单个网络日志数据的导入

项目制作者：唐琪

目录；

1. 考核项目概要

（考核时间，内容要求，项目简介）

1. 考核项目报告内容

第一阶段:研读题目和原始数据的观测

第二阶段：运用SAS软件编写代码

第三阶段：对结果进行解释分析

1. 个人的不足之处
2. 考核项目中的体会
3. 对SAS的认识

一，考核项目概要

考核时间：2014年3月至2014年5月

考核项目简介

项目名称：单个网络日志数据的导入

项目要求：自主调研分析确定运用SAS软件完成考核项目，并简述项目功能，编写出相关的代码。数据说明。具体要求如下

1、按要求提取关键数据。

2、对所提取的数据进行简短说明。

项目时间：两个月。

对单个网络日志数据的导入。通过运用SAS软件编写代码来提取所需的关键数据并对所提取的数据进行说明，达到所需的要求结果。此项目数据的导入，能有效的剔除无用数据，保存有用关键数据。

此项目的意义：在于能将散乱的网络日志数据按类型分类，并导入SAS中。也让我了解到了很多关于在SAS中导入日志数据的知识。

二，考核项目报告

考核项目分为三阶段；

**第一阶段:研读题目和原始数据的观测**

**第二阶段：运用SAS软件编写代码**

**第三阶段：对结果进行解释分析**

**第一阶段:研读题目和原始数据的观测**

本次考核项目题目为：**单个网络日志的导入**

**原始数据的观测。**

对于一个网络日志导入的前提，是首先要对所导入数据的信息,作用有所了解，才能提取相对应所需要的有用信息数据，剔除不相关无用的数据。通过对原始数据的观测。能够大致了解项目的要求。从而对之后的程序编写过程奠定基础，相应减少编写的时间。

**数据说明**

本样本数据包提供了多个随机抽取的样本用户，以及相应的样本人口属性信息。

样本数据包中的数据文件可分为2部分，其中behavior文件夹中是按日期归档的样本行为日志，demographic.csv是样本的人口属性信息，二者可通过样本ID关联。

在样本行为日志中，以“字段<=>值”的格式记录数据，若一行中有多个字段、值对，则以“[=]”分隔。

例如，一个典型的样本行为日志如下：

*Last<=>26037*

*L\_Start<=>2012-08-06 08-49-16*

*…*

*T<=>8933[=]P<=>360chrome.exe[=]I<=>1224[=]U<=>item.taobao.com/item.htm[=]A<=>205e0[=]B<=>203b2[=]V<=>5.3.0.922*

*T<=>8935[=]P<=>explorer.exe[=]I<=>4052[=]W<=>10302[=]V<=>6.1.7600.16385*

*T<=>8952[=]P<=>Foxmail.exe[=]I<=>4608[=]W<=>404a0[=]V<=>7.0.1.91*

*T<=>8960[=]P<=>explorer.exe[=]I<=>4052[=]W<=>60c24[=]V<=>6.1.7600.16385*

*T<=>8986[=]P<=>iSeeQView.exe[=]I<=>8436[=]W<=>20d36[=]V<=>1.2[=]N<=>看图精灵[=]C<=>北京远图科技*

*T<=>8988[=]P<=>explorer.exe[=]I<=>4052[=]W<=>60c24[=]V<=>6.1.7600.16385*

*…*

样本用户每次开机时，都会形成一个对应的日志文件，文件的开头2行会记录以下2个值：

Last表示日志的最后一次更新时间点距离开机时间的秒数；

L\_Start表示样本用户的开机时间（年-月-日 时-分-秒）。

接下来，数据采集程序会以2秒一次的频率扫描样本用户计算机的当前焦点窗口，若焦点窗口发生变化，则会在日志中追加一条记录。其中：

T表示该记录的时间点，以距离开机时间的秒数形式记录，例如8933表示距离开机时间的8933秒后；

P表示焦点窗口的窗口进程名；

I表示焦点窗口的窗口进程名；

U表示浏览器窗口的地址栏内容，即当前浏览网页的URL（为了保护样本隐私，已将URL中第1个问号之后的参数部分截断）；

A表示浏览器窗口的地址栏句柄；

B表示浏览器窗口的当前标签页句柄；

V表示焦点窗口对应的程序版本号；

W表示非浏览器焦点窗口的窗口句柄；

N表示焦点窗口对应的程序名称（仅在第一次出现时记录）；

C表示焦点窗口对应的程序所属公司名（仅在第一次出现时记录）。

通过对原始数据的观测，大致可得出数据的总体情况。

1，单个日志数据必定会存在

(Last日志的最后一次更新时间点距离开机时间的秒数)

（L Start样本用户的开机时间）

（T记录的时间点，以距离开机时间的秒数形式记录）

（p焦点窗口的窗口进程名）

(I焦点窗口的窗口进程名)

(v焦点窗口对应的程序版本号)。

2，数据大致字符长度的确定。

**→→→→→**

方法

主要的问题

**→**

正则表达式记录文本的规律

数据的特点，有无规律

用length对所导入数据进行长度定义

导入数据的长度

**→**

字符串函数进行截取

主要数据的截取

**→**

**→**

详见下面具体说明

编写代码三阶段

**→**

能直观的观测日志数据

日志导入作用

**第二阶段：运用SAS软件编写代码**

数据代码，必要的代码作用说明

详细的代码设计

在SAS的基础上，需要进行编码的初步设计。在设计中，描述实现具体数据导入涉及到的主要函数，及调用关系等，需要说明代码在SAS中每个每个函数或表达式，以便进行测试。且应当保证啥季节足够详细，能够根据详细涉及进行编码。

数据代码：

%let a=d:\work\。。。。;

%let b=.txt;

%let filename="&a&b";

libname c 'd:\work'; /\*定义逻辑库\*/

**data** c.qq; /\*数据集存储到指定\*/

infile &filename lrecl=**32767** end=fend length=length encoding=utf8;

/\* LRECL是用来规定逻辑记录长度，后面的数字是用来定义读取的文件逻辑长度为多少字节的encoding了解安装的SAS系统是否支持读取UTF-8编码，否则很多数据SAS读取以后会出现乱码\*/

if \_n\_=**1** then do; /\*if的转移语句\*/

Ltp=prxparse("/Last<=>.{1,}/"); /\*定义正则表达式\*/

Stp=prxparse("/L\_Start<=>.{1,}/");

Tstr=prxparse("/T<=>\d{1,}/");

Pstr=prxparse("/P<=>[^\[]\*/");

Istr=prxparse("/I<=>[^\[]\*/");

Ustr=prxparse("/U<=>[^\[]\*/");

Astr=prxparse("/A<=>\d{1,}/");

Bstr=prxparse("/B<=>\d{1,}/");

Vstr=prxparse("/V<=>[\d{1,} \*,.]\*/");

Wstr=prxparse("/W<=>[^\[]\*/");

Nstr=prxparse("/N<=>[^\[]\*/");

Cstr=prxparse("/C<=>[^\[]\*/");

end;

retain Ltp Stp Tstr Pstr Istr Ustr Astr Bstr Vstr Wstr Nstr Cstr; /\*对变量进行值的初始化和保留到下一个迭代步\*/

length Ltime Time proID Inpro addr Baddr $**20**; /\*定义变量的长度\*/

length Stime version Waddr $**30**;

length name company $**40**;

length Url $**100**;

length str $**5000**;

input str $varying. length; /\*输入可变参数，默认长度\*/

if \_n\_=**1** then do;

call prxsubstr(Ltp,str,start,length); /\*对SAS子程序进行调用\*/

if start gt **0** then do;

Ltime=substr(str,index(str,'>')+**1**,**5**); /\*运用SUBSTR和index函数截取数据\*/

end;

else do; Ltime="";end;

retain Ltime;

delete; /\*DELETE语句告诉SAS系统停止处理当前的观测值，SAS系统返回DATA步的开始处进行下一次执行\*/

end;

if \_n\_=**2** then do;

call prxsubstr(Stp,str,start,length); /\*返回匹配模式的起始位置和长度\*/

if start gt **0** then

Stime=substr(str,index(str,'>')+**1**,**20**);

else Stime="";

retain Stime;

delete;

end;

call prxsubstr(Tstr,str,start,length);

if start gt **0** then

Time=scan(str,**2**,'>['); /\*scan函数查找特殊字符并分隔字符串\*/

else Time="";

call prxsubstr(Pstr,str,start,length);

if start gt **0** then

proID=scan(str,**4**,'>[');

else proID="";

call prxsubstr(Istr,str,start,length);

if start gt **0** then

Inpro=scan(str,**6**,'>[');

else Inpro="";

call prxsubstr(Ustr,str,start,length);

if start gt **0** then

Url=scan(str,**8**,'>[');

else Url="";

call prxsubstr(Astr,str,start,length);

if start gt **0** then

addr=scan(str,**10**,'>[');

else addr="";

call prxsubstr(Bstr,str,start,length);

if start gt **0** then

Baddr=scan(str,**12**,'>[');

else Baddr="";

call prxsubstr(Vstr,str,start,length);

if start gt **0** then

Version=substr(str,start+**4**,length-**4**); /\*....\*/

/\*scan(substr(str,index(str,'>')+1,30),1,'>[')\*/

else Version="";

call prxsubstr(Wstr,str,start,length);

if start gt **0** then

Waddr=scan(str,**8**,'>[');

else Waddr="";

call prxsubstr(Nstr,str,start,length);

if start gt **0** then

name=scan(str,-**3**,'>[');

else name="";

call prxsubstr(Cstr,str,start,length);

if start gt **0** then do;

start=start+**4**;

company=scan(str,-**1**,'>');

end;

else do; company="";end;

keep Ltime Stime Time proID Inpro Url addr Baddr version Waddr name company;

**run**;

**数据源代码结构及主要运用的函数语句的说明**。

**在以上数据源代码中的数据成员，函数成员和表达式成员**。

数据成员：

Ltime (日志的最后一次更新时间点距离开机时间的秒数 )

Stime (样本用户的开机时间)

Time （以距离开机时间的秒数 ）

Version （焦点窗口对应的程序版本号）

proID （焦点窗口的窗口进程名 ）

Waddr （非浏览器焦点窗口的窗口句柄）

Inpro （焦点窗口的窗口进程名 ）

Name （焦点窗口对应的程序名称）

addr （浏览器窗口的地址栏句柄）

company (焦点窗口对应的程序所属公司名)

Baddr （浏览器窗口的当前标签页句柄 ）

Url （浏览器窗口的地址栏内容 ）

函数成员：

字符串函数：

Substr(str,n,m); 实现字符串的截取功能

Scan(str, , ‘’); 寻找一个字符串中第一个出现的某字符的位置

Substr(str,index(str, ), ); substr函数通过index字符串函数取字符串的开始位置

表达式成员：

prxparse(“”); 正则表达式用于字符串处理

call prxsubstr(str, ,start,length); 返回匹配模式的起始位置和长度

**在以上数据源代码中调用语句。**

If ……then do ……end;

If ……else 判断语句

**其他语句。**

Retain； 对变量进行值的初始化和保留到下一个迭代步

Length； 定义变量的长度

Delete； 告诉SAS系统停止处理当前的观测值，SAS系统返回DATA步的开始处进行下一次执行

$varying； 输入可变参数，默认长度

Keep; 输出缓冲器中保存相关的变量

LRECL 是用来规定逻辑记录长度，后面的数字是用来定义读取的文件逻辑长度为多少字节的

第三阶段：对结果进行解释分析

对过程运用的函数，语句的分析

1. length对变量进行长度的定义

length Ltime Time proID Inpro addr Baddr $**20**; length Stime version Waddr $**30**;

length name company $**40**;

length Url $**100**;

length str $**5000**;

由于日志数据中，每个数据的长度都不一样。这就要在数据的初步观测阶段。对数据进行一个简单的长度估计。

2.正则表达式prxparse(“”);用于字符串的处理

经常会有查找符合某些复杂规则的字符串的需要。正则表达式就是用于描述这些规则的工具。换句话说，正则表达式就是记录文本规则的代码。

对于下列正则表达式

Tstr=prxparse("/T<=>\d{1,}/");

Astr=prxparse("/A<=>\d{1,}/");

Bstr=prxparse("/B<=>\d{1,}/");

Vstr=prxparse("/V<=>[\d{1,} \*,.]\*/");

分别是以“T<=>”“A<=>”“B<=>”“V<=>”开头

“\d” 为匹配一位数字(0，或1，或2，或……)

。“\d{1，}”保证字符串里包含1到 连续位数字。

由于 “V<=>”数据包含“.”所以需要用到“\\*”

对于下列正则表达式

Pstr=prxparse("/P<=>[^\[]\*/");

Istr=prxparse("/I<=>[^\[]\*/");

Ustr=prxparse("/U<=>[^\[]\*/");

Wstr=prxparse("/W<=>[^\[]\*/");

Nstr=prxparse("/N<=>[^\[]\*/");

Cstr=prxparse("/C<=>[^\[]\*/");

分别是以“P<=>”“I<=>”“U<=>”“W<=>”“N<=>”“C<=>”开头

“^” 匹配要用来查找的字符串的开头[]。数据包含“.”

正则表达式函数

例如 ：call prxsubstr(Stp,str,start,length)

返回匹配字符串在目标字符串中的开始位置长度

3. 运用字符串函数截取数据。

对于Last 和 L\_Start在单个日志数据中只出现一次。

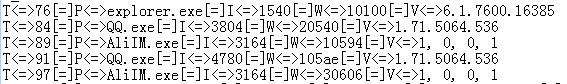
即可用字符串函数substr(str,index(str,’>’)+1, )进行数据的截取

例如：中

可先用index(str,’>’)寻找到‘>’字符，然后用substr(str, ,5)截取数据11525

对于其他日志数据出现多次，且数据不同的情况。可直接用简单的

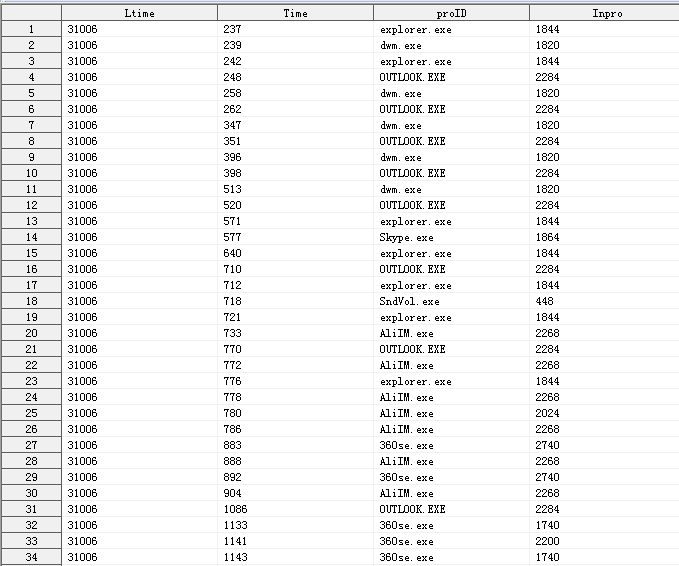
scan(string,i,"char") .。“string”为原数据，“i”为第i个字符串，“char”为分隔符。

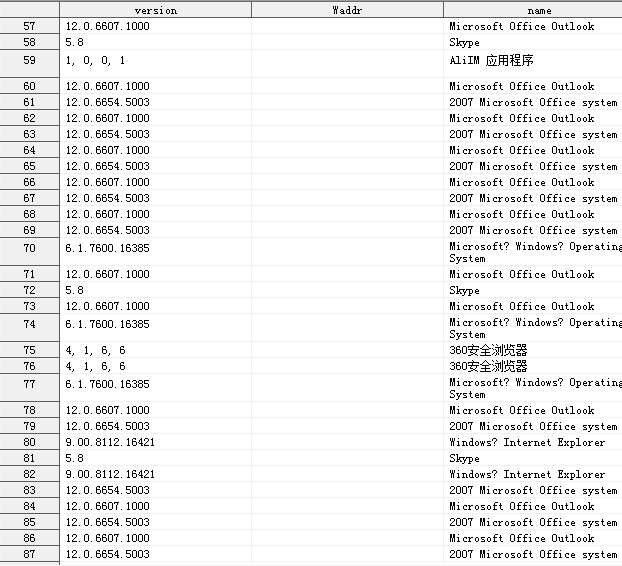
例如

Time=scan(str,**2**,'>[')即可对“T<=>”进行数据截取

上诉为代码的简单说明，通过在SAS中的运行，即可得出日志数据的导入结果。

部分结果图；







1. 存在的不足。

没能时刻都静下心来学习SAS这门语言，在前期的学习中也走了不少的弯路。加之性格内向的原因，向别人询问知识的次数不是很多。但自认为学习还是比较积极的，但刚开始没有经验不知道有什么地方做得不足，大家能够提意见的意见，我也会很乐意接受。

通过这次考核，发现了一些自己不足的地方需要改进。比如(在写报告方面就欠缺一些能力)。希望以后能在以后的学习中得到更多的锻炼和改进。

1. 学习SAS的体会

SAS最大的难点在于它那一套自成一体的编程语言。在SAS底下，所有的问题都要自己编程解决。而且，它的语法不是大家熟悉的类似C语言的语法，显得繁琐怪异。对于刚开始学习SAS，很大的精力就花在学习这些机械的语法上。

刚开始学习SAS就觉得很难，摸不清方向。相对于曾经学过的C语言来说，是一种新的编写语言。也没及时请教学长，学姐。写程序就缺乏经验了。

在考核项目中，在边学习编完成项目过程中，我对SAS重新认识的起点。要真正的学好SAS，是要结合自己学习内容和性质出发。从我个人出发，我不希望成为专业的SAS程序员，因为SAS只是我完成工作的工具，我的工作能力不是通过她来体现的。而如果决定了认真学习SAS，那么在自己学习的过程中，就会完全使用SAS解决自己的问题。不东学一个西学一个。

1. 对SAS的认识

SAS是统计软件。可以把SAS简单看成两大块。一块是SAS/STAT，另一块是SAS/BASE。当然最强大的是SAS/STAT，几乎所有的统计算法。 SAS/BASE能够帮你对数据进行处理，整合。

统计是用数学解决现实问题的理论方法，而SAS/STAT就是解决问题的具体工具了。SAS提供了从基本统计数的计算到各种试验设计的方差分析，相关回归分析以及多变数分析的多种统计分析过程，几乎囊括了所有最新分析方法，其分析技术先进，可靠。分析方法的实现通过过程调用完成。许多过程同时提供了多种算法和选项。

现在是大数据时代。大数据指集合形式的所有数据，其大小或复杂性使得无法通过常用技术以合理的成本并在可接受的时限内对其进行捕获、管理和处理。它是以下三种技术趋势汇聚的结果：

大事务数据：事务数据量的大规模增长

大集成数据：交互数据（例如[社交媒体](http://www.so.com/s?q=%E7%A4%BE%E4%BA%A4%E5%AA%92%E4%BD%93&ie=utf-8&src=wenda_link)、[传感器技术](http://www.so.com/s?q=%E4%BC%A0%E6%84%9F%E5%99%A8%E6%8A%80%E6%9C%AF&ie=utf-8&src=wenda_link)、详细呼叫记录和其它来源）的激增

大数据处理：借助 Hadoop 进行新的高度可扩展的处理。

学习好SAS能为以后的未来打下良好的基础。