prossimo
        END;
        WRITELN('Lunghezza media della coda: ',(LunMedia/NumArrivi))
    END.

```

Programma 6.

```

PROGRAM Bayes(OUTPUT);
CONST
    NumProve=100;
VAR Prova,Num6,Dado: INTEGER;
FUNCTION IntPow (Base: REAL; NonNegInt: INTEGER): REAL;
    {extract from Soft Warehouse DERIVE support.pas file}
    BEGIN
        IF NonNegInt = 0 then IntPow := 1
        ELSE IF ODD (NonNegInt) THEN IntPow := Base * IntPow (Base, NonNegInt - 1)
            ELSE IntPow := SQR (IntPow (Base, NonNegInt DIV 2))
        END;
    FUNCTION PBA(n,k: INTEGER): REAL; {cfr. testo}
        BEGIN
            PBA:=(IntPow(1/3,k)*IntPow(2/3,n-k))/(IntPow(1/6,k)*IntPow(5/6,n-k)+
                IntPow(1/3,k)*IntPow(2/3,n-k))
        END;
    END;

```



```
BEGIN
RANDOMIZE;
Prova:=0; Num6:=0;
IF RANDOM(2) = 0 THEN BEGIN {scelta casuale dado truccato/non truccato}
    WRITELN('Dado non truccato');
    FOR Prova:=1 TO NumProve DO
        BEGIN
            Dado:=TRUNC(RANDOM*6+1);
            IF Dado = 6 THEN Num6:=Num6+1;
            WRITELN(Prova,PBA(Prova,Num6))
        END
    END
ELSE BEGIN
    WRITELN('Dado truccato');
    FOR Prova:=1 TO NumProve DO
        BEGIN
            Dado:=TRUNC(RANDOM*6+1);
            IF Dado =1 THEN Dado:=6;
            IF Dado = 6 THEN Num6:=Num6+1;
            WRITELN(Prova,PBA(Prova,Num6));
        END
    END
END.
```