

Track SAS Viya user logins to identify usage and drive improvement using actionable Intelligence

Anurag Kulkarni, RevSpring, Columbus, OH

ABSTRACT

Do you want to track how your users are taking advantage of tools within SAS Viya? Do you offer reports to your clients and want to analyze how the reports are being used to drive improvement? Are you a SAS superuser, want to track usage, schedule jobs intelligently and manage your environment proactively? The research will walk you through how to archive the data, derive KPIs and build interactive visuals to track this information.

INTRODUCTION

This paper references tools in SAS Viya 3.4 (Linux environment), hence a basic understanding of the following tools is helpful but not required: Report Viewer, Report Designer/Visual Analytics, Environment Manager, SAS Job Execution and SAS Studio. The paper applies to all versions of SAS Viya that support audit logs. First, we will walk through how to archive audit logs data in SAS Viya (since SAS Viya retains only 7 days of audit logs) and latter two sections discuss how to use Report Designer to build reports with actionable intelligence around the audit logs data.

ARCHIVING AUDIT LOGS

There are two ways to archive audit logs (or any other system generated tables) in SAS Viya environment: on the Viya server and on the Viya web interface. We will only discuss the latter in this research. The linux server is not always accessible to every user and hence the second approach may be more beneficial depending on the end goal. The second approach also gives more control over the archiving process including what data needs to be archived, the time frame and the aggregation of the archived data. For trending purposes it's more economical to aggregate the data, whereas analyzing detailed logs may require columns to be slimmed down. Either way, it's great to have that flexibility with this approach.

SAS STUDIO JOB

The job to archive audit logs (or any other datasets) can be written in SAS Studio. We can then use SAS Job execution to schedule this job to run. Now, let's walk through the SAS code step by step.

First step is to declare a CAS session and use CASLIB _ALL_ ASSIGN to include all CAS libraries in SAS Studio (You should see this on the left):

```
cas mycas;  
caslib _all_ assign;
```

Next, we use a DATA _NULL_ step to check if the base table exists. When you first run the program or after the server reboots, there won't be any base table in CAS. A good practice is to save a backup table in a different CAS library. The code below references the backup table audit_history in Public CASLIB if the data isn't available in SAS_Data CASLIB:

```
data _null_;  
  if (exist("SAS_Data.audit_history"))  
  then call symput('add_file',"SAS_Data.audit_history");  
  else call symput('add_file',"Public.audit_history");  
run;
```

SAS Viya automatically generates and updates Audit CAS table every 3 hours (may differ based on the configuration). We can specify the type of log to archive. In the following code, we keep logs from reports application:

```
data Audit;
  set SAS_Data.Audit;
  if application= 'reports' and user ne '';
run;
```

Next, we create/import the history table in SAS Studio:

```
data Audit_History;
  set &add_file;
  /* if application= 'reports' and user ne ''; */
run;
```

Then we append Audit table into the Audit_history table:

```
proc datasets;
  append base=audit_history
  data=audit force;
run;
```

Depending on how often we run the archive job, there can be overlap and it's likely to have duplicate rows. The following sort steps removes duplicates:

```
proc sort data=audit_history; by id; run;
data Audit_history;
  set audit_history;
  by id;
  if last.Id = 1;
run;
```

The above datasets were generated in SAS Studio work. Last, we use PROC CASUTIL to load/reload the data in CAS so that the Report Designer can access the data:

```
proc casutil;
  droptable casdata="Audit_History" incaslib="SAS_Data";
  load data=work.Audit_history outcaslib=SAS_Data;
  promote casdata="Audit_history" incaslib="SAS_Data" outcaslib=SAS_Data;
  save casdata="Audit_history" incaslib="SAS_Data" outcaslib=Public
  casout="Audit_history" replace ;
run;
quit;
cas mycas terminate;
```

Note: before setting up this job, the audit table needs to be moved to a Caslib that can be accessed in SAS Studio. This can be changed within Configuration in Environment Manager.

SCHEDULING THE JOB

After we have written and saved the SAS job, we can use SAS Job Execution to schedule the Job to run daily. We would need to save the following code in a job definition before exporting the job to SAS Environment Manager.

Program	Status	Run Time	Submitted
PersistAuditLogsAutomation	✓	00:00:11.9	Apr 5, 2019 07:30 AM

Figure 1. SAS Job Execution

```
cas mySession host="your-hostname" port=5570 sessopts=(caslib=casuser
timeout= 5400 locale="en_US" REMOVENODECANCELTIMEOUT=5400);
```

```
filename myJob FILESRVC folderpath='/Projects/SAS Development'
filename='PersistAuditLogs.sas';
```

```
%include myJob / source;
```

BUILDING REPORTS USING AUDIT LOGS

In this section we will cover deriving KPI's and building interactive reports arounds these KPI's. SAS Report Designer with SAS Viya 8.x provides some great reporting and data preparation options.

DERIVING THE KPI'S

Most of the calculations needed are simple to derive in Report Designer. However, some require more compounded functions and aggregations. Let's walk through how each of these KPI's can be derived in Report Designer.

The number of **Days Since Last Login** can be used to track recent usage. It can be derived as follows:

```
Floor(( ( TreatAs(_Number_, Now()) - TreatAs(_Number_,
'Time Stamp Date'n) ) / 86400 ))
```

The audit log table is structured in a way that allows us to easily look at **Activity**. Activity is just the frequency in this table. This may change if the data is aggregated any differently.

The total number of **Days Logged In** gives us an insight into long term trend like **Activity** does. Sometimes users can be more frequent than others and login multiple times a day, so by strictly looking at distinct logins per day (i.e. more than one login would count as just one login for one day), we eliminate the tendency to a skew. The code for this aggregated measure as is follows:

```
Distinct [_ByGroup_] ('Time Stamp Date'n)
```

Average Monthly Logins/Activity can again provide insight into long term trends. The aggregation can be derived as follows:

```
Distinct [_ByGroup_] ('Time Stamp Date'n) / Distinct [_ByGroup_]
('Time Stamp Month'n)
```

Keeping an eye on **Number of Users** can give us an idea about how many users we are supporting compared to the capacity of the server.

Usage by weekday/time

BUILDING INTERACTIVE VISUALS

We will now cover how to plot the above metrics into interactive visuals that gives us quick insight.

Days Since Last Login can be viewed on a bar chart grouped by department/client. This can be used to identify more recent usage or groups with no usage in the last x number of days.

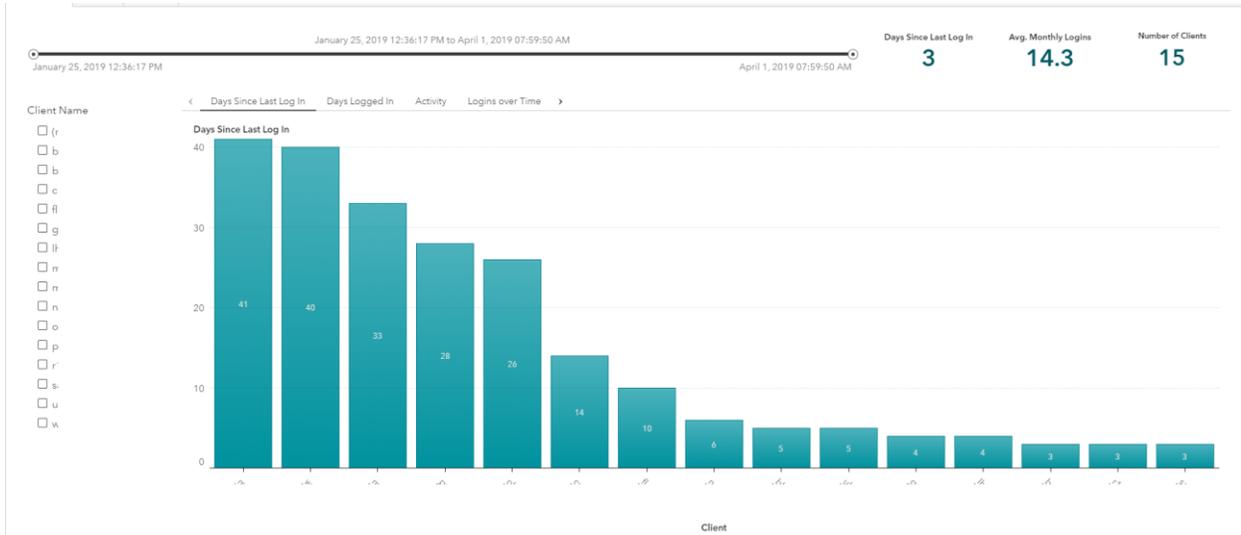


Figure 2. Days Since Last Login

Activity and **Days Logged-In** can be plotted over month to see long term usage and trending. This information can be combined with the number of users to see how the usage increases compared to the number of users. As we accumulate more data, this can also tell us about any seasonal trends.

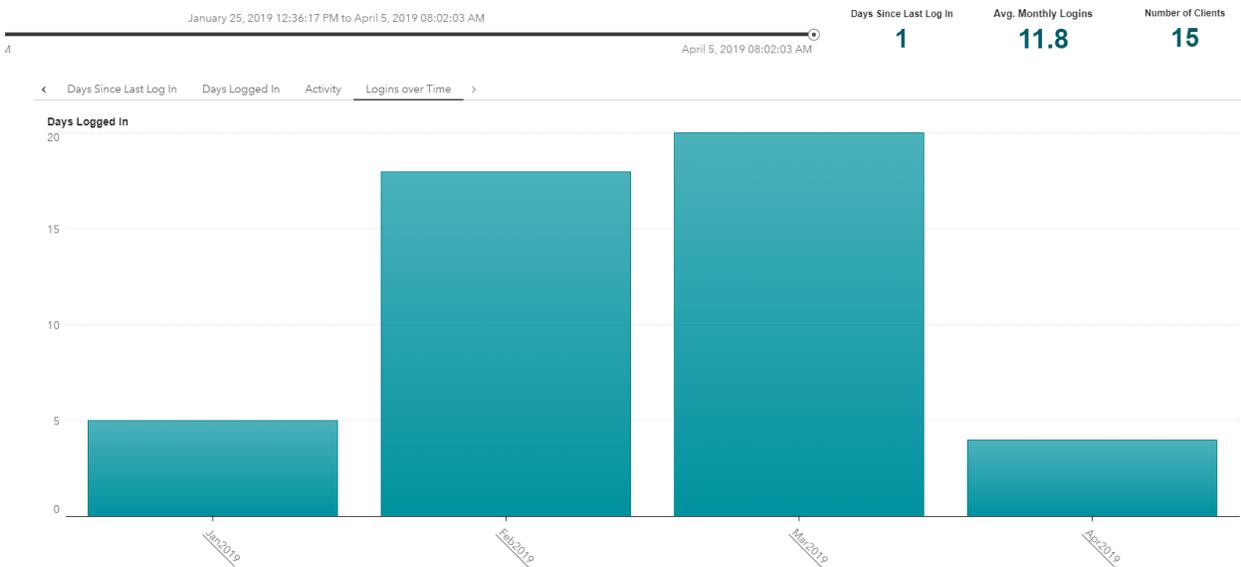


Figure 3. Activity over month

Activity over weekdays and time can give more granular information on usage. This information can be used to schedule maintenance or routine jobs during low usage windows and targeted messaging during high usage windows.

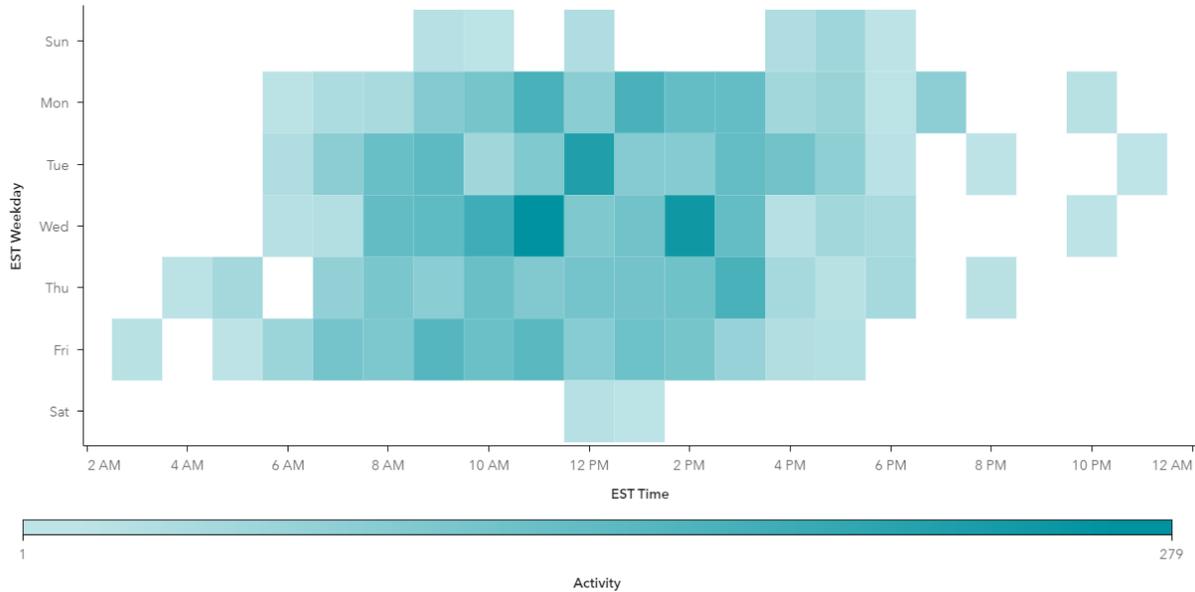


Figure 4. Activity time heatmap

CONCLUSION

We covered different KPI's and visuals that can be used to proactively track usage and take appropriate actions whether it is a client that's accessing the report or internal users. This paper can also serve as a guide on archiving any other datasets, creating more compounded calculated items and plotting them.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Anurag Kulkarni
 RevSpring
 Anuragkulkarni4@gmail.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.