Discrete mathematical models

Lecture 8. Poisson process and Steady-State Behavior, estimation of absolute performance

Процессы Пуассона и устойчивое поведение системы; оценка абсолютных показателей

Poisson process and steady-state behavior of the system

Пуассоновский процесс и стационарное поведение системы

Poisson process
Процесс Пуассона

Consider random events such as the arrival of jobs at a job shop, the arrival of e-mail to a mail server, the arrival of boats to a dock, the arrival of calls to a call center, the breakdown of machines in a large factory, and so on. These events may be described by a counting function $N(t)$ defined for all $t > 0$. This counting function will represent the number of events that occurred in $[0, t]$. Time zero is the point at which the observation began, regardless of whether an arrival occurred at that instant. For each interval [0, t], the value $N(t)$ *is* an observation of a random variable where the only possible values that can be assumed by $N(t)$ are the integers $0,1,2,\ldots$.

The counting process, $\{N(t),\ t > 0\}$, is said to be a Poisson process with mean rate λ if the following assumptions are fulfilled:

a. Arrivals occur one at a time.

b. $\{N(t),\ t > 0\}$ has stationary increments: The distribution of the number of arrivals between t and $t + s$ depends only on the length of the interval s, not on the starting point t. Thus, arrivals are completely at random without rush or slack periods.

c. $\{N(t),\ t > 0\}$ has independent increments: The number of arrivals during nonoverlapping time intervals are independent random variables. Thus, a large or small number of arrivals in one time interval has no effect on the number of arrivals in subsequent time intervals. Future arrivals occur completely at random, independent of the number of arrivals in past time intervals.

Properties (1/2)
Свойства процесса

If arrivals occur according to a Poisson process, meeting the three preceding assumptions, it can be shown that the probability that ***N(t)*** is equal to n is given by

$$P(N(t)=n)=\frac{e^{-\lambda t}(\lambda t)^n}{n!}\ \ for\ t\geq 0 \text{ and } n=0,1,2..$$

It can be seen that ***N(t)*** has the Poisson distribution with parameter $\alpha=\lambda t$. Thus, its mean and variance are given by

$$E[N(t)]=\alpha=\lambda t=V[N(t)]$$

For any times ***s*** and ***t*** such that ***s < t,*** the assumption of stationary increments implies that the random variable ***N(t) — N(s),*** representing the number of arrivals in the interval from ***s*** to ***t,*** is also Poisson-distributed with mean $\lambda(t-s)$. Thus,

$$P(N(t)-N(s)=n)=\frac{e^{-\lambda(t-s)}(\lambda(t-s))^n}{n!}\ \ for\ n=0,1,2..$$

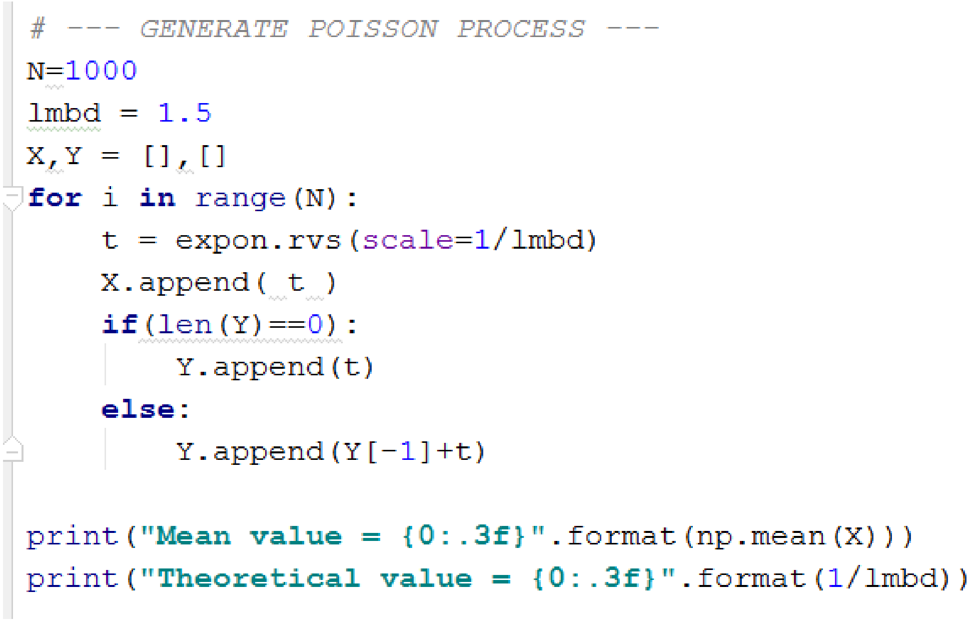
Properties (2/2)
Свойства процесса

Exponential distribution is memoryless — that is, the probability of a future arrival in a time interval of length ***s*** is independent of the time of the last arrival. The probability of the arrival depends only on the length of the time interval ***s***. Thus, the memoryless property is related to the properties of independent and stationary increments of the Poisson process.

Example:

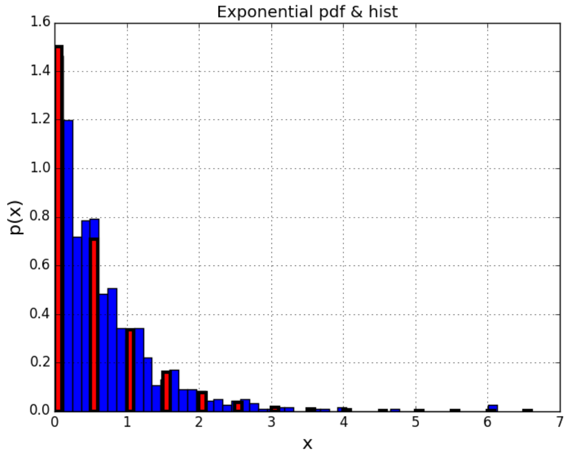
The jobs at a machine shop arrive according to a Poisson process with a mean of $\lambda = 2$ jobs per hour. Therefore, the interarrival times are distributed exponentially, with the expected time between arrivals being $E(A) = \frac{1}{\lambda} = 1/2$ hour.

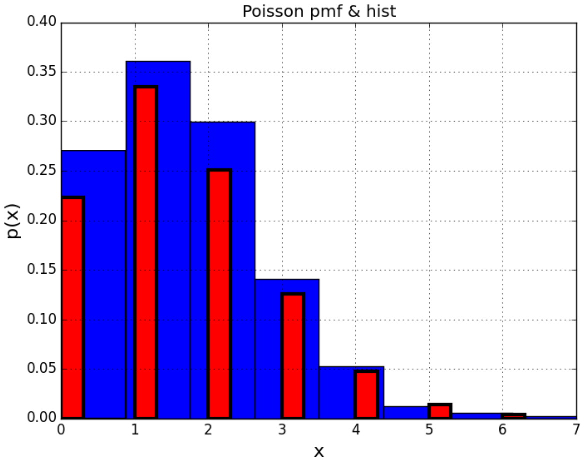
Example
Пример



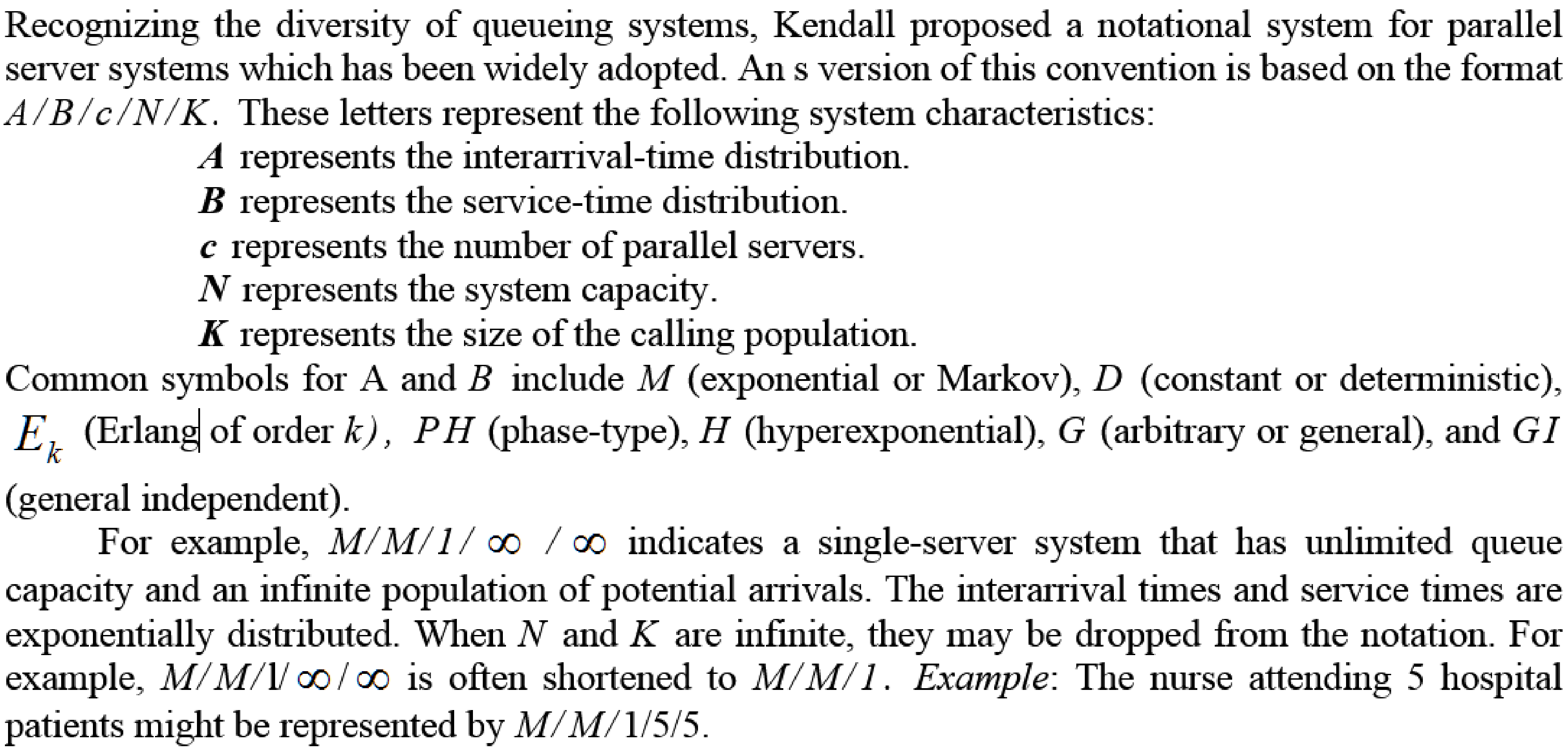
```
# --- GENERATE POISSON PROCESS ---
N=1000
lmbd = 1.5
X,Y = [],[]
for i in range(N):
    t = expon.rvs(scale=1/lmbd)
    X.append( t )
    if(len(Y)==0):
        Y.append(t)
    else:
        Y.append(Y[-1]+t)

print("Mean value = {0:.3f}".format(np.mean(X)))
print("Theoretical value = {0:.3f}".format(1/lmbd))
```





Queueing Notation
Нотация для очередей



Recognizing the diversity of queueing systems, Kendall proposed a notational system for parallel server systems which has been widely adopted. An s version of this convention is based on the format $A/B/c/N/K$. These letters represent the following system characteristics:

$\boldsymbol{A}$ represents the interarrival-time distribution.
$\boldsymbol{B}$ represents the service-time distribution.
$\boldsymbol{c}$ represents the number of parallel servers.
$\boldsymbol{N}$ represents the system capacity.
$\boldsymbol{K}$ represents the size of the calling population.

Common symbols for A and B include M (exponential or Markov), D (constant or deterministic), E_k (Erlang of order k), PH (phase-type), H (hyperexponential), G (arbitrary or general), and GI (general independent).

For example, $M/M/1/\infty/\infty$ indicates a single-server system that has unlimited queue capacity and an infinite population of potential arrivals. The interarrival times and service times are exponentially distributed. When N and K are infinite, they may be dropped from the notation. For example, $M/M/1/\infty/\infty$ is often shortened to $M/M/1$. *Example*: The nurse attending 5 hospital patients might be represented by $M/M/1/5/5$.

Пример: медсестра, обслуживающая 5 пациентов в больнице описывается как система M/M/1/5/5

Reminder for measure of performance
Метрики производительности (напоминание)

P_n - Steady - state probability of having n customers in system

$P_n(t)$ - Probability of n customers in system at time t

λ - Arrival rate

λ_e - Effective arrival rate

μ - Service rate of one server

ρ - Server utilization

A_n - Interarrival time between customers n - 1 and n

S_n - Service time of the n - th arriving customer

W_n - Total time spent in system by the n - th arriving customer

$\mathrm{W}_\mathrm{n}^\mathrm{Q}$ - Total time spent waiting in queue by customer n

L(t) - The number of customers in system at time t

$\mathrm{L}_\mathrm{Q}(t)$ - The number of customers in queue at time t

L - Long - run time - average number of customers in system

L_Q - Long - run time - average number of customers in queue

w - Long - run average time spent in system per customer

w_Q - Long - run average time spent in queue per customer

Estimated values
Оцениваемые величины

The most interesting parameters // Наиболее интересные параметры:

ρ - Server utilization

L - Long-run time-average number of customers in system

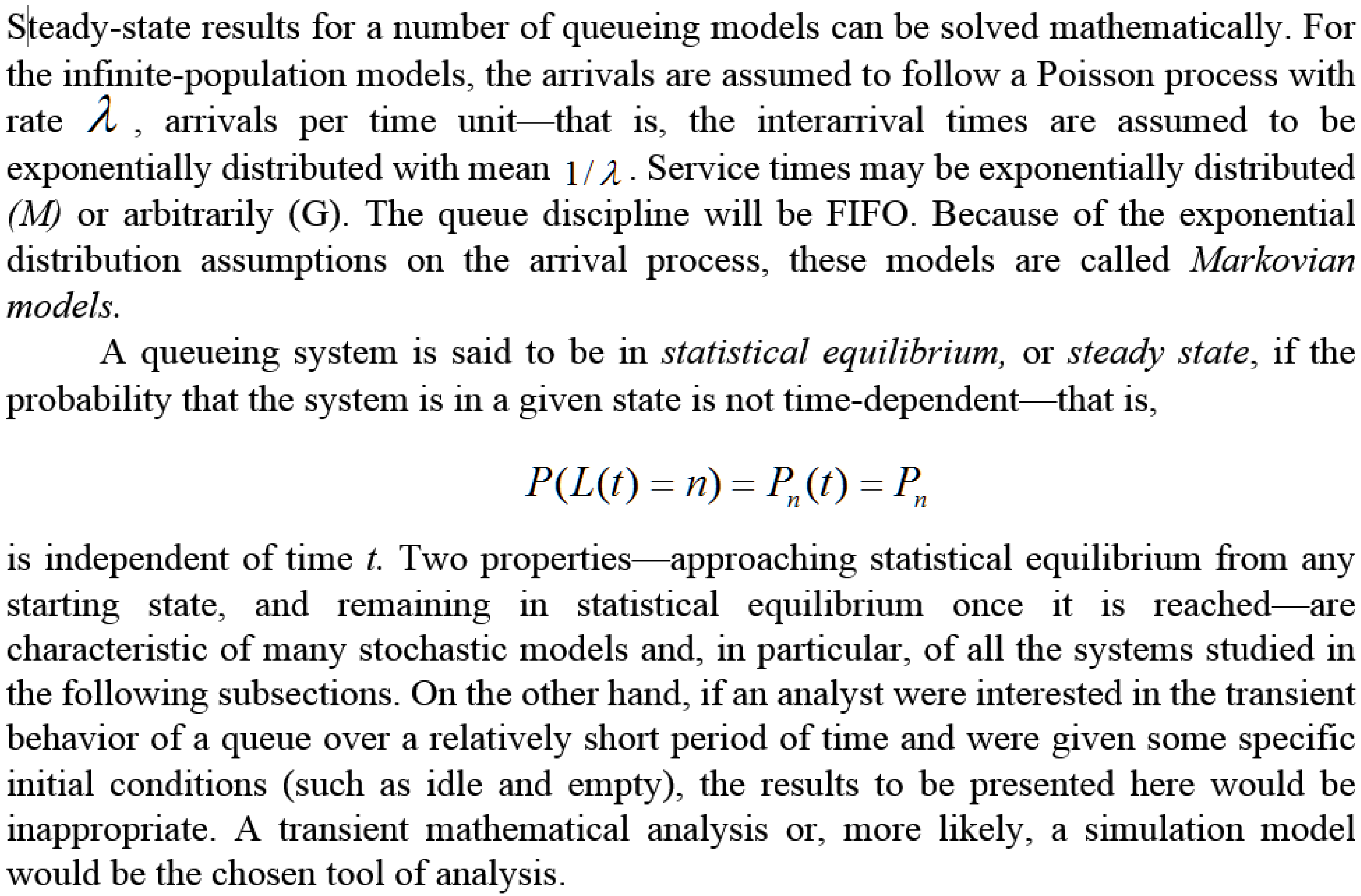
L_Q - Long-run time-average number of customers in queue

w - Long-runaverage time spent in system per customer

w_Q - Long-run average time spent in queue per customer

Other interesting parameters: for example, the number of customers waiting for service over a certain period of time or of the time when the queue was more than a certain size.

Steady-State Behavior of Infinite-Population Markovian Models // Оценка устойчивости



Steady-state results for a number of queueing models can be solved mathematically. For the infinite-population models, the arrivals are assumed to follow a Poisson process with rate λ , arrivals per time unit—that is, the interarrival times are assumed to be exponentially distributed with mean $1/\lambda$. Service times may be exponentially distributed *(M)* or arbitrarily (G). The queue discipline will be FIFO. Because of the exponential distribution assumptions on the arrival process, these models are called *Markovian models*.

A queueing system is said to be in *statistical equilibrium,* or *steady state*, if the probability that the system is in a given state is not time-dependent—that is,

$$P(L(t) = n) = P_n(t) = P_n$$

is independent of time *t*. Two properties—approaching statistical equilibrium from any starting state, and remaining in statistical equilibrium once it is reached—are characteristic of many stochastic models and, in particular, of all the systems studied in the following subsections. On the other hand, if an analyst were interested in the transient behavior of a queue over a relatively short period of time and were given some specific initial conditions (such as idle and empty), the results to be presented here would be inappropriate. A transient mathematical analysis or, more likely, a simulation model would be the chosen tool of analysis.

Steady-State Behavior (1/2)
Теоретические оценки для устойчивого поведения

For relatively simple systems of a parameter L (long-term average number of customers in the system) can be calculated by

$$L = \sum_{n=0}^{\infty} nP_n$$

where $\{P_n\}$ - steady-state distribution of the number of customers in the system. Knowing L we can calculate other important parameters :

$$w = \frac{L}{\lambda}$$

$$w_Q = w - \frac{1}{\mu}$$

$$L_Q = \lambda w_Q$$

Conditions of stability: the coefficient of utilization ($\rho = \frac{\lambda}{c\mu}$) <1

One server with a Poisson arrival of customers (M / G / 1) // Система с одним сервером

In general, there is no solution for the probabilities $P_0, P_1, P_2 \cdots$

In case $\lambda < \mu$ we can write the following expressions:

$$\rho = \lambda / \mu$$

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

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

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

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

$$P_0 = 1 - \rho$$

Example // Пример

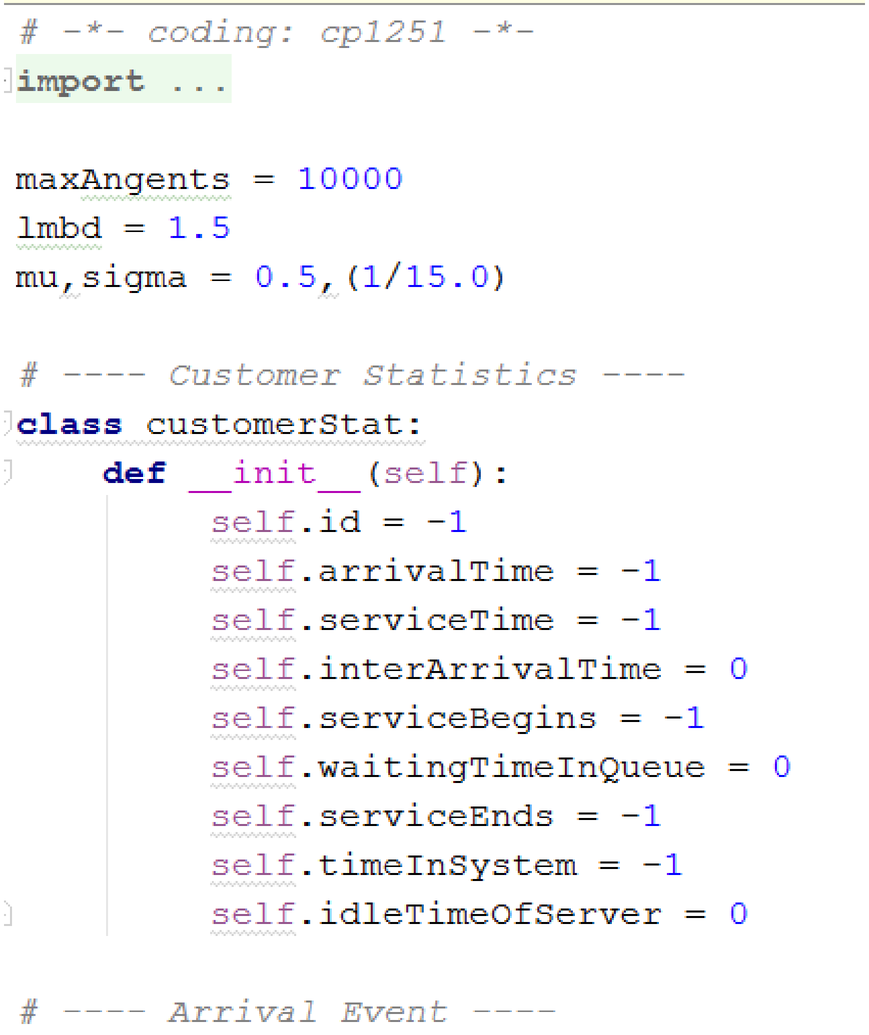
Customers arrive at a walk-in shoe repair shop apparently at random. It is assumed that arrivals occur according to a Poisson process at the rate $\lambda = 1.5$ per hour. Observation over several months has found that shoe repair times by the single worker take an average time of 30 minutes, with a standard deviation of 20 minutes. Thus the mean service time $1/\mu = 1/2$ hour, the service rate is $\mu = 2$ per hour and $\sigma^2 = (20)^2$ minutes^2 = 1/9 hour^2. The "customers" are the people needing shoe repair, and the appropriate model is the M/G/1 queue, because only the mean and variance of service times are known, not their distribution.

It is easy to calculate the employee's utilization (the server): $\rho = \lambda / \mu = 1.5 / 2 = 0.75$

Knowing the utilization is easy to calculate L:

$$L = 0.75 + \frac{(1.5)^2 \left[0.5^2 + 1/9 \right]}{2(1 - 0.75)} = 0.75 + 1.625 = 2.375$$

Comparison with simulation
Сравнение модели и теоретической оценки



```
# -*- coding: cp1251 -*-
import ...

maxAngents = 10000
lmbd = 1.5
mu,sigma = 0.5,(1/15.0)

# ---- Customer Statistics ----
class customerStat:
    def __init__(self):
        self.id = -1
        self.arrivalTime = -1
        self.serviceTime = -1
        self.interArrivalTime = 0
        self.serviceBegins = -1
        self.waitingTimeInQueue = 0
        self.serviceEnds = -1
        self.timeInSystem = -1
        self.idleTimeOfServer = 0

# ---- Arrival Event ----
```

Average time a customer spends in the system: 1.27

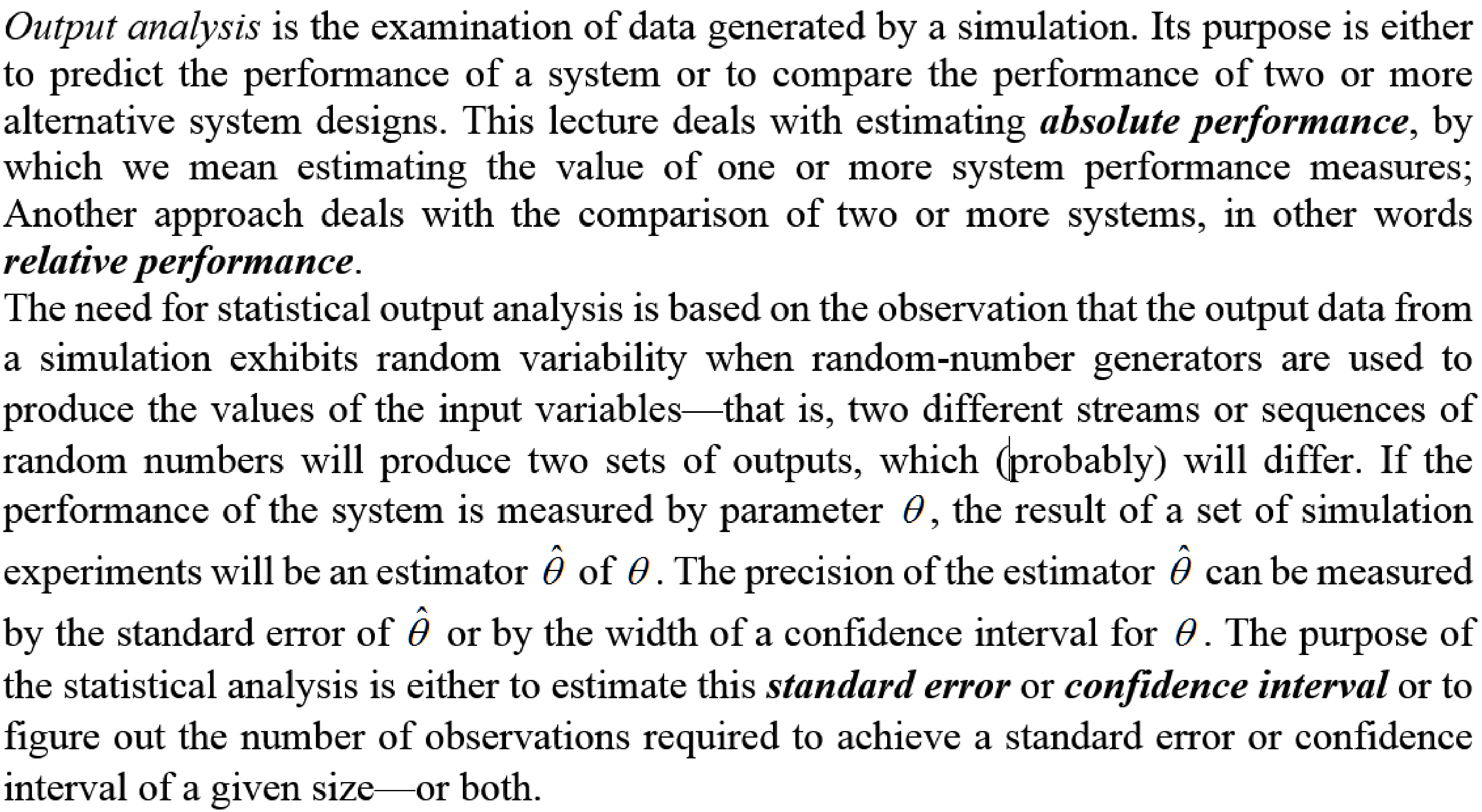
Average time a customer spends in the system (alternative): 1.27

Average time a customer spends in the system (theoretical): 1.26

…

Estimation of absolute performance
Оценка абсолютной производительности

Performance analysis and its goals
Анализ производительности и его цели



Output analysis is the examination of data generated by a simulation. Its purpose is either to predict the performance of a system or to compare the performance of two or more alternative system designs. This lecture deals with estimating ***absolute performance***, by which we mean estimating the value of one or more system performance measures; Another approach deals with the comparison of two or more systems, in other words ***relative performance***.

The need for statistical output analysis is based on the observation that the output data from a simulation exhibits random variability when random-number generators are used to produce the values of the input variables—that is, two different streams or sequences of random numbers will produce two sets of outputs, which (probably) will differ. If the performance of the system is measured by parameter θ, the result of a set of simulation experiments will be an estimator $\hat{\theta}$ of θ. The precision of the estimator $\hat{\theta}$ can be measured by the standard error of $\hat{\theta}$ or by the width of a confidence interval for θ. The purpose of the statistical analysis is either to estimate this ***standard error*** or ***confidence interval*** or to figure out the number of observations required to achieve a standard error or confidence interval of a given size—or both.

Autocorrelation effect
Эффект автокорреляции

Consider a typical output variable *Y*, the total cost per week of an inventory system; *Y* should be treated as a random variable with an unknown distribution. A simulation run of length 1 week provides a single sample observation from the population of all possible observations on *Y*. By increasing the run length, the sample size can be increased to *n* observations, $Y_1, Y_2 \ldots Y_n$, based on a run length of *n* weeks. However, these observations do not constitute a random sample, in the classic sense, because they are not statistically independent. In this case, the inventory on hand at the end of one week is the beginning inventory on hand for the next week, and so the value of Y_i has some influence on the value of Y_{i+1}. Thus, the sequence of random variables $Y_1, Y_2 \ldots Y_n$ could be *autocorrelated* (i.e., correlated with itself). The methods must be properly modified and the simulation experiments properly designed for valid inferences to be made.

In addition to the autocorrelation present in most simulation output data, the specification of the initial conditions of the system at time 0 can pose a problem for the simulation analyst and could influence the output data. By "time 0" we mean whatever point in time the beginning of the simulation run represents. For example, the inventory on hand and the number of backorders at time 0 (Monday morning) would most likely influence the value of Y_1, the total cost for week 1. Because of the autocorrelation, these initial conditions would also influence the costs $Y_1, Y_2 \ldots Y_n$ for subsequent weeks. The specified initial conditions, if not chosen well, can have an effect on estimation of the steady-state (long-run) performance of a simulation model. For purposes of statistical analysis, the effect of the initial conditions is that the output observations might not be identically distributed and that the initial observations might not be representative of the steady-state behavior of the system.

Types of Simulations with Respect to Output Analysis
Типы моделирования в отношении анализа результатов

In analyzing simulation output data, a distinction is made between ***terminating*** or ***transient*** simulations and ***steady-state*** simulations.

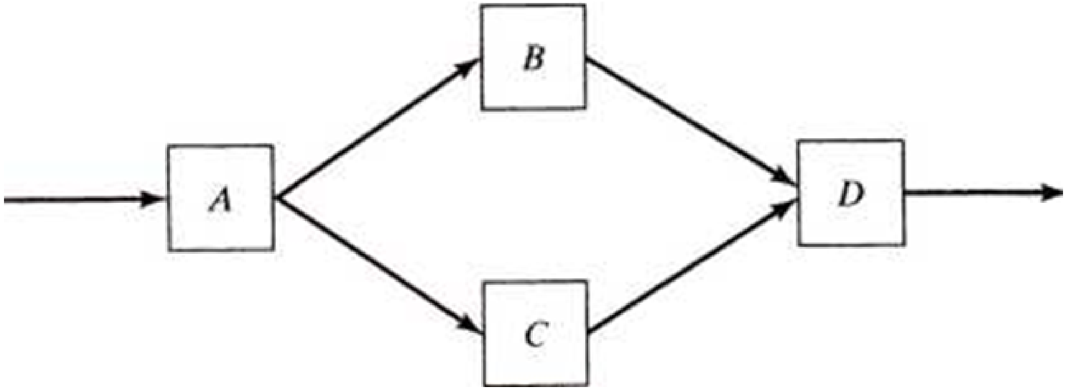
A terminating simulation is one that runs for some duration of time T_E, where *E* is a specified event (or set of events) that stops the simulation. Such a simulated system opens at time 0 under well-specified initial conditions and closes at the stopping time T_E.

Example 1. The Shady Grove Bank opens at 8:30 A.M. (time 0) with no customers present and 8 of the 11 tellers working (initial conditions) and closes at 4:30 P.M. (time T_E = 480 minutes). Here, the event *E* is merely the fact that the bank has been open for 480 minutes. The simulation analyst is interested in modeling the interaction between customers and tellers over the entire day, including the effect of starting up and closing down at the end of the day.

Example 2. Consider the Shady Grove Bank of Example 1, but restricted to the period from 11:30 A.M. (time 0) to 1:30 P.M., when it is especially busy. The simulation run length is T_E = 120 minutes. The initial conditions at time 0 (11:30 A.M.) could be specified in essentially two ways: (a) the real system could be observed at 11:30 on a number of different days and a distribution of number of customers in system (at 11:30 A.M.) could be estimated, then these data could be used to load the simulation model with customers at time 0; or (b) the model could be simulated from 8:30 A.M. to 11:30 A.M. without collecting output statistics, and the ending conditions at 11:30 A.M. used as initial conditions for the 11:30 A.M. to 1:30 P.M. simulation.

Transient system example
Пример систем переходного типа

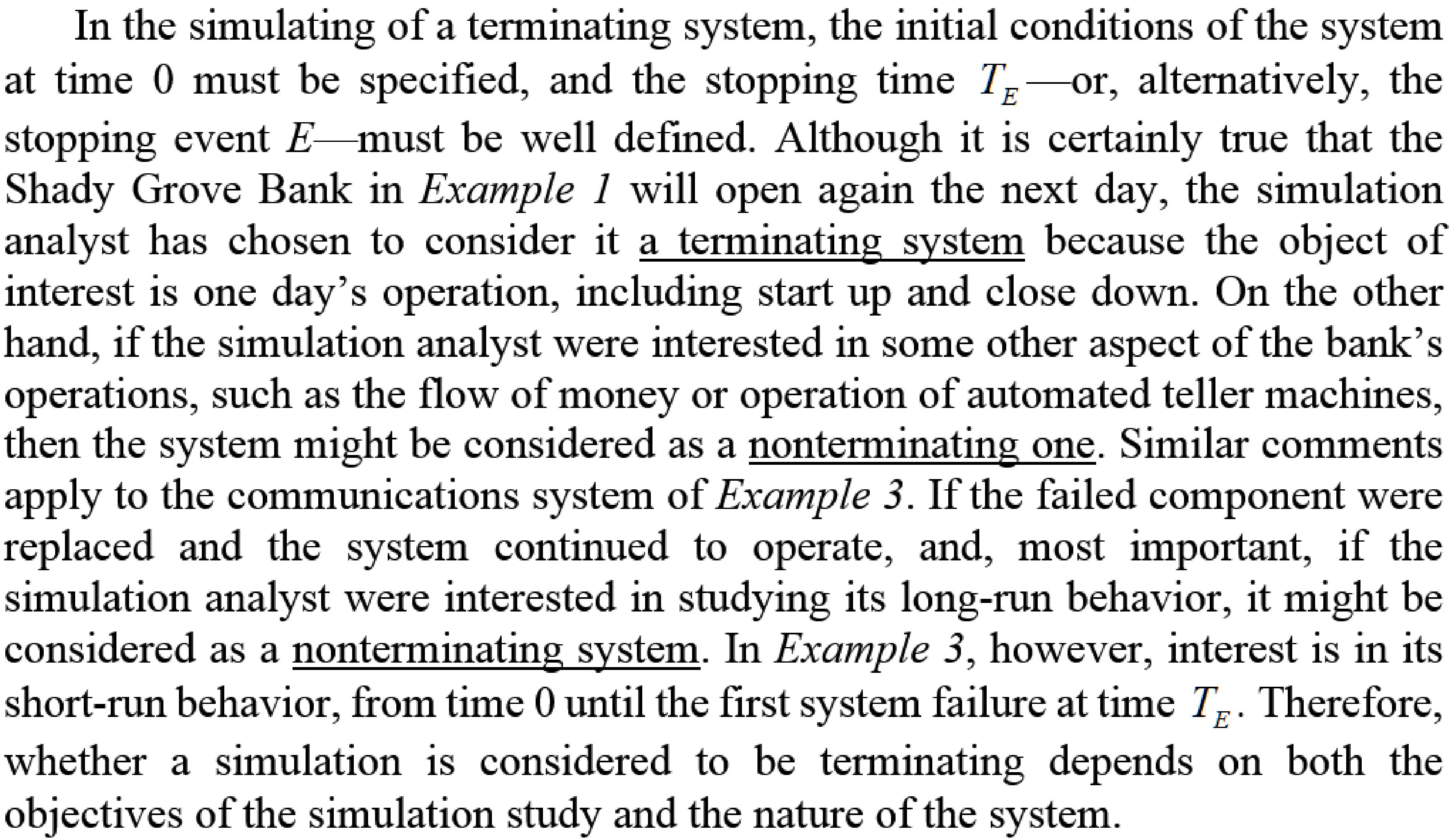
Example 3. A communications system consists of several components plus several backup components. It is represented schematically in Figure. Consider the system over a period of time T_E, until the system fails. The stopping event *E* is defined by *E*= {*A* fails, or *D* fails, or (*B* and C both fail)}. Initial conditions are that all components are new at time 0.



Example of a communications system.

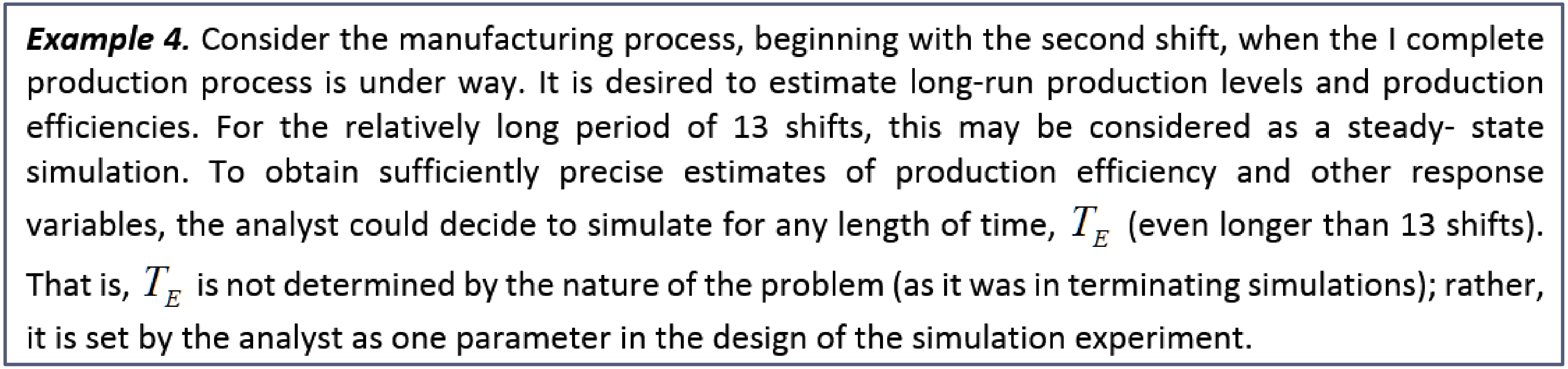
Notice that, in the bank model of Example, the stopping time T_E = 480 minutes is known, but in Example 3, the stopping time T_E is generally unpredictable in advance; in fact, T_E is probably the output variable of interest, as it represents the total time until the system breaks down. One goal of the simulation might be to estimate $E(T_E)$, the mean time to system failure.

When the simulation should be terminated?
Когда моделирование должно быть остановлено?

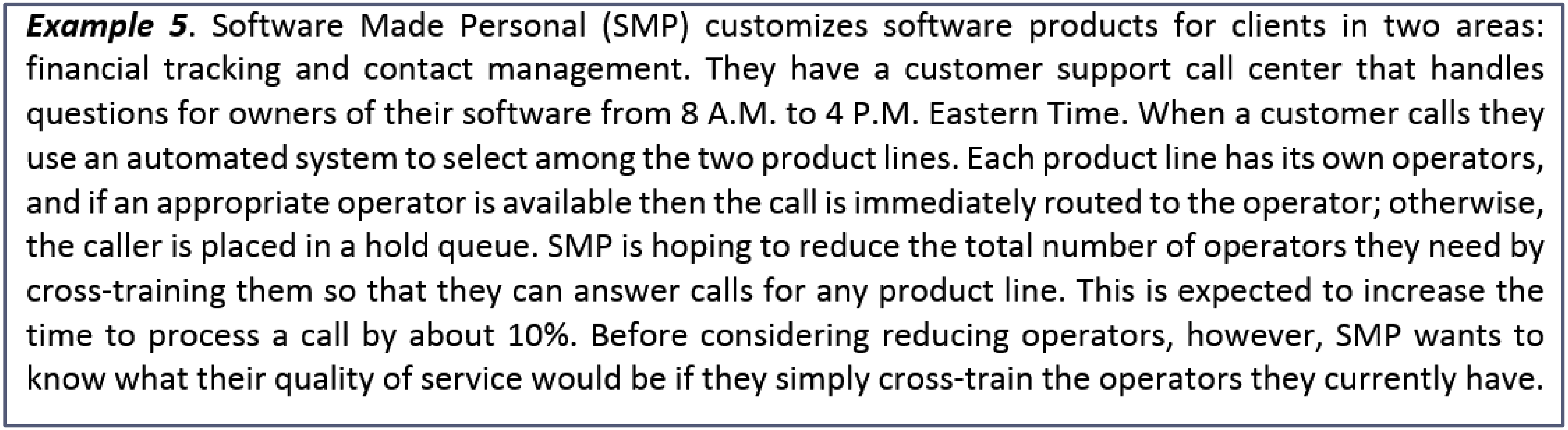


In the simulating of a terminating system, the initial conditions of the system at time 0 must be specified, and the stopping time T_E—or, alternatively, the stopping event E—must be well defined. Although it is certainly true that the Shady Grove Bank in *Example 1* will open again the next day, the simulation analyst has chosen to consider it a terminating system because the object of interest is one day's operation, including start up and close down. On the other hand, if the simulation analyst were interested in some other aspect of the bank's operations, such as the flow of money or operation of automated teller machines, then the system might be considered as a nonterminating one. Similar comments apply to the communications system of *Example 3*. If the failed component were replaced and the system continued to operate, and, most important, if the simulation analyst were interested in studying its long-run behavior, it might be considered as a nonterminating system. In *Example 3*, however, interest is in its short-run behavior, from time 0 until the first system failure at time T_E. Therefore, whether a simulation is considered to be terminating depends on both the objectives of the simulation study and the nature of the system.

Steady-state simulation models
Имитационные модели с устойчивым состоянием



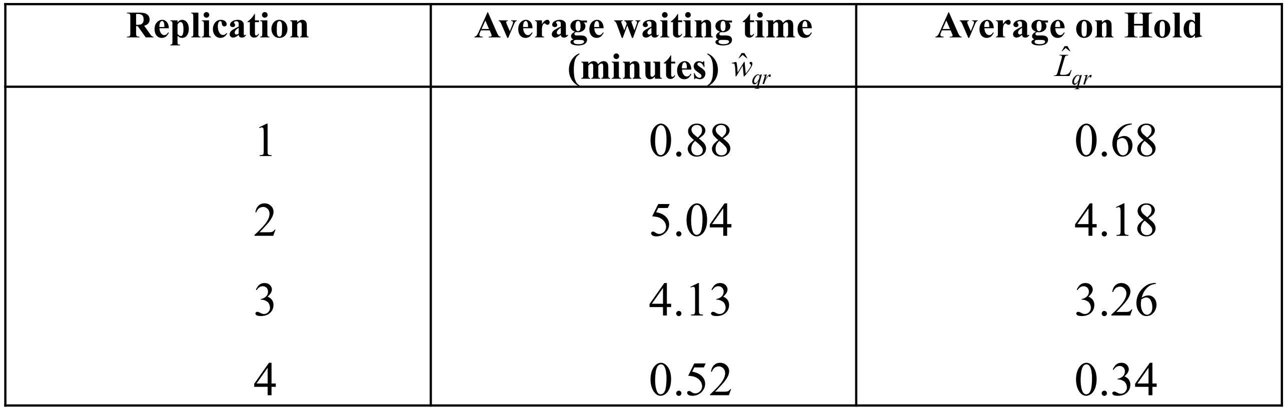
Example 4. Consider the manufacturing process, beginning with the second shift, when the I complete production process is under way. It is desired to estimate long-run production levels and production efficiencies. For the relatively long period of 13 shifts, this may be considered as a steady- state simulation. To obtain sufficiently precise estimates of production efficiency and other response variables, the analyst could decide to simulate for any length of time, T_E (even longer than 13 shifts). That is, T_E is not determined by the nature of the problem (as it was in terminating simulations); rather, it is set by the analyst as one parameter in the design of the simulation experiment.



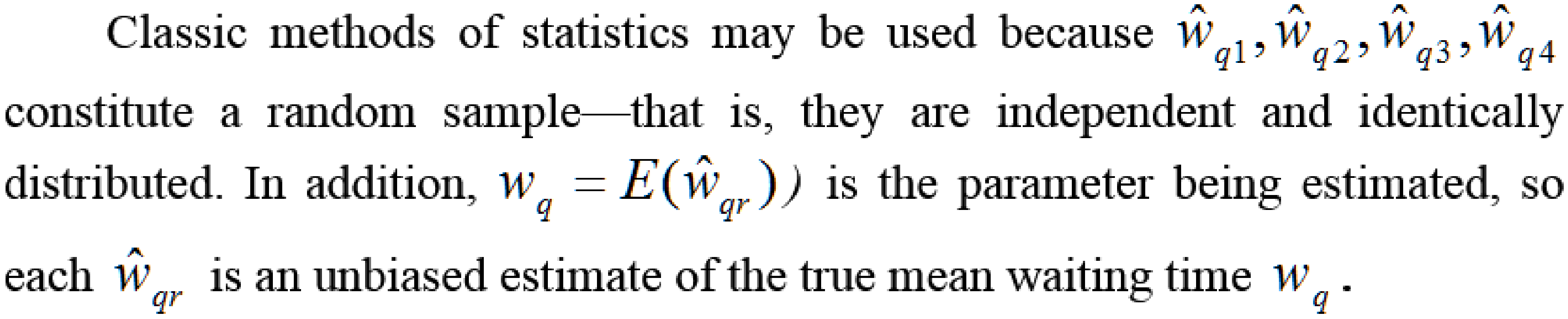
Example 5. Software Made Personal (SMP) customizes software products for clients in two areas: financial tracking and contact management. They have a customer support call center that handles questions for owners of their software from 8 A.M. to 4 P.M. Eastern Time. When a customer calls they use an automated system to select among the two product lines. Each product line has its own operators, and if an appropriate operator is available then the call is immediately routed to the operator; otherwise, the caller is placed in a hold queue. SMP is hoping to reduce the total number of operators they need by cross-training them so that they can answer calls for any product line. This is expected to increase the time to process a call by about 10%. Before considering reducing operators, however, SMP wants to know what their quality of service would be if they simply cross-train the operators they currently have.

Data analysis example

Пример анализа результатов



Replication	Average waiting time (minutes) $\hat{w}_{qr}$	Average on Hold $\hat{L}_{qr}$
1	0.88	0.68
2	5.04	4.18
3	4.13	3.26
4	0.52	0.34



Classic methods of statistics may be used because $\hat{w}_{q1}, \hat{w}_{q2}, \hat{w}_{q3}, \hat{w}_{q4}$ constitute a random sample—that is, they are independent and identically distributed. In addition, $w_q = E(\hat{w}_{qr}))$ is the parameter being estimated, so each $\hat{w}_{qr}$ is an unbiased estimate of the true mean waiting time w_q .

Confidence interval & prediction interval
Доверительный интервал и вероятностный интервал

To understand confidence intervals fully, it is important to understand the difference between a *measure of error* and a *measure of risk.* One way to make the difference clear is to contrast a *confidence interval* with a *prediction interval* (which is another useful output-analysis tool).

$$S^2 = \frac{1}{n-1}\sum_{i=1}^{n}(Y_i - \bar{Y})^2$$

Variance // Дисперсия

$$\bar{Y} \pm t_{\alpha/2,n-1}\frac{S}{\sqrt{n}}$$

Confidence interval // Доверительный интервал

$$\bar{Y} \pm t_{\alpha/2,n-1} S\sqrt{1+\frac{1}{n}}$$

Prediction interval // Вероятностный интервал

Estimation of prediction interval
Оценка вероятностного интервала

Example: 120 repetitions of industrial process are made, average time is 5.8 hours, std is 1.6 hours

$$t_{0.025,119} = 1.98$$

95% prediction interval (вероятностный интервал) is defined as:

$$5.8 \pm 1.98(1.60)\sqrt{1 + \frac{1}{120}} \Rightarrow 5.8 \pm 3.18$$

Estimation of confidence interval
Оценка доверительного интервала

$$\bar{Y} = \hat{w}_{qr} = \frac{0.88 + 5.04 + 4.13 + 0.52}{4} = 2.64$$ mean

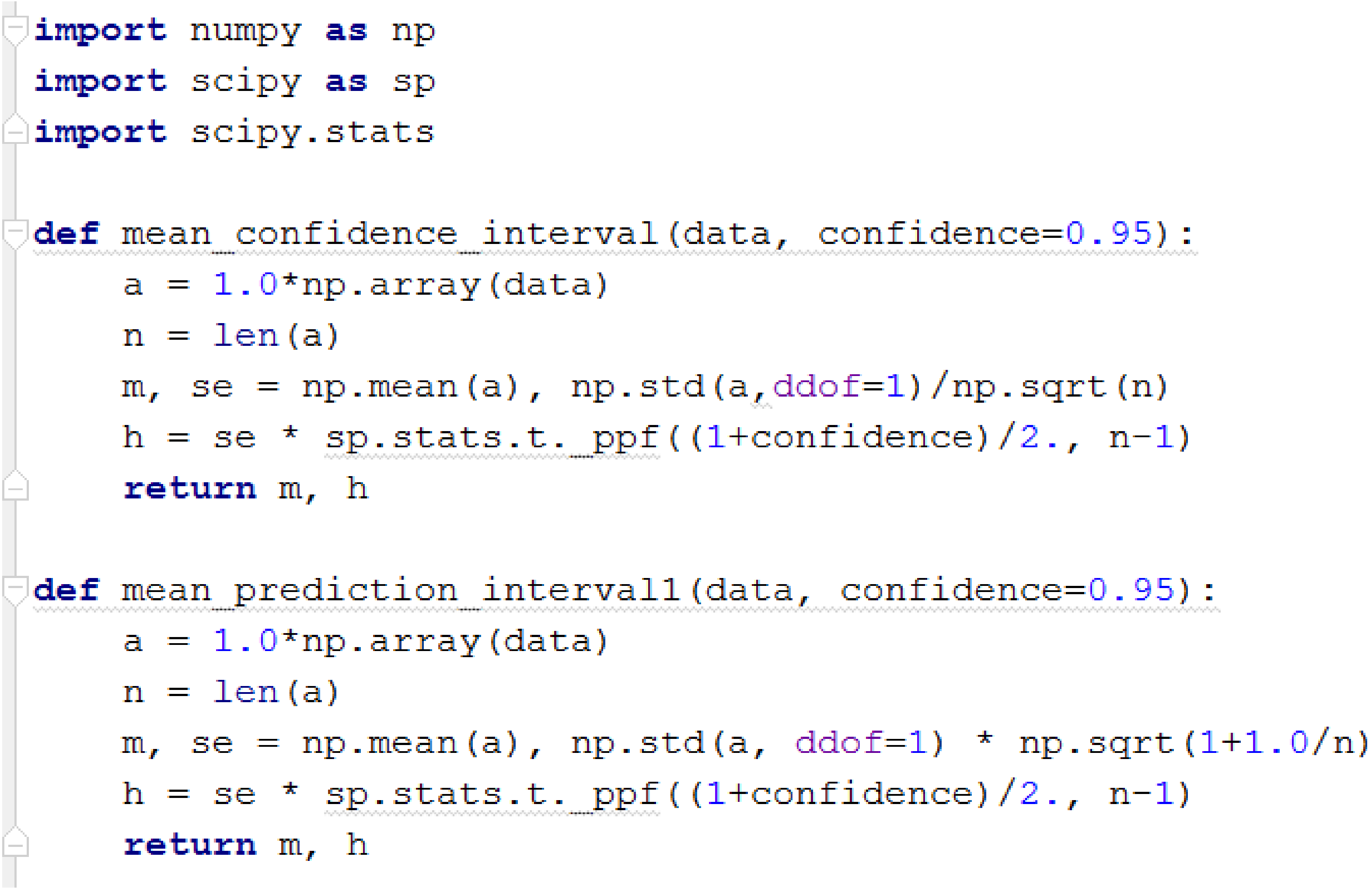
$$S^2 = \frac{(0.88 - 2.64)^2 + \ldots + (0.52 - 2.64)^2}{4 - 1} = (2.28)^2$$ variance

95% confidence interval for mean value is defined as:

$$H = t_{0.025,3}\frac{S}{\sqrt{4}} = (3.18)(1.14) = 3.62$$ ⇨ 2.64 ± 3.62 ! time<0😱

Conclusion: Four values aren’t enough for estimation of confidence interval

Confidence & predictive intervals estimation using Python
Оценка интервалов на языке Python



```
import numpy as np
import scipy as sp
import scipy.stats

def mean_confidence_interval(data, confidence=0.95):
    a = 1.0*np.array(data)
    n = len(a)
    m, se = np.mean(a), np.std(a,ddof=1)/np.sqrt(n)
    h = se * sp.stats.t._ppf((1+confidence)/2., n-1)
    return m, h

def mean_prediction_interval1(data, confidence=0.95):
    a = 1.0*np.array(data)
    n = len(a)
    m, se = np.mean(a), np.std(a, ddof=1) * np.sqrt(1+1.0/n)
    h = se * sp.stats.t._ppf((1+confidence)/2., n-1)
    return m, h
```

Confidence intervals with specific precision
Доверительный интервал заданной точности

Main condition is // Основное условие

$$H = t_{\alpha/2,R-1}\frac{S_0}{\sqrt{R}} \le \varepsilon \Rightarrow R \ge \left(\frac{t_{\alpha/2,R-1}S_0}{\varepsilon}\right)^2, R > R_0$$

Condition for *Ro*: $R \ge \left(\frac{z_{\alpha/2}S_0}{\varepsilon}\right)^2$

If the precision is not reached the process can be repeated:

$$\bar{Y} - t_{\alpha/2,R-1}\frac{S}{\sqrt{R}} \le \theta \le \bar{Y} + t_{\alpha/2,R-1}\frac{S}{\sqrt{R}}$$

Если точность не достигнута, процесс можно повторить