Автор: A.K.Aphoshar
28.04.2011 11:32 - Обновлено 28.12.2012 13:29

Статистические расчёты представляют собой несложную по содержанию, но крайне трудоёмкую математическую задачу. В большинстве своём эти расчёты связаны с первичной обработкой статистических данных, с построением точечных и интервальных оценок и реализацией алгоритмов проверок статистических гипотез. Существуют мощные программные среды символьных математических вычислений, которые содержат специальные статистические модули с большим набором функций для решения практически любых задач прикладной статистики. Среди этих математических пакетов наиболее известны Mathcad (http :// ptc . com / products / mathcad), Matlab (http :// mathworks . com / products / matlab) и Maple (http :// maplesoft . com / products / Maple).

Все перечисленные программные продукты являются коммерческими и легальное их использование требует материальных затрат, причём немалых, если ориентироваться прежде всего на студенческую аудиторию. Например, лицензия на использование последней на сегодняшний день версии Mathcad 15 обойдётся читателю в $ 1195 (продукт работает только на ОС семейства Windows), использование профессиональной версии Matlab r 2010 a

стоит примерно $ 900 (имеется версия по
GNU
/
Linux
, так что «линуксоидам» придётся здесь раскошелиться) и, наконец, почитатели канадского продукта
Maple
14 заплатят за профессиональную версию $ 1995 (этот пакет написан по все известные ОС:
Windows
,
GNU
/
Linux
и
Mac
). Далее, справедливости ради
,
следует заметить, что освоение вышеназванных продуктов на уровне программирования известных алгоритмов является нелёгкой и трудоёмкой задачей. По сути своей эта задача равносильна освоению базового синтаксиса какого-либо интерпретируемого языка программирования, например
PHP
,
JavaScript
,
Python
,
Ruby
и т.д. Кроме того, нельзя не упомянуть о «привязке» пользователя (прежде всего студента) к настольному компьютеру или в лучшем случае к ноутбуку, если он изберёт в качестве инструмента для решения статистических задач перечисленные выше пакеты. Очевидно, что для проведения практических занятий по математической статистике потребуется специально оборудованные компьютерные классы. В случае многочисленной студенческой аудитории реализация таких компьютерных лабораторий требует не только значительных материальных затрат, но и большого потребления электроэнергии. Наконец, уровень сложности учебных студенческих задач по статистике явно не соответствует мощности перечисленных выше математических программных продуктов. Применение этих средств для решения учебных задач равносильно зачастую стрельбе из пушки по воробьям. В связи с вышесказанным возникает вопрос о существовании мобильной, автономной и по возможности эффективной аппаратно-программной альтернативы, служащей для решения вычислительных математических задач, несущих прежде всего учебный характер.

Сразу же отбросим в сторону возможность применения всевозможных инженерных калькуляторов с различными встроенными статистическими функциями как инструментов безнадёжно устаревающих и не имеющих никакого будущего. Гораздо больший функционал имеют современные мобильные телефоны, смартфоны и коммуникаторы. Именно их мы и будем использовать для решения прикладных статистических задач. Напомним, что смартфоны и коммуникаторы отличаются от обычных мобильных телефонов наличием специальной мобильной операционной системы. В настоящее время наибольшее распространение получили мобильные операционные системы семейства Symbian, Windows Mobile, Android и т.д. Не вдаваясь в пространные и запутанные рассуждения о том, что такое операционная система и зачем она нужна, отметим здесь лишь тот факт, что её присутствие позволяет пользователю устанавливать на свой аппарат различное прикладное программное обеспечение, в частности, интерпретаторы некоторых высокоуровневых языков программирования. Среди этих языков необходимо выделить прежде всего
Python
и
JavaScript
. Кстати, интерпретатор последнего языка является неотъемлемой частью любого браузера, в том числе и мобильного. Отсюда мы делаем вывод о возможности использования
JavaScript
для разработки сценариев, связанных с решением прикладных статистических задач непосредственно на мобильном телефоне
. При этом для написания скрипта можно использовать самый простой текстовый редактор, а для отладки и запуска его необходим браузер, который имеется в наличии практически у любого современного мобильного телефона (
необязательно смартфона или коммуникатора
). Продвинутый читатель может быть несколько удивлён такому нетрадиционному использованию
JavaScript
, поскольку известно, что он применяется в основном в клиентском
web
-программировании. Тем не менее, встроенная в
JavaScript
математическая библиотека
Math
с набором элементарных функций и функцией
random
, развитая организация вычислительных потоков и управление ими, функциональное и объектно-ориентированное программирование позволяют разрабатывать скрипты для решения самых разнообразных вычислительных задач. При этом ещё раз особо подчеркнём, что
разработка, отладка и запуск скрипта происходят на самом мобильном телефоне (совсем необязательно, чтобы он был смартфоном или коммуникатором). Конечно

скорость выполнения трудоёмких вычислительных задач, решаемых с помощью JavaScript в сравнении с Python примерно на порядок ниже. Например, система линейных уравнений с матрицей 100×100 была решена на JavaScript примерно за 7 часов, Python справился с этой же задачей за 42 минуты (использовался средний смартфон с частотой процессора 200 МГц). Такое заметное снижение производительности связано прежде всего с тем, что JavaScript запускается в среде мобильного браузера, а его работа требует очень много оперативной памяти. Надо заметить, что приведённый выше пример несёт экстремальный характер. Чтобы узнать разницу в скорости выполнения реальных студенческих вычислительных задач (например, решение той же линейной системы с матрицей 5×5 или даже 7×7) потребуется встроить в исполняемый код специальный программный таймер, поскольку «на глаз» такие задачи решаются практически одновременно.

В заключении этой статьи поговорим об установке интерпретатора Python S60 для смартфонов с операционной системой Symbian и Python CE для коммуникаторов на базе ОС Windows Mobile. С интернет-ресурса http://sourceforge.net/projects/pys60/files/ загружаем два файла:

PythonForS
601.4.5_3
rdEd
.
sis
(размер этого файла 581,1
Kb
) и
PythonScriptShell
1.4.5_3
rdEd
.
sis
(~
21,4
Kb
). Оба файла устанавливаем
в том же порядке
на смартфон на базе ОС
Symbian
. Второй файл нужен для работы в консоли и для запуска скриптов. При этом следует отметить, что данный интерпретатор построен на ядре
Python
2.2.2. Можно установить другой интерпретатор на базе ядра
Python
2.5 (
https
://
garage
.
maemo
.
org
/
projects
/
pys
60/
), но при этом следует помнить, что память смартфона «похудеет» примерно на 5
Mb
. Этот ёмкий дистрибутив по своему функционалу практически нечем не отличается от предыдущего (
если оставаться в рамках наших учебных вычислительных задач
), разве что присутствием дополнительного модуля
cmath
, который содержит элементарные функции для работы с комплексными числами.

Известно, что в прикладной статистике эти функции используются крайне редко.

Для установки интерпретатора Python на коммуникаторы, работающие под управлением ОС Windows Mobile, соответствующий дистрибутив загружаем по адресу (http ://
sourceforge . net / projects / pythonce / files / (объём файла ~ 5 Mb). В отличии от Python S 60, который имеет очень мощную графическую поддержку (включая OpenGL), Python CE имеет только голую консоль.

Для обеспечения простейшей графики необходимо установить специальный графический модуль Tkinter. Установка этого модуля подробно описана на нашем сайте www . fm . cdml . ru в разделе «Диск». Там же можно найти многочисленные примеры скриптов по вычислительной математике, написанные на Python

S
60,
Python
CE
и
JavaScript
.

При работе в программных средах Python S60 и Python CE для решения учебных статистических задач можно выполнять инструкции языка
Python
непосредственно в консоли, но разумнее и рациональнее в самом начале написать скрипт, используя практически любой текстовый редактор (на нашем сайте
www
.
fm
.
cdml
.
ru
в разделе «Диск» читатель найдёт полнофункциональный редактор скриптов, написанный на
Python
S
60) и потом запускать его в оболочке
PythonScriptShell
или с помощью инструкции
execfile
('*'), где вместо * нужно записать полный путь к файлу скрипта в файловой системе смартфона или коммуникатора (более подробно об этом написано в
нашей книге
). В этой связи в дальнейших примерах мы будем указывать тексты (или их фрагменты) отдельных скриптов. Все разбираемые скрипты помещены на нашем сайте
www
.
fm
.
cdml
.
ru
в разделе «Диск».

Самым подходящим объектом для работы с выборочными статистическими данными является объект list (т.е. список или вектор с изменяемыми компонентами). Сама выборка может извлекаться из реальной генеральной совокупности или же образовываться искусственно с помощью встроенного генератора псевдослучайных чисел. В качестве обучающего примера рассмотрим численно-графическую обработку некоторой выборки из нормальной генеральной совокупности.

Сначала рассмотрим скрипт, написанный для Python S60 и прежде всего рассмотрим вычислительную часть скрипта

```
1 import appuifw, e32

2 from math import ceil, floor

3 from random import *

4 from key_codes import *

5 n=appuifw.query(u'n=','number')

6 fl=file(u'E:\\Data\\stat.txt','w')

7 lt=[gauss(0,5) for i in range(n)]

8 lt.sort()
```

```
9 fl.write(str(lt))

10 fl.close()

11 a, b=floor(lt[0]), ceil(lt[-1])

12 a0, h=a,(b-a)/10.

13 ls=[0 for i in range(10)]

14 j = 0

15 a+ = h

16 for i in range(n):

17 while True:

18 if lt[i]<=a:

19 ls[j]+=1:

20 break
```

```
21 else:

22 a+ = h

23 j+ = 1

24 f=lambda x, y: x+y

25 g=lambda x, y: x+y*y

26 M=reduce(f,lt)/float(n)

27 D=(reduce(g,lt)-lt[0]+lt[0]**2)/float(n)-M*M

28 print ‘ls=%ls\n’%ls

29 print ‘M=%f\n’%M

30 print ‘D=%f\n’%D
```

В этих 30 строчках кода построен вариационный ряд (список lt), ряд абсолютных частот (список ls), рассчитаны выборочные средние (переменная M

) и выборочная дисперсия (переменная
D
).

Рассмотрим более подробно реализацию этих действий. В строчках (1)-(4) импортируются необходимые модули или отдельные методы этих модулей. В строке (5) у пользователя запрашивается объём выборки. В строке (6) открывается файловый объект fl для записи (параметр ‘w’). Директория E:\\Data должна быть создана заранее на съёмном диске смартфона. С помощью функции
gauss
(
a
,
σ
) из модуля
random
организуется выборка объёма
n
из нормальной генеральной совокупности
N
(0,5) (строка (7)). В строке (8) с помощью метода списков
sort
строится вариационный ряд
lt
, который также записывается в файл
fl
, созданный ранее (строка (6)). Строки (11)-(23) посвящены построению ряда абсолютных частот (для примера осуществлена разбивка на 10 интервалов). В строчках (24)-(27) рассчитываются выборочное среднее
M
и выборочная дисперсия
D
. Наконец, в строчках (28)-(50) на дисплей смартфона выводятся значения
ls
,
M
и
D
.

У читателя может возникнуть вопрос о порядке значения объёма выборки n. Как показывает практика быстродействие среднего смартфона (~ 200 МГц) вполне хватает,

чтобы за
несколько секунд
обработать выборку объёмом
n
~ 10000!

Теперь займёмся графической частью нашего скрипта. Её основной целью является построение гистограммы для полученной выборки. Для решения этой задачи необходимо добавить следующий код.

```
appuifw.app.screen=’full’
c=appuifw.canvas()
appuifw.app.body=c
c.clear(0×00FF00)
class K:


def _ _ init _ _ (self, x, y):

self.x, self.y=x,y

self.c, self.tc=c,u’’

self.tl=u’’

def dron(self)

self.c.rectangle((self.x, self.y, self.x+24,310), 0x000000,
fill=0xff0000)

self.c.text((self.x+1, self.y+15), self.tc)

self.c.text((self.x+5, 315), self.tl)

m=float(max(ls))

def enter():

for i in range(10)

k=K(i*24, 310*(1-ls[i]/m)+5)
```

```
k.tc=unicode(ls[i])

k.tl=unicode(round(a0+i*h))

k.dron()

def quit():

app_lock.signal()

appuifw.app.exit_key_handle=quit

c.bind(EKeySelect, enter)

app_lock=e32. A0_lock

app_lock.wait
```

Теперь работа завершена. Этот скрипт нужно сохранить под произвольным именем с расширением .py в директории C:\Python или E:\Python в зависимости от того, куда установлен интерпретатор языка
Python
. Приведём некоторые комментарии к графической части скрипта. В строках (31)-(34) для отображения гистограммы, используется весь дисплей (параметр ’
full
’
), в качестве фона выбирается зелёный цвет (0X00FF00), основной объект для отображения графических примитивов – объект Canvas модуля appuifw. В строках (35)-(43) построен простой класс K. Экземпляры этого класса – прямоугольники красного цвета (0XFF0000) с чёрной границей (0X000000), координаты (
x
,
y
) левой верхней его вершины определяются конструктором класса, два атрибута tc и tl позволяют вывести внутри прямоугольника произвольные текстовые значения. В строках (44)-(50) вводится функция enter, которая запускается нажатием центральной кнопки джойстика (EKeySelect). Эта функция инициализирует построение гистограммы. Объект ‘замок’ e32.A0_
lock
необходим для удержания гистограммы на дисплее (
app
_
lock
.

wait
() строка 56). Выход из приложения происходит после нажатия правой софт-клавиши смартфона (см. строчки (51)-(53)).

На нашем сайте www.fm.cdml.ru в статье Диск\Разные задачи\Математическая статистика, читатель найдёт различные скрипты по обработке выборок и построению гистограмм, написанные на Python S60 и JavaScript. Помимо только что разобранного скрипта (кстати говоря, на сайте он имеет дополнительный функционал, связанный с выбором распределений и их параметров) следует обратить внимание на скрипты обрабатывающие распределение десятичных знаков чисел π и e. В главе о проверке статистических гипотез мы ещё вернёмся к этому вопросу в связи с гипотезой о равномерном распределении этих знаков.

Вместо объекта list, который мы использовали в предыдущем параграфе, в языке JavaScript используется аналогичный по свойствам и методам объект Array (массив). Первая часть скрипта по обработке выборок, написанная на Python S60 в контексте JavaScript выглядит следующим образом:

```
<script type=’text/javascript’>
var l=eval(prompt(’l=’,’[]’)), a, b, n;
var ls=[], h, M = 0, D = 0, i, j;
function num(x,y) {
return x-y;
}
l.sort(num);
document.write(’[’+l+’]’+’<br>’);
n=l.lendth
a=Math.floor(l[0]);
b=Math.ceil(l[n-1]);
h=(b-a)/10;
for (i=0; i<n; i++)
M+=l[i];
M/=n;
for (i=0; i<n; i++)
D+=l[i]* l[i]
D/=n; D-=M*M;
```

```
for (j=0; j<10; j++)
ls[j]=0;
j=0; a+h;
for (i=0; i<n; i++){
while (true){
if (l[i]<=a){
ls[j]+=1;
break;
}
else{
a+=h;
j+=1;
}
}
}
document.write('['+ls+']'+'<br>');
document.write('M'+M+'<br>');
document.write('D'+D+'<br>');
</script>
```

Сделаем необходимые комментарии к этому коду.

Строка (1) указывает браузеру о необходимости обработки текста между тегами <script> … </
script
> с помощью интерпретатора JavaScript. В строках (2), (3) объявляются переменные, в том числе два массива: l – основная выборка и ls – ряд абсолютных частот. Функция
n
um(
x
,
y
) нужна для сортировки элементов массивов по возрастанию (см. строки (4), (5)). Метод sort с аргументом num превращает случайную выборку l в вариационный ряд (строка (6)), который выводится на дисплей смартфона (строка(7)). Далее в строках (8)-(11) определяется спектр значений исследуемой случайной величины, т.е. интервал [a,b] и рассчитывается шаг дискретизации h (для примера мы рассматриваем 10 интервалов). В строках (12)-(32) рассчитывается выборочное среднее M и выборочная дисперсия D, а

также строится ряд абсолютных частот ls. В строках (33)-(35) полученные результаты выводятся на дисплей. Теперь поговорим о графическом представлении полученных результатов средствами JavaScript.

Для отображения графики необходимы дополнительные расширения в форме модулей и библиотек. В настоящее время имеется большое количество подобных расширений, отличающихся друг от друга своим функционалом. По существу все они апеллируют к расширенным возможностям языка HTML и к некоторым графическим примитивам. Для полноты изложения упомянем некоторые наиболее распространённые графические библиотеки, а также интернет-ресурсы, на которых читатель найдёт о них исчерпывающую информацию.

1. JS Charts (http://www.jscharts.com/)

2. Flot (http://code.google.com/p/flot/)

3. Style Chart (http://chart.inetsoft.com/)

4. Raphael (http://raphaeljs.com/)

5. ProtoChart (http://www.deensoft.com/lab/protochart/)

6. PlotKit (http://www.liquidx.net/plotkit/)

7. EJSChart (http :// www . ejchart . com /) (коммерческая графическая библиотека)

8. JavaScript Diagram Builder (http://www.lutanho.net/diagram/)

9. Canvas 3D Graph (http://dragan.yourtree.org/code/canvas-3d-graph/)

10. Dynamic Drive Line Graph Script (http://www.dynamicdrive.com/dynamicindex11/linegraph.html/)

11. Bluff (http://bluff.jcoglan.com/)

12. JSXGraph (http :// jsxgraph . uni-bayreuth . de / wp /) (бесплатная библиотека с расширенными математическими возможностями).

Перечисленные выше библиотеки обладают неплохой «кросс браузерностью» – последние версии Internet Explorer, Opera, Firefox при соответствующих настройках поддерживают их функционал безукоризненно. Однако следует заметить, что здесь речь идёт прежде всего о «настольных» версиях этих браузеров. Мобильные версии перечисленных браузеров, которые нас больше всего интересуют в этой главе, поддерживают функции этих модулей лишь отчасти, либо же вообще отказываются работать.

Среди библиотек, которые полностью поддерживаются мобильными браузерами следует упомянуть библиотеку M.Bostock (ссылка на этот ресурс довольно громоздкая и мы отсылаем читателя на наш сайт www.fm.cdml.ru в раздел Диск/Математическая статистика/, где можно найти скрипты, использующие этот модуль, а также ссылку на первоисточник).

Теперь вернёмся к нашему скрипту и дополним его некоторым кодом, необходимым для построения гистограмм

```
<script type=’text/javascript’ src=’graph.js’></script>
<script type=’text/javascript’>
var g=new Graph(200,300);
```

```
g.scale=5;
g.setXScale();
g.inc=1;
g.rows=eval('['+'['+ls+']'+']');
g.built();
</script>
```

Построение гистограммы завершено. Поясним, что сделано в этом коде. В строке (37) подключается основной графический модуль graph.js. Размер его ~ 16 Kb и это самый маленький модуль из перечисленных ранее. В строке (39) создаётся графический объект с шириной 200 px и высотой 300 px. В строках (40)-(43) устанавливаются шкалы по оси Y (интервал равен 5) и по оси X (точка отсчёта O, интервал равен 1).

В (43) строке графическому объекту передаётся выборка ls абсолютных частот, полученная ранее. И наконец, метод build объекта Graph осуществляет построение гистограммы в соответствии с введёнными данными (строка (44)).

В заключении отметим, что объект Graph имеет целый ряд других полезных атрибутов и методов, для ознакомления с которыми мы отсылаем читателя к первоисточнику (выйти на него можно через любую поисковую систему по ключевому слову JavaScript GraphBuilder). В частности, на одном графике можно расположить сразу несколько гистограмм, что мы и осуществили в нашей программе по исследованию распределения цифр в десятичных разложениях чисел e и π (http :// www . fm . cdml . ru).