# Progetto di Simulazione

A cura di Eduart Mustafa

# Analisi McDrive Terni

Università degli Studi di Perugia

Anno Accademico 2020/2021

# Introduzione

Il responsabile del McDonald di Terni ha problemi a determinare il numero di dipendenti da attivare, considerando in particolare il servizio McDrive e l’afflusso settimanale .

Decidi quindi di simulare il modello di sistema ponendoti tu stesso dei limiti (per es. attesa max in coda tollerata dalle/dai clienti). Sviluppa il modello di simulazione e implementalo nel linguaggio che desideri.

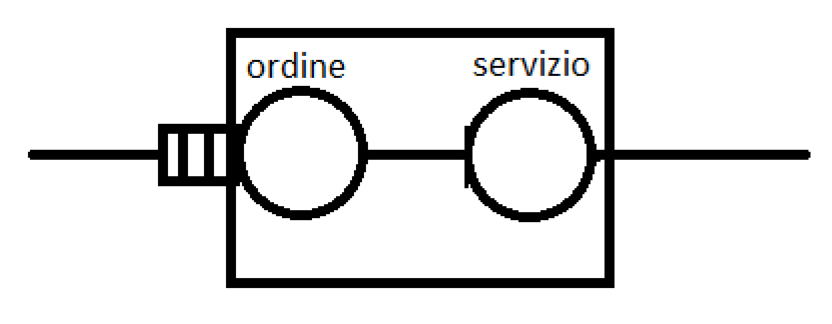
Tenta di stimare, a partire da reali osservazioni, i modelli di arrivo e di servizio, così come la disciplina delle code. Usa parte delle osservazioni per sviluppare le distribuzioni empiriche e convalida poi il simulatore usando le rimanenti osservazioni (al 90% del livello di confidenza).

Quindi cerca di determinare le soluzioni per i problemi del responsabile.

# Struttura del McDrive

Il McDrive funziona nel seguente modo: I clienti entrato nell'area del McDrive, giungono davanti al box di un unico operatore che prende l'ordine. Una volta ordinato, il cliente ordinante si sposta nel box dove deve pagare e prelevare l'ordine ordinato. La coda è unica.

La struttura può essere rappresentata in questo modo:




Il nostro obiettivo è quello di simulare questo modello e valutare se debba essere migliorato, assumendo che il valore atteso del **tempo di risposta** non debba essere superiore ai **5 minuti**.

# Raccolta dei dati

Per poter modellare teoricamente il modello, ho analizzato il flusso dei clienti tra le 12:00 e le 14:30, orario in cui fluisce un maggior numero di macchine, ma che allo stesso tempo consente di poter analizzare il problema in condizioni stazionarie (**Non ho considerato la sera, ad esempio, perché è molto probabile che nei giorni di punta la condizione di stazionarietà venga meno, a causa del grande numero di clienti**).

Nelle prossime slide andrò ad analizzare i dati relativi a 3 situazioni:

1. Arrivi nel tempo considerato considerando un arco di 10 minuti alla volta

2. Tempi di ordinazione per ogni macchina arrivata

3. Tempi di pagamento e servizio per ogni macchina arrivata

# Analisi del numero di arrivi 1

| Orario | Numero arrivi |
|---|---|
| 12.00-12.10 | 0 |
| 12.10-12.20 | 1 |
| 12.20-12.30 | 4 |
| 12.30-12.40 | 3 |
| 12.40-12.50 | 5 |
| 12.50-13.00 | 4 |
| 13.00-13.10 | 4 |
| 13.10-13.20 | 3 |
| 13.20-13.30 | 2 |
| 13.30-13.40 | 3 |
| 13.40-13.50 | 3 |
| 13.50-14.00 | 1 |
| 14.00-14.10 | 2 |
| 14.10-14.20 | 2 |
| 14.20-14.30 | 2 |
| | **39** |

Nell'arco di tempo considerato sono giunte 39 macchine, con un arrivo che si è intensificato nell'ora di pranzo. Prendendo come campione il numero di macchine arrivate abbiamo questa distribuzione di frequenze:

| Macchine osservate | Frequenza |
|---|---|
| 0 | 1 |
| 1 | 2 |
| 2 | 4 |
| 3 | 4 |
| 4 | 3 |
| 5 | 1 |
| | **15** |

Poiché le frequenze osservate sono inferiori a 30, ho utilizzato il test di Kolmogorov-Smirnov.

# Analisi del numero di arrivi 2

Prima di utilizzare il test, ho dovuto cercare una distribuzione candidata. Due indici molto utilizzati per riconoscere le distribuzioni candidate sono la media e la varianza campionaria. I risultati ottenuti sono i seguenti:

| | |
|---|---|
| Media | 2,6 |
| Varianza | 1,828571429 |

Possiamo osservare come media e varianza siano piuttosto simili. Una distribuzione nota che può essere usata è la Poissoniana, la quale ha la particolarità di avere media uguale alla varianza*.

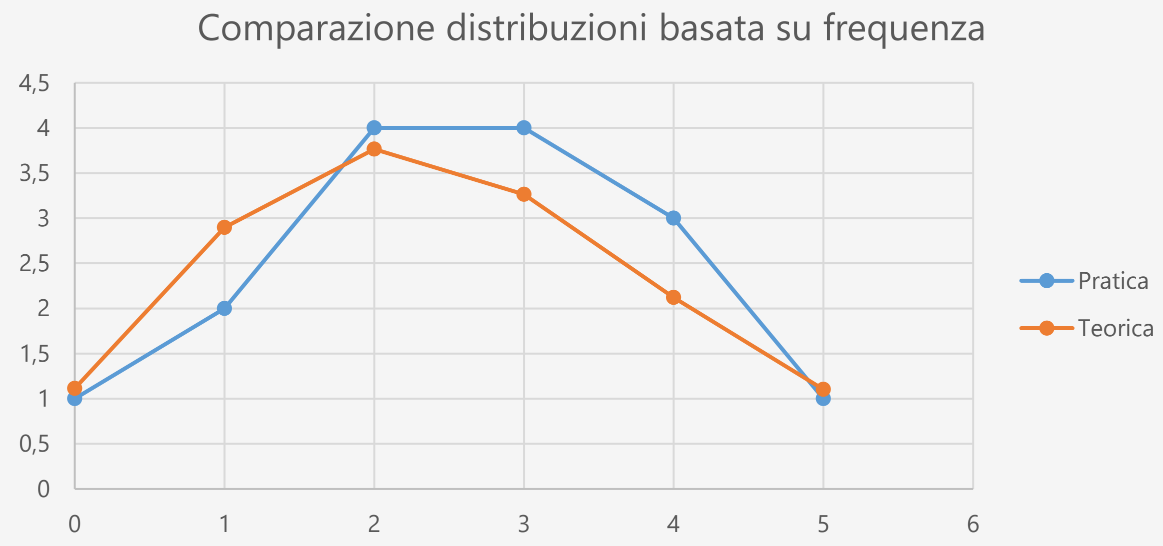
*I dati sperimentali sono soggetti ad errori di misurazione e variabili esterne non direttamente analizzabili, quindi è altamente probabile che la distribuzione sperimentale non rappresenti al 100% la reale.

# Analisi del numero di arrivi 3 (Kolmogorov-Smirnov)

Osservando la differenza assoluta tra le cumulative, il picco massimo è stato rilevato nella frequenza relativa al numero di arrivi pari ad 1.

Considerando le **15** osservazioni, il test di KS è valido se questo picco è minore a **0.304**. Quindi, con buona probabilità la distribuzione effettivamente analizzata corrisponde ad una **Poisson**.

| Differenza assoluta cumulative |
|---|
| 0,007606912 |
| 0,067384882 |
| 0,051762909 |
| 0,002668311 |
| 0,055909844 |
| 0,049037152 |
| |
| **0,067384882** |




A sinistra possiamo notare le distribuzioni messe a confronto.

# Analisi dei tempi di ordinazione 1

| Campione | Secondi impiegati |
|---|---|
| 1 | 14 |
| 2 | 40 |
| 3 | 64 |
| 4 | 32 |
| 5 | 153 |
| 6 | 39 |
| 7 | 75 |
| 8 | 103 |
| 9 | 97 |
| 10 | 93 |
| 11 | 55 |
| 12 | 161 |
| 13 | 51 |
| 14 | 64 |
| 15 | 35 |
| 16 | 40 |
| 17 | 21 |
| 18 | 52 |
| 19 | 82 |
| 20 | 165 |
| 21 | 65 |
| 22 | 11 |
| 23 | 21 |
| 24 | 33 |
| 25 | 38 |
| 26 | 32 |
| 27 | 15 |
| 28 | 39 |
| 29 | 13 |
| 30 | 62 |
| 31 | 13 |
| 32 | 80 |
| 33 | 39 |
| 34 | 18 |
| 35 | 121 |
| 36 | 32 |
| 37 | 38 |
| 38 | 18 |
| 39 | 34 |

A lato sono mostrati il tempo espresso in secondi, spesi dai vari clienti per ordinare.

Poiché abbiamo 39 campioni possiamo pensare di utilizzare il Goodness of Fit, per convalidare la distribuzione scelta.

Come usuale, cerchiamo prima di identificare la distribuzione candidata.

# Analisi dei tempi di ordinazione 2

Calcolando media e varianza ho riscontrato i seguenti valori.

| | |
|---|---|
| **Media** | 55,3333333 |
| **Varianza** | 1668,33333 |

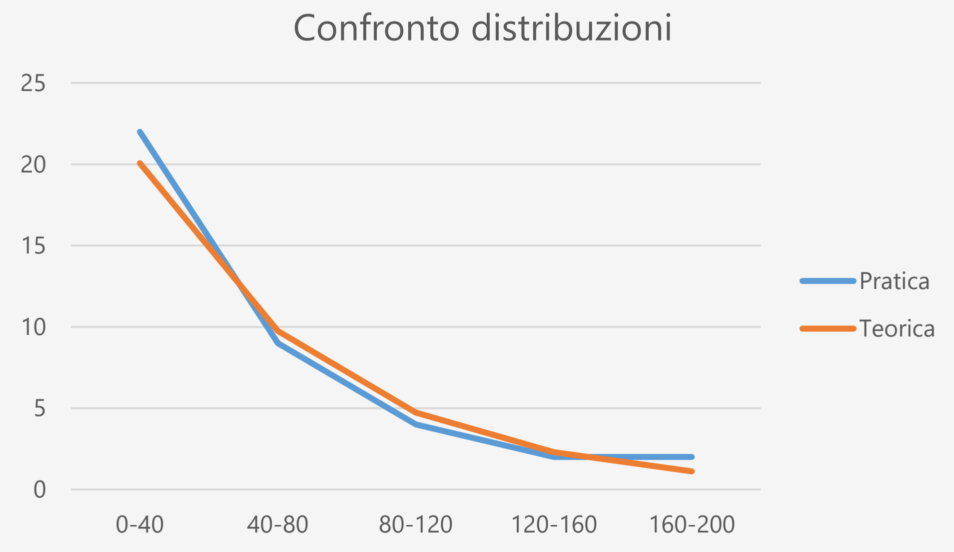
I valori ottenuti (varianza molto alta) hanno subito fatto pensare ad una distribuzione esponenziale. In realtà il valore teorico della varianza è poco più di un mezzo della varianza teorica, tuttavia considerando l’esiguo numero di prove e la fluttuazione significativa che la varianza ha in queste circostanze, ho ritenuto di dover comunque provare questa distribuzione.

# Analisi dei tempi di ordinazione 3

Come possiamo osservare, la distribuzione teorica si avvicina di molto a quella pratica, quindi possiamo ragionevolmente pensare che effettivamente siamo in presenza di una distribuzione esponenziale.

Per una completa visione dei dati si può far riferimento al file Excel.

| Classi | Frequenza pratica | Frequenza teorica | Errore chi |
|---|---|---|---|
| 0-40 | 22 | 20,07147431 | 0,185298363 |
| 40-80 | 9 | 9,74162608 | 0,056459696 |
| 80-200 | 8 | 8,136570223 | 0,002292296 |
| | | | |
| | 1 grado | P10-p90 accettato | **0,244050354** |



# Analisi dei tempi di servizio 1

| Campione | Secondi impiegati |
|---|---|
| 1 | 35 |
| 2 | 143 |
| 3 | 50 |
| 4 | 61 |
| 5 | 78 |
| 6 | 83 |
| 7 | 100 |
| 8 | 93 |
| 9 | 100 |
| 10 | 37 |
| 11 | 39 |
| 12 | 76 |
| 13 | 39 |
| 14 | 60 |
| 15 | 22 |
| 16 | 83 |
| 17 | 33 |
| 18 | 40 |
| 19 | 33 |
| 20 | 68 |
| 21 | 193 |
| 22 | 51 |
| 23 | 53 |
| 24 | 181 |
| 25 | 31 |
| 26 | 187 |
| 27 | 23 |
| 28 | 229 |
| 29 | 38 |
| 30 | 65 |
| 31 | 107 |
| 32 | 94 |
| 33 | 73 |
| 34 | 33 |
| 35 | 154 |
| 36 | 30 |
| 37 | 153 |
| 38 | 29 |
| 39 | 152 |

A lato sono mostrati il tempo espresso in secondi, spesi dai vari clienti per pagare e prendere l’ordine.

Poiché abbiamo 39 campioni possiamo pensare di utilizzare il Goodness of Fit, per convalidare la distribuzione scelta.

Come usuale, cerchiamo prima di identificare la distribuzione candidata.

# Analisi dei tempi di servizio 2

Calcolando media e varianza ho riscontrato i seguenti valori.

| | |
|---|---|
| **Media** | 80,7435897 |
| **Varianza** | 2986,30094 |

I valori ottenuti (varianza molto alta) hanno subito fatto pensare ad una distribuzione esponenziale. In realtà il valore teorico della varianza è poco meno di un mezzo della varianza teorica, tuttavia considerando l’esiguo numero di prove e la fluttuazione significativa che la varianza ha in queste circostanze, ho ritenuto di dover comunque provare questa distribuzione.

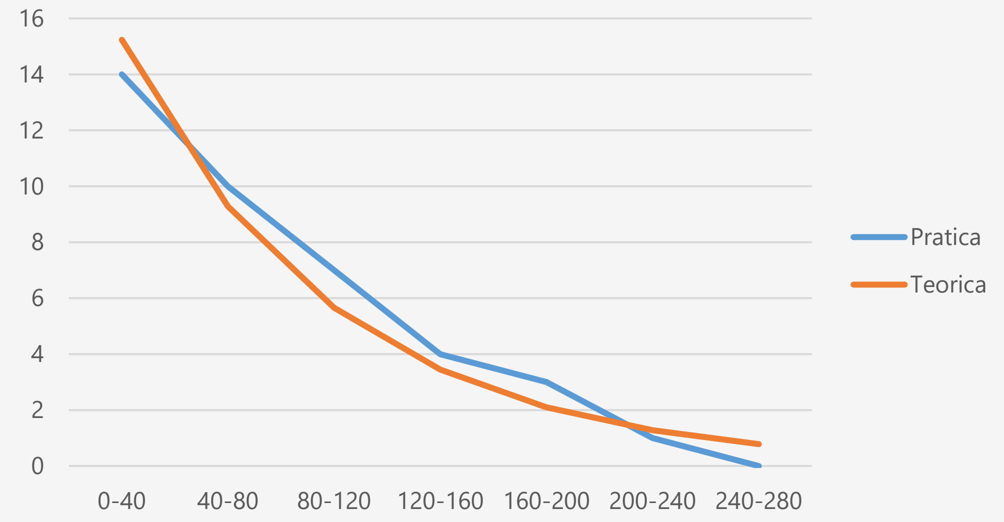
# Analisi dei tempi di servizio 3

Come possiamo osservare, la distribuzione teorica si avvicina di molto a quella pratica, quindi possiamo ragionevolmente pensare che effettivamente siamo in presenza di una distribuzione esponenziale.

Per una completa visione dei dati si può far riferimento al file Excel.

| Classi | Frequenza pratica | Frequenza teorica | Errore chi |
|---|---|---|---|
| 0-40 | 14 | 15,23613184 | 0,100289361 |
| 40-80 | 10 | 9,283831496 | 0,055246299 |
| 80-120 | 7 | 5,656916607 | 0,318879193 |
| 120-160 | 4 | 3,446928729 | 0,08874214 |
| 160-320 | 4 | 4,635077656 | 0,087015506 |
| | | | |
| | 3 gradi | P10-p90 accett. | **0,650172499** |

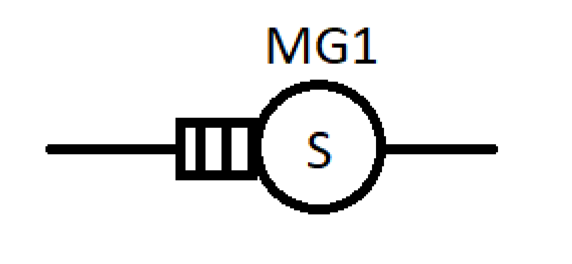
Confronto distribuzioni



# Riflessioni sulle distribuzioni e sul modello

Analizzando le varie distribuzioni, ho dimostrato che il numero di arrivi segue una distribuzione poissoniana, quindi con tempo di interrarrivi esponenziali. Per quanto riguarda i servizi ho invece due distribuzioni esponenziali rispetto al tempo di servizio.

Il problema non può essere modellato con due MM1 in quanto la coda è unica. (le code non sono indipendenti). Considerando tuttavia i tempi esponenziali di servizio, il modello può essere ridotto ad una MG1, ove il servizio è determinato da una 2-erlagiana generalizzata, chiamata anche hypoesponenziale a due parametri.



# Hypoesponenziale a due parametri

La distribuzione hypoesponenziale è una generalizzazione della k-erlangiana. La k-erlangiana nella fattispecie rappresenta una serie di k distribuzioni esponenziali con parametro $\lambda$ fissato.

La distribuzione hypoesponenziale generalizza la k-erlangiana in quanto concatena k distribuzioni esponenziali con parametro $\lambda$ variabile.

La distribuzione hypoesponenziale a due parametri è composta da 2 distribuizioni esponenziali con parametri $\lambda_1$ e $\lambda_2$.

Il primo parametro è dato dal tasso di servizio dell'operatore delle ordinazioni, mentre il secondo parametro è dato dal tasso di servizio dell'operatore del pagamento e della consegna dell'ordine.

# Modello teorico

Di seguito svilupperemo il modello teorico della MG1. Partiamo dai parametri. Ho scelto di svolgere i calcoli in minuti.

$$\lambda = 0.26 \text{ min}^{-1}$$

$$\mu = \frac{1}{\frac{1}{\mu_1} + \frac{1}{\mu_2}} = \frac{1}{\frac{1}{1.09} + \frac{1}{0.75}} \approx 0.44\ min^{-1}$$

$$\rho = \frac{\lambda}{\mu} = \frac{0.26}{0.44} \approx 0.59 < 1$$

Ove $\lambda$ è il tasso di interarrivo, $\mu_1$ e $\mu_2$ sono i tassi di servizio rispettivamente del primo e secondo servente.

$\mu$ è invece il tasso di servizio totale.

# Modello teorico

Per poter calcolare il tempo di risposta medio dobbiamo affidarci all'equazione di Khintchine-Pollaczek e al teorema di Little. Nella fattispecie abbiamo:

$$N = \lambda R \; \rightarrow R = \frac{N}{\lambda} \; (Little)$$

$$N = \rho + \frac{\rho^2(1+\mu^2\sigma^2)}{2(1-\rho)}$$

Da queste due relazioni segue che:

$$R = \frac{\rho + \frac{\rho^2(1+\mu^2\sigma^2)}{2(1-\rho)}}{\lambda}$$

# Modello teorico

La varianza di una hypoesponenziale a due parametri è uguale alla somma delle varianze delle esponenziali con parametri $\lambda_1, \lambda 2$, ovvero:

$$\sigma^2 = \frac{1}{\lambda_1^2} + \frac{1}{\lambda_2^2}$$

Nel nostro caso $\lambda_1 = \mu_1, \lambda_2 = \mu_2$, quindi:

$$\sigma^2 = \frac{1}{1.09^2} + \frac{1}{0.75^2} = 0.842 + 1.78 \approx 2.662$$

Avremo quindi che:

$$R = \frac{0.59 + \frac{0.59^2(1 + 0.44^2 * 2.662^2)}{2(1 - 0.59)}}{0.26} \approx 4.77\ minuti$$

$$N = \lambda R = 4.77 * 0.26 \approx 1.24$$

Utilizzeremo questo risultato per convalidare il simulatore.

# Il simulatore

Il simulatore è stato scritto sulla falsa riga del simulatore visto a lezione. Non ho usato C++ ma Python.

Si tratta di un simulatore ad eventi che genera progressivamente eventi appartenenti a tre tipologie: «**ARRIVAL**» (arrivi), «**DEPARTURE**» (partenze) e «**SIM_END**» (fine simulazione). I tempi sono ovviamente dettati dai parametri precedentemente discussi.

La simulazione per un tempo stabilito dall'analista. Una volta terminata, vengono restituite le statistiche con le informazioni ritenute più significative.

Di interesse particolare è la gestione della generazione delle variabili. Dobbiamo infatti generare tempi esponenziali (per gli interarrivi) e tempi hypoesponenziali per il servizio.

Vediamo come.

# Il simulatore

- **Generazione dei tempi esponenziali**

Generare tempi esponenziali non è particolarmente difficile. Si utilizza la tecnica della trasformazione inversa e si ottiene l'equazione:

$$t = -\frac{\ln(1 - random(0,1))}{\lambda}$$

- **Generazione dei tempi Hypoesponenziali**

La cumulative della hypoesponenziale è definita in questo modo:

$$F(x) = 1 - \frac{\lambda_2}{\lambda_2 - \lambda_1} e^{-\lambda_1 x} + \frac{\lambda_1}{\lambda_2 - \lambda_1} e^{-\lambda_2 x}$$

La funzione è invertibile ma l'inversa non ha in generale una formula chiusa, quindi non possiamo utilizzare la tecnica della trasformazione inversa.

# Il simulatore

Per risolvere il problema ho utilizzato la tecnica del rifiuto. Generiamo punti seguendo una distribuzione uniforme nel piano euclideo. Se il punto si trova all'interno della funzione di densità della hypoesponenziale, allora restituiamo la coordinata x, altrimenti generiamo un altro punto fino a quando non otteniamo un valore buono.

La funzione di densità della hypoesponenziale è:

$$f(x) = \frac{\lambda_1 \lambda_2}{\lambda_1 - \lambda_2}(e^{-\lambda_2 x} - e^{-\lambda_1 x})$$

**Come generare le coordinate dei punti?**

Dobbiamo scegliere il range uniforme di valori con cui scegliere le coordinate dei punti. Se scegliamo male il range, potremmo perderci parte della distribuzione di probabilità oppure potremmo definire uno spazio troppo grande, con un conseguenze incremento del numero di fallimenti.

# Il simulatore

**Generare la coordinata y per l'hypoesponenziale**

Dobbiamo generare un valore compreso tra $[0, y_{max}]$ scegliendo opportunatamente $y_{max}$. Sappiamo che l'hypoesponenziale è una funzione che ha un unico punto di massimo. Ci basta quindi derivare la funzione di densità e porla uguale a 0.

$$f'(x) = 0$$

Facendo un po' di calcoli si ottiene:

$$x_{\max y} = \frac{\ln(\lambda_2) - \ln(\lambda_1)}{\lambda_2 - \lambda_1}$$

Il punto massimo sarà quindi:

$$y_{max} = f(x_{\max y})$$

# Il simulatore

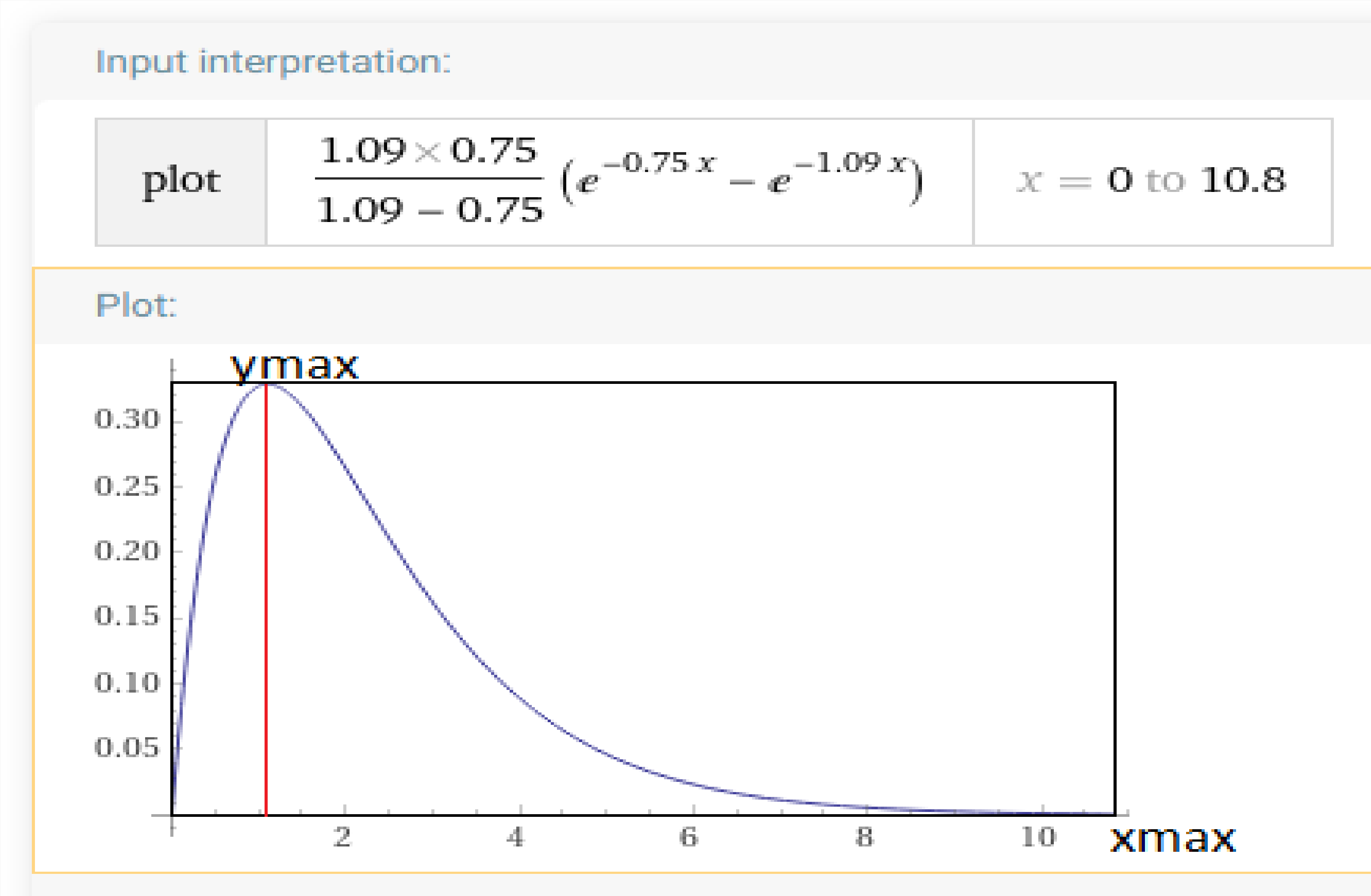
**Generare la coordinata x per l'hypoesponenziale**

Dobbiamo generare un valore compreso tra $[0, x_{max}]$ scegliendo opportunatamente $x_{max}$. Un metodo semplice per farlo è considerare la coordinata $x$ del punto di massimo ed incrementare di un certo scarto $\epsilon > 0$, verso destra fino a quando la funzione cumulativa non sarà vicina all'1.

Un esempio potrebbe essere prendere $\epsilon = \frac{x_{maxy}}{100}$.

Come threshold per interrompere la ricerca della $x_{max}$ ho scelto $F(x) \geq 0.999$ $(99.9\%)$

# Il simulatore




Funzione di densità con parametri utilizzati nel simulatore.

$x_{max}$ e $y_{max}$ sono stati calcolati con i metodi precedentemente descritti.

# Il simulatore

```python
class ExpRandom:
  def __init__(self, lam):
    self.lam = lam

  def generate(self):
    return ((-np.log(1-(np.random.uniform(low=0.0,high=1.0)))) / self.lam)
```

La classe ExpRandom genera tempi secondo una distribuzione esponenziale di parametro $\lambda$.

La classe HypoExpRandom genera tempi secondo una distribuzione Hypoesponenziale di parametro $\lambda_1$, $\lambda_2$.

```python
class HypoExpRandom:
  def __init__(self, lam1, lam2):
    self.lam1 = lam1
    self.lam2 = lam2
    x = (np.log(lam1) - np.log(lam2)) / (lam1 - lam2)
    self.maxy = self.__calculate(x)

    increment = x/100
    while self.__cumulative(x) < 0.999:
      x += increment

    self.maxx = x

  def generate(self):
    while True:
      x = np.random.uniform(low=0.0,high=self.maxx)
      y = np.random.uniform(low=0.0,high=self.maxy)
      if y <= self.__calculate(x):
        return x

  def __calculate(self, x):
    return ((self.lam1*self.lam2) / (self.lam1 - self.lam2)) \
    * (np.exp(-x*self.lam2) - np.exp(-x*self.lam1))

  def __cumulative(self, x):
    return 1 - (self.lam2 / (self.lam2 - self.lam1)) * np.exp(-x*self.lam1) \
    + (self.lam1 / (self.lam2 - self.lam1)) * np.exp(-x*self.lam2)
```

# Test con il simulatore

```
simulation = MH1Simulator(0.26, 1.09, 0.75, sim_end_time = 30000)
simulation.start()
```

```
Risultati della simulazione
Arrivi: 7757
Partenze: 7753
Throughput: 0.25843333333333335
Utilizzazione: 0.5838703418689638
Numero medio di clienti nel sistema: 1.191652639794207
Tempo medio di permanenza nel sistema: 4.611064000235549
Fine della simulazione
```

```
simulation = MH1Simulator(0.26, 1.09, 0.75, sim_end_time = 100000)
simulation.start()
```

```
Risultati della simulazione
Arrivi: 26184
Partenze: 26184
Throughput: 0.26184
Utilizzazione: 0.5948211793655365
Numero medio di clienti nel sistema: 1.2435758568676074
Tempo medio di permanenza nel sistema: 4.749373116665167
Fine della simulazione
```

```
simulation = MH1Simulator(0.26, 1.09, 0.75, sim_end_time = 300000)
simulation.start()
```

```
Risultati della simulazione
Arrivi: 78291
Partenze: 78291
Throughput: 0.26097
Utilizzazione: 0.5846305070509059
Numero medio di clienti nel sistema: 1.2097458114938042
Tempo medio di permanenza nel sistema: 4.635574247974112
Fine della simulazione
```

```
simulation = MH1Simulator(0.26, 1.09, 0.75, sim_end_time = 500000)
simulation.start()
```

```
Risultati della simulazione
Arrivi: 129803
Partenze: 129801
Throughput: 0.259602
Utilizzazione: 0.581729541535794
Numero medio di clienti nel sistema: 1.2034041207013568
Tempo medio di permanenza nel sistema: 4.635573380410616
Fine della simulazione
```

# Test con il simulatore

Come possiamo osservare il simulatore ha un comportamento molto simile al modello teorico calcolato. Nella fattispecie abbiamo i seguenti risultati:

$$R_{teorico} \approx 4.77$$

$$R_{simulato} \approx 4.65$$

$$R_{pratico} \approx 4.72$$

$$N_{teorico} \approx 1.24$$

$$N_{simulato} \approx 1.2$$

$$\rho_{teorico} \approx 0.59$$

$$\rho_{simulato} \approx 0.588$$

Per accertarci che i risultati non siano soltanto un caso, andiamo ad analizzare l'andamento del tempo di risposta con il test delle prove ripetute.

# Test con il simulatore

Il test delle prove ripetute è stato svolto prendendo in considerazione 3300 campioni. Da questi è stato eliminata l'influenza dello stato iniziale e sono stati presi infine 2500 campioni, suddivisi in 50 categorie.

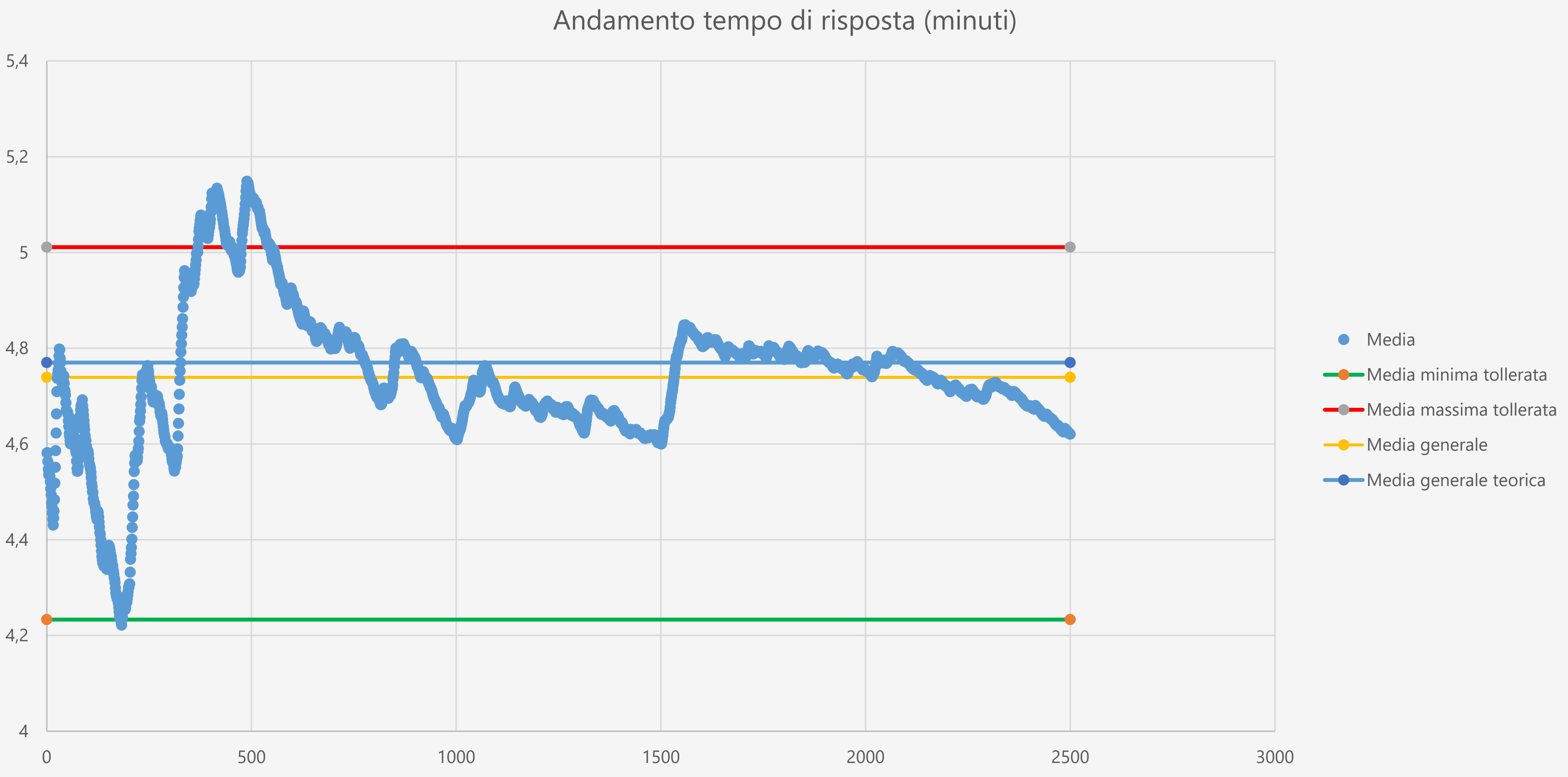
Ogni categorie è stata utilizzata per produrre le $y_j$, dalle quali sono stati ottenuti i due valori della media del tempo di risposta con intervallo di confidenza pari al 90% ( $u_{\frac{\alpha}{2}} = 1.645$).

$$y^{-} = \frac{1}{p}\sum^{p} y_j$$

$$s_y^2 = \frac{1}{p-1}\sum^{p}\left(y_j - y^{-}\right)^2$$

$$(\delta_{-}, \delta^{-}) = \left(y^{-} - \frac{s_y}{\sqrt{p}} u_{\frac{\alpha}{2}}, y^{-} + \frac{s_y}{\sqrt{p}} u_{\frac{\alpha}{2}}\right)$$

# Il simulatore



# Miglioramento del modello

Il tempo di risposta medio con il seguente modello è stato comunque inferiore ai 5 minuti. Il responsabile del McDrive quindi non ha bisogno di cambiare il modello già implementato. A fini puramente pratici proveremo con il simulatore a vedere cosa succede utilizzando 2 MG1 o 3 MG1, seguendo lo stesso modello discusso precedentemente. Assumiamo che i tempi di servizio siano gli stessi e che gli utenti si dividano ora equamente tra le 2(3) code.

```
Risultati della simulazione
Arrivi: 1950
Partenze: 1949
Throughput: 0.12993333333333335
Utilizzazione: 0.2875722608593525
Numero medio di clienti nel sistema: 0.37118251861161194
Tempo medio di permanenza nel sistema: 2.856715125281775
Fine della simulazione
```

Utilizzando 2 MG1 possiamo notare come il tempo di servizio crolli a soli 2.85 minuti. Gli utenti guadagnano quindi in media 2 minuti, grazie alla riduzione della coda effettuata.

```
Risultati della simulazione
Arrivi: 26870
Partenze: 26870
Throughput: 0.08956666666666667
Utilizzazione: 0.20183591196145653
Numero medio di clienti nel sistema: 0.23993060174577555
Tempo medio di permanenza nel sistema: 2.678793469435529
Fine della simulazione
```

Utilizzando 3 MG1 non si nota un reale guadagno rispetto alla 2 MG1. Ciò a significare che il tempo speso è ormai intrinseco alla velocità di servizio dei dipendenti e non dipende dal sistema in sé.