Taskscheduler and Crontab with R

Aritra Biswas

July 19, 2016

From the point of view of a data scientist this is important to explore a phenomena over time. While conducting an experiment with social networking sites websites I've found these beautiful tools which can run r scripts automatically after a per specified time by the user. Here is a tutorial explaining task scheduling process for Windows 10 and Linux Debian Jessie (This is the latest OS for Raspberry Pi).

For windows:

To demonstrate this process we need a small R script which will be executed after specific time (i.e. every 5 minutes) on windows using the default application Task scheduler. For demonstration purpose we are considering this code:

Let us save the above code a file called systime.R. Note down the path of the folder which contains this file. In my case this case the file is saved the file in `C:\Users\Aritra\Desktop`.

Now go to **Control Panel-> Administrative Tools -> Task Scheduler**.

Select Create Basic Task, provide a name and a description of the task so that this can be easily recognized later. Click on next.

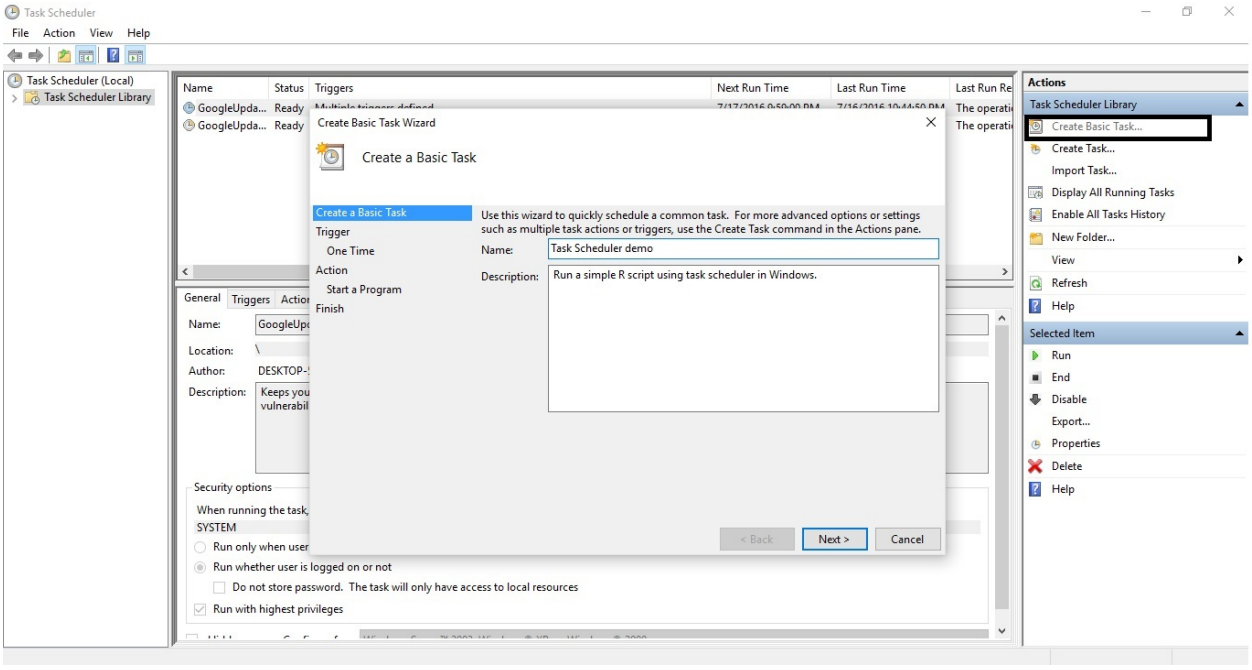


Figure 1:

Select **One time** and click on next. This will create a one time job initially. Later on we will change the task so that it will be executed after a specified time. The time lag on which the script will be executed (i.e.once a day, week or month) will be specified later.

Select the date and time when the task will be executed for the first time. This is recommended to set the time a few minutes ahead from current time.

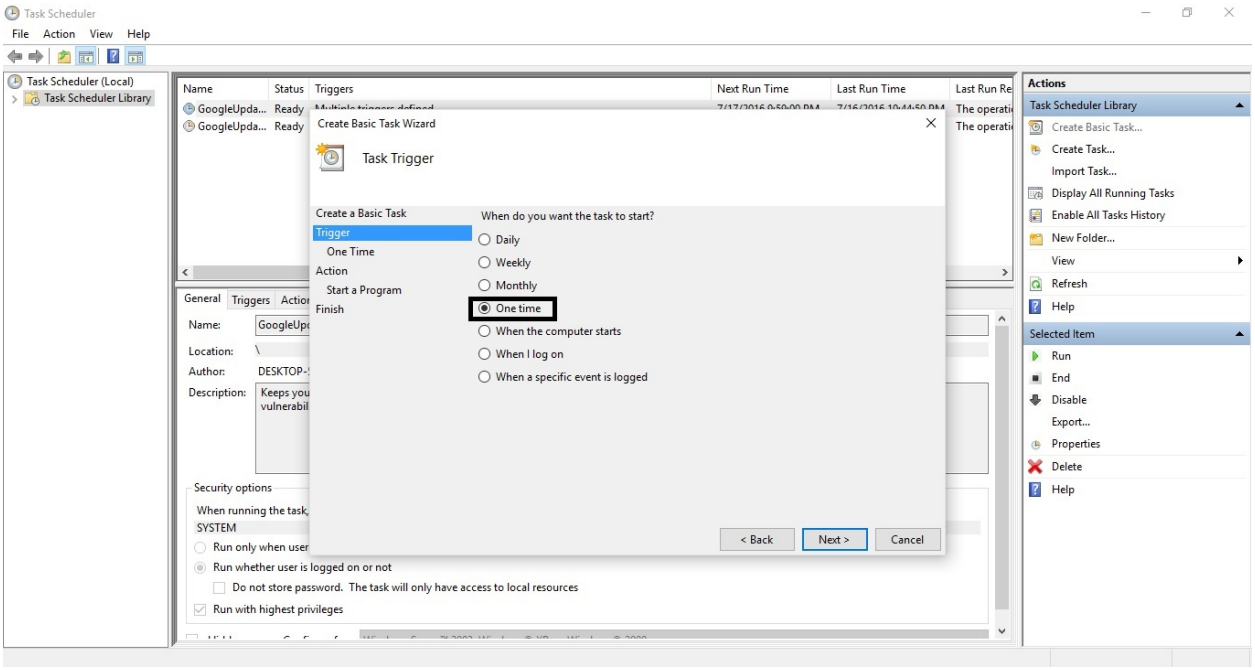


Figure 2:

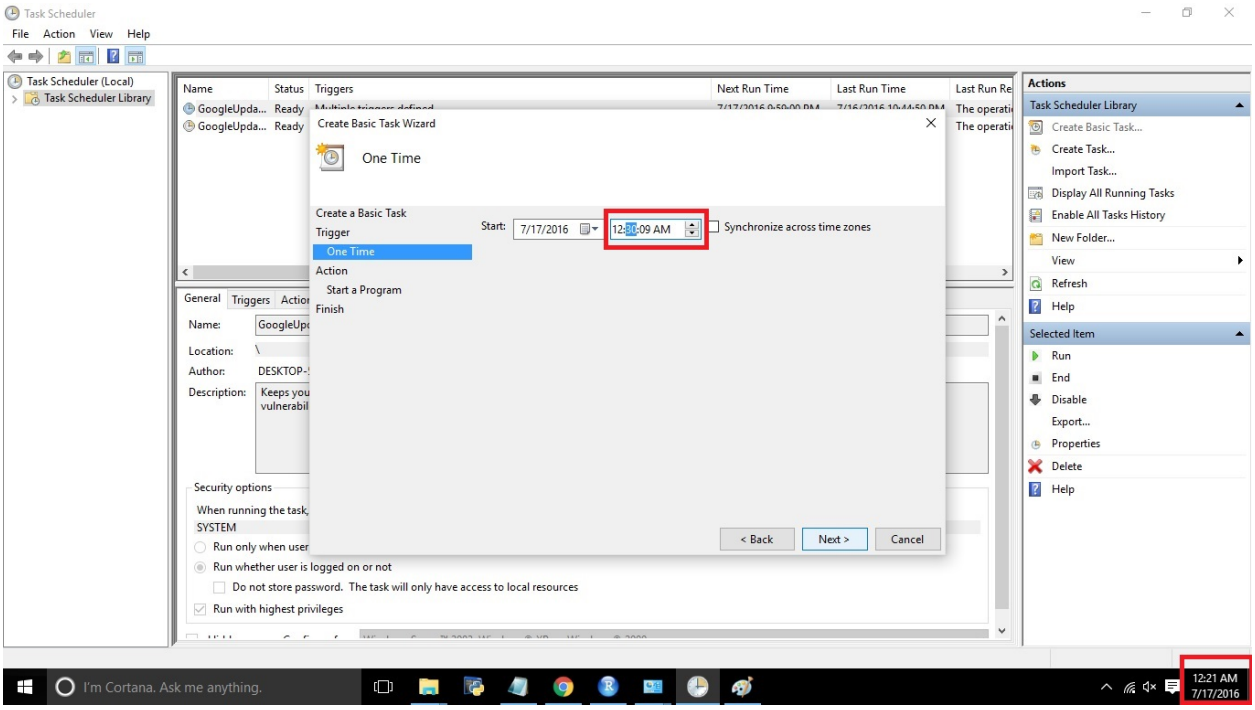


Figure 3:

Under Action choose **Start a program, click on next**.

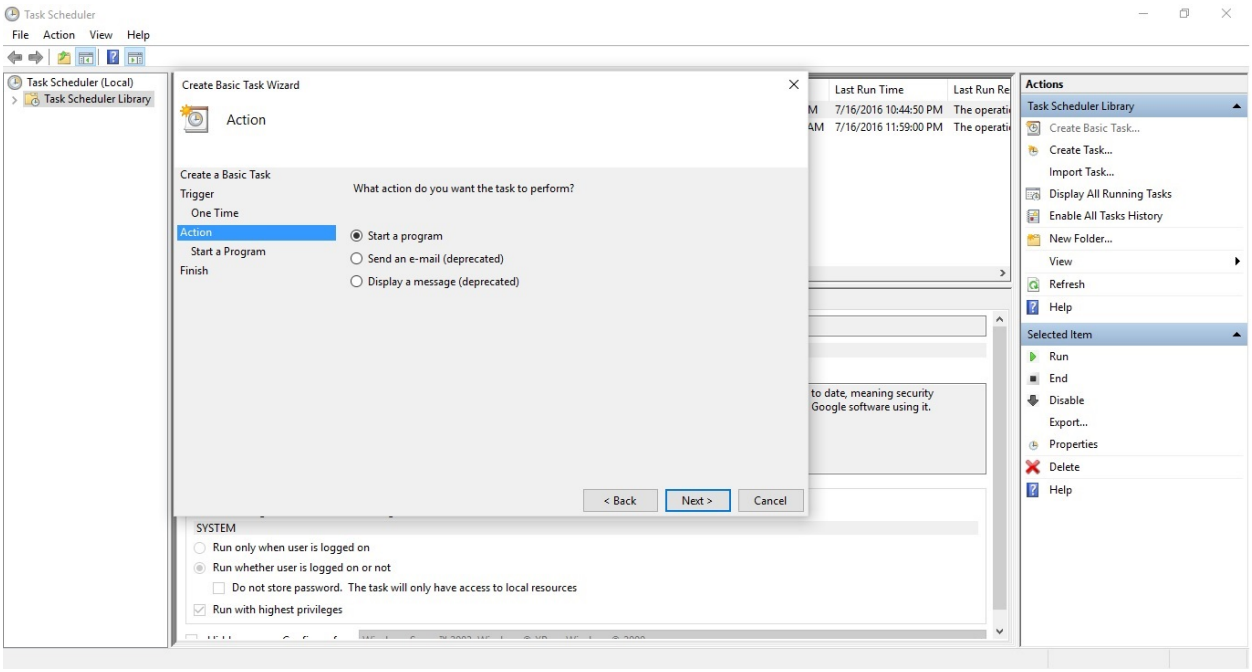


Figure 4:

Now, a .bat file has to be created. It will be executed from command prompt in windows. The analog of this .bat file in Linux is .sh. Write the following lines of code in a notepad file:

```
@echo off

R CMD BATCH C:\Users\Aritra\systime.R
```

Save this file as a .bat file. Here it is saved as systime.bat. To check whether this .bat file is working properly or not, one may simply click on the .bat file. This should execute the R script from command prompt and generate a .csv and a .Rout file. The.csv file should contain the system time of execution of the R script and .Rout file should contain all the text printed out in R terminal. This may be useful to diagnose error if there is any.

Here the path `"C:\Users\Aritra\systime.R"` is the location of my script, this path must be change with the file location of your r script.

In task scheduler, **Browse the .bat file** in the field of Program Script. Click Next.

Check in Open the Properties dialog. Click on Finish.

Under General tab, **check in Run with highest priviliges**.

Change and shift to Triggers tab. Select task and click on edit.

Check in Repeat task every and select duration. When you click on a time lag then you can simply change the value from keyboard. This will enable to execute task at and given time. Further options such as Expire can be edited if required. If not, simply click on OK.

This .csv file will be generated in the directory of r script.

This .Rout file will be generated in the same directory. This will contain all the text printed in the R terminal.

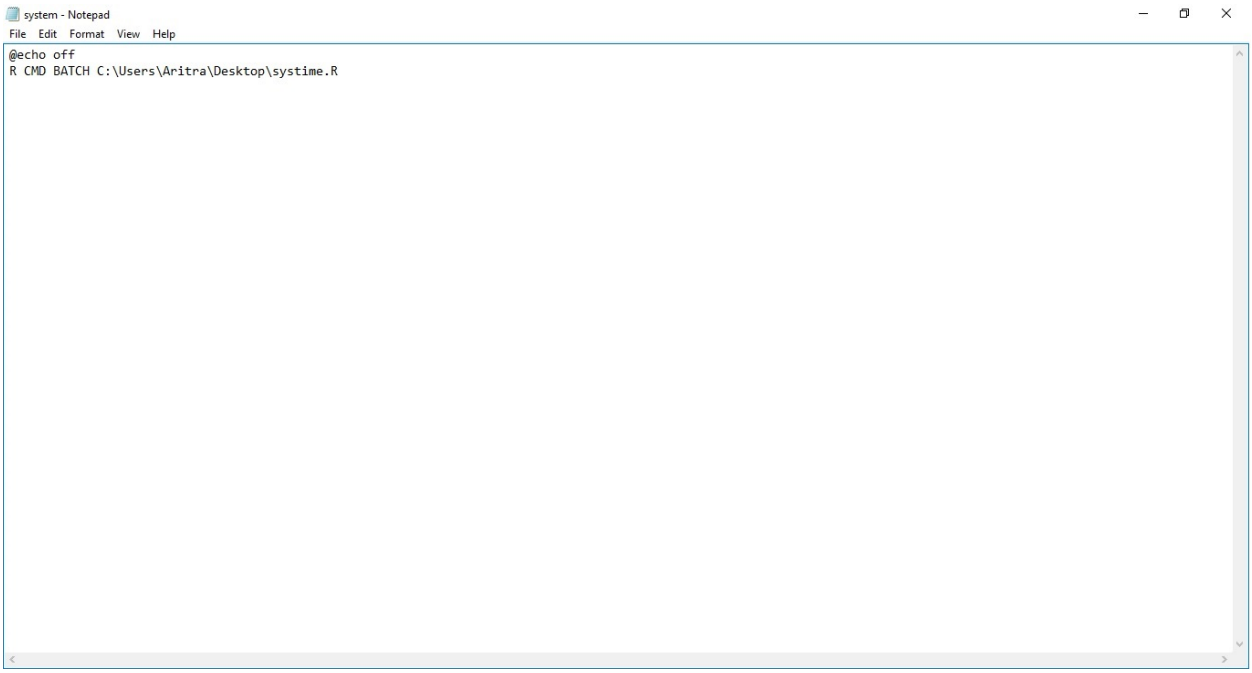


Figure 5:

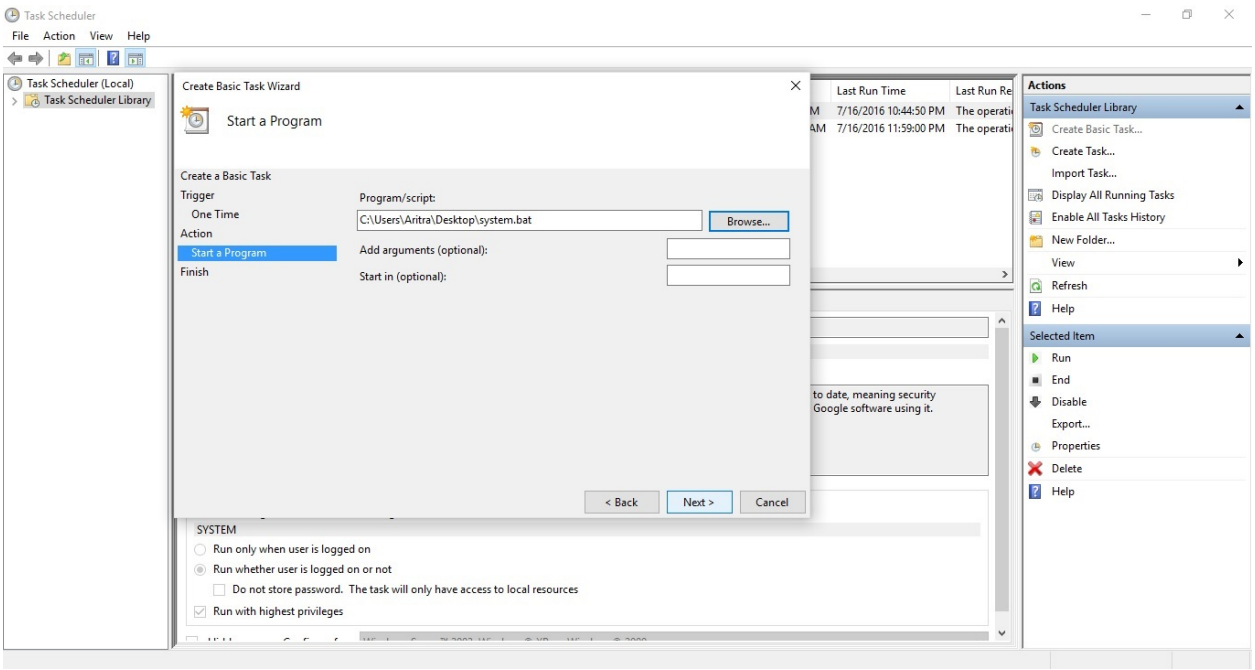


Figure 6:

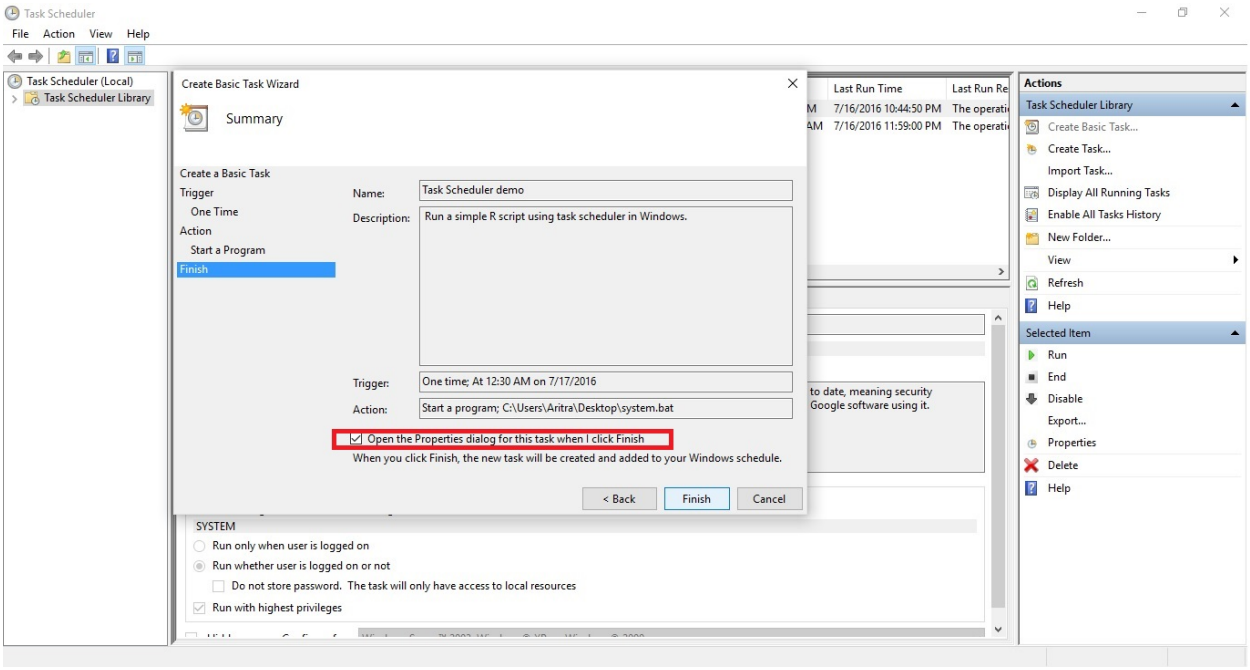


Figure 7:

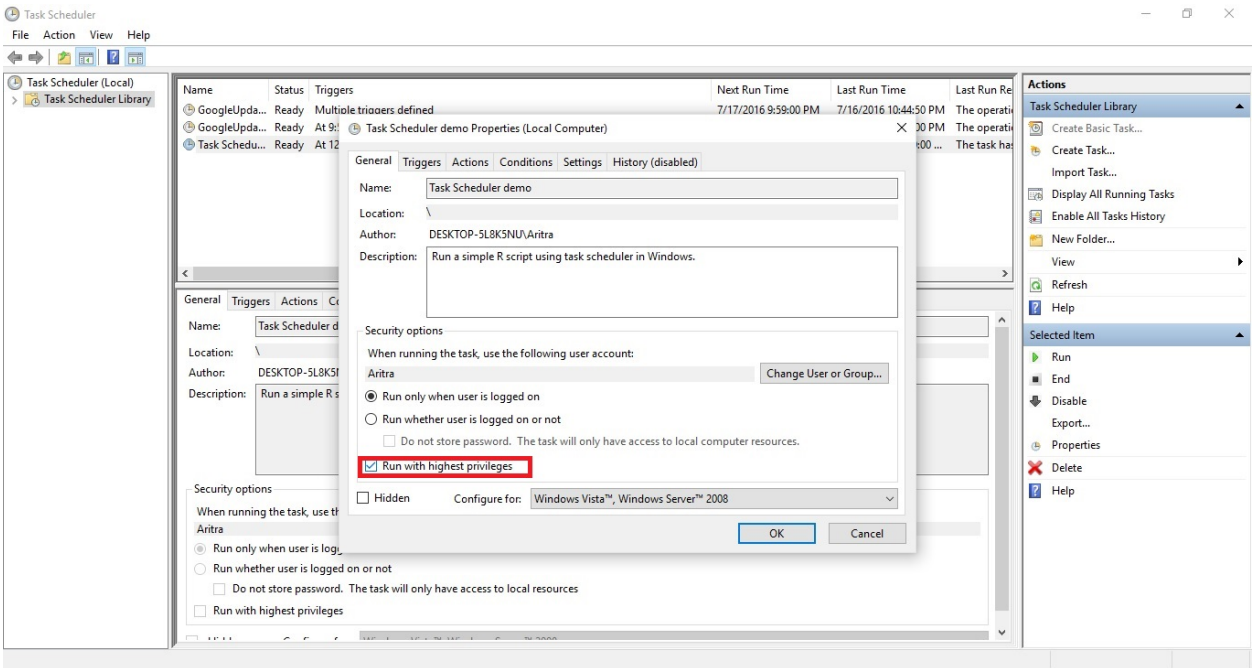


Figure 8:

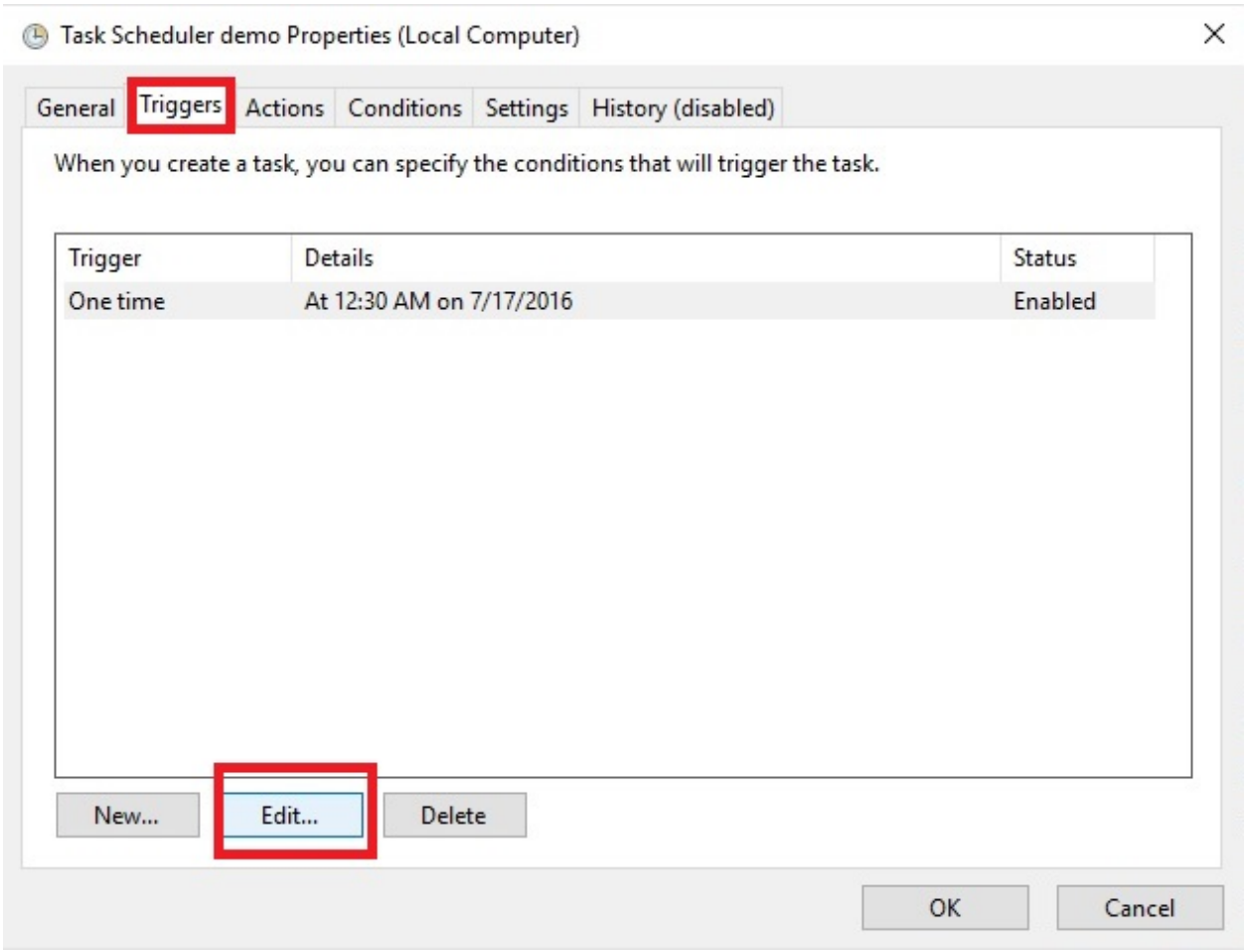

Task Scheduler demo Properties (Local Computer)
General
Triggers
Actions
Conditions
Settings
History (disabled)
When you create a task, you can specify the conditions that will trigger the task.
Trigger
Details
Status
One time
At 12:30 AM on 7/17/2016
Enabled
New...
Edit...
Delete
OK
Cancel

Figure 9:

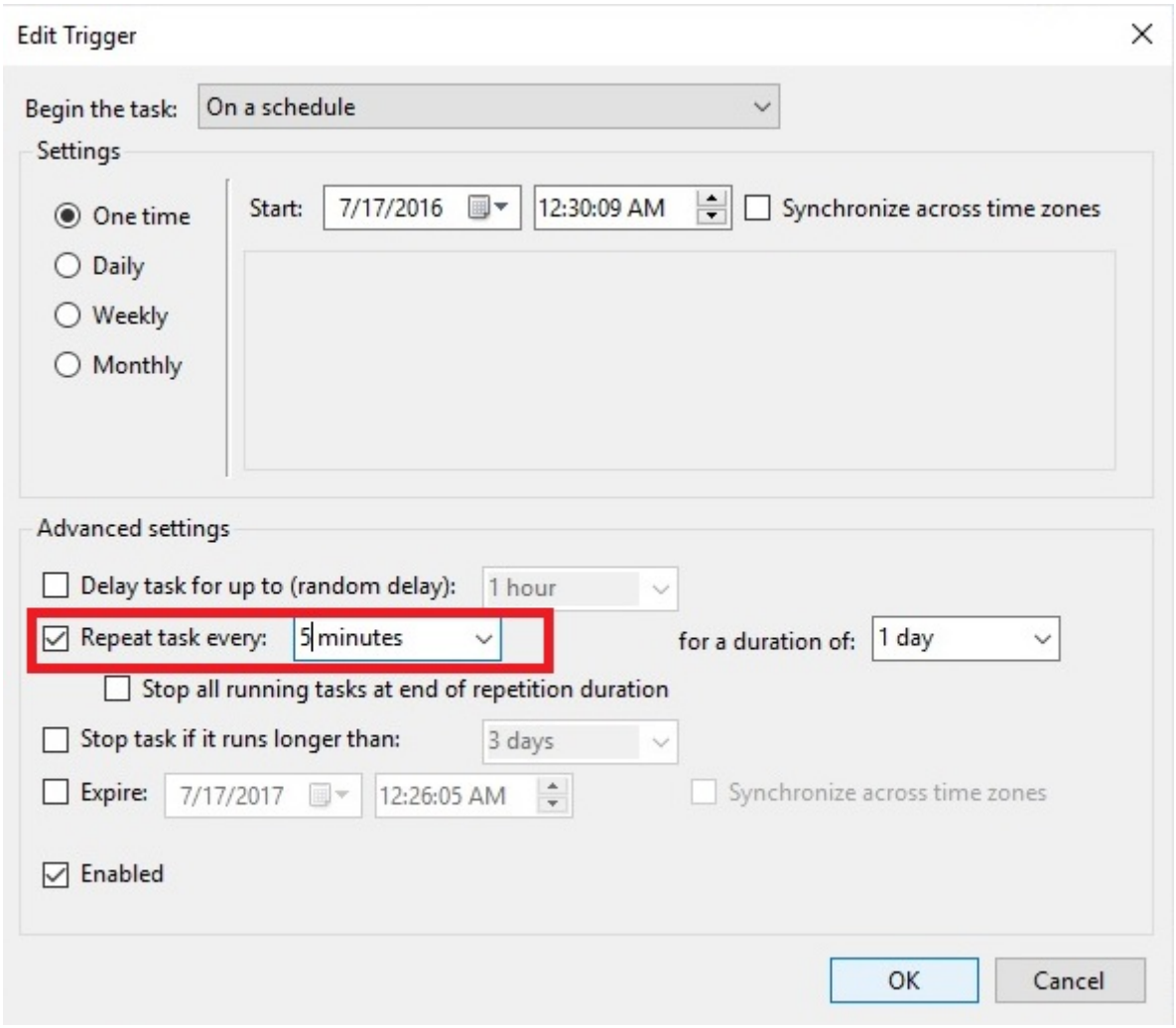

Edit Trigger
Begin the task:
On a schedule
Settings
One time
Daily
Weekly
Monthly
Start:
7/17/2016
12:30:09 AM
Synchronize across time zones
Advanced settings
Delay task for up to (random delay):
1 hour
Repeat task every:
5 minutes
for a duration of:
1 day
Stop all running tasks at end of repetition duration
Stop task if it runs longer than:
3 days
Expire:
7/17/2017
12:26:05 AM
Synchronize across time zones
Enabled
OK
Cancel

Figure 10:

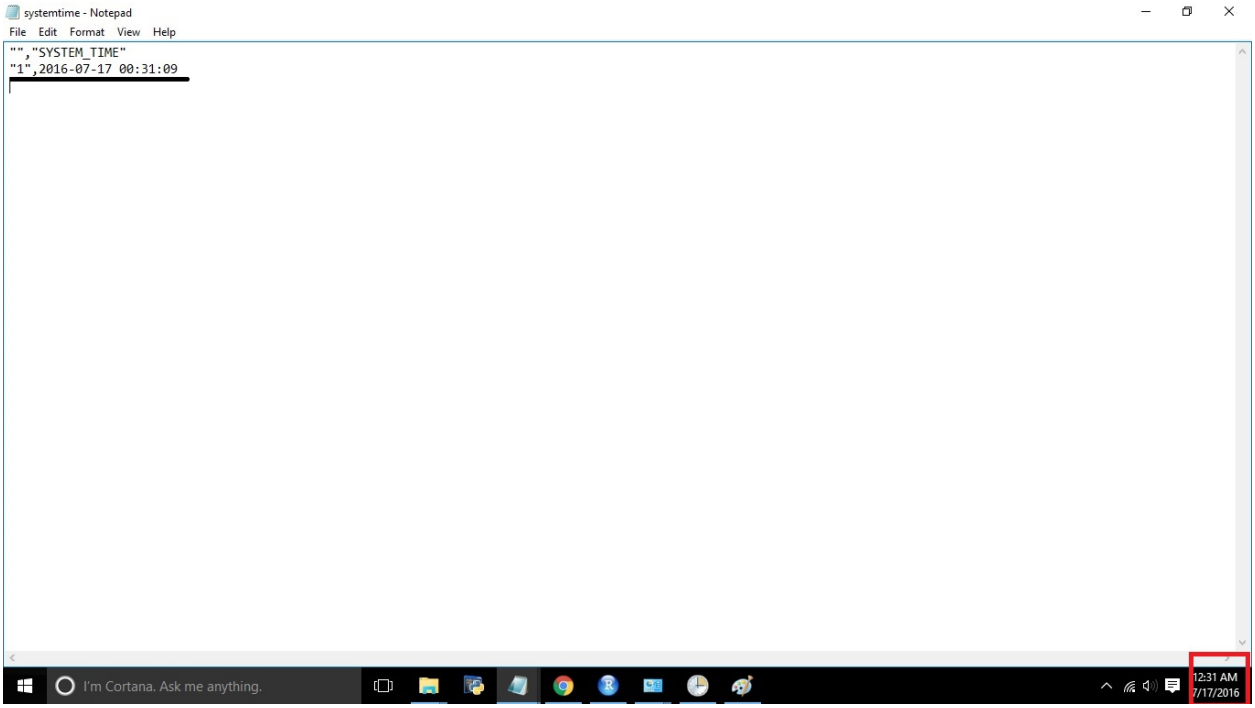


Figure 11:

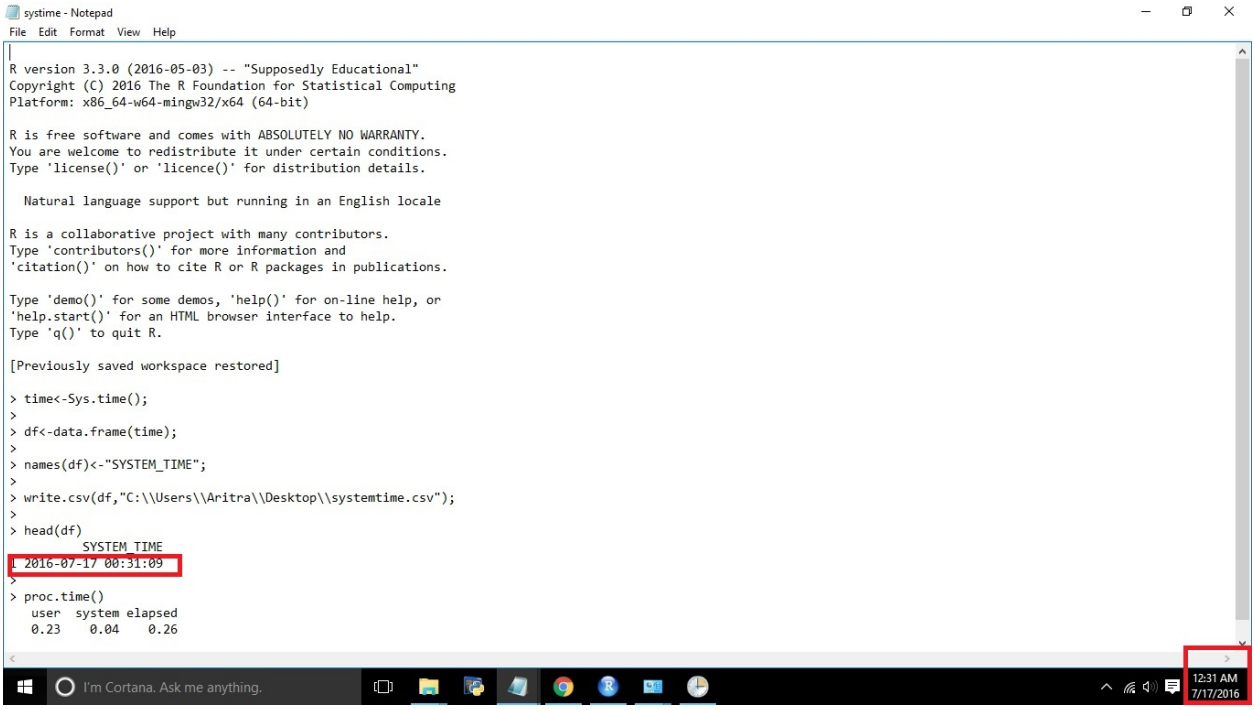


Figure 12:

For Linux:

Now here we will create the same thing in a Linux environment on raspberry pi. The OS in this device is Debian Jessie. In order to run R script in this device we need to create a .sh file which will contain the bash or terminal script to run R script. At first create the systime.R file as mentioned earlier.

Creating a shell executable script:

Now save this file to into a location. Note down the path. This will be required later. I've save my file on Desktop. Now open terminal, change directory using cd. In my case,

```
cd Desktop
```

Now, open a text editor. I'm going to use Leaf pad, which comes pre-installed with raspbian. Type the following line of codes:

```
#! /bin/bash
```

```
R CMD BATCH /home/pi/Desktop/systime.R
```

and save this file as systime.sh.

Changing permission:

Now, make the .sh file executable by executing the following command from terminal

```
sudo chmod +x systime.sh
```

or simple right click on the script and go to **Properties -> Permissions -> Allow executing file as program**.

To verify the script is running simply click on the .sh file. The script should be executed and the .csv and .Rout file should be generated.

Using crontab:

As task scheduler in windows, Linux has its own version of called crontab. To run it use the following command in terminal:

```
sudo crontab -e
```

If crontab is running for the first time, choose nano to edit the file. Add the following lines of code at the end of the file. # at the beginning of a line means that the line is commented out. Those lines will not be executed. Do not put # in front of the line.

```
* * * * * /home/pi/Desktop/systime.sh
```

This line will direct the operating system to run the .sh file after every one minute. There is a option to change this duration. To see how to change this time duration in crontab simply read the commented out lines printed in the terminal.

After the required changes save it using Ctrl+X. Choose Yes, to save changes.

Now the R script should run from terminal and the latest run time of the script should be saved in a .csv file and as well as in the .Rout file in the selected directory.

Application in industry:

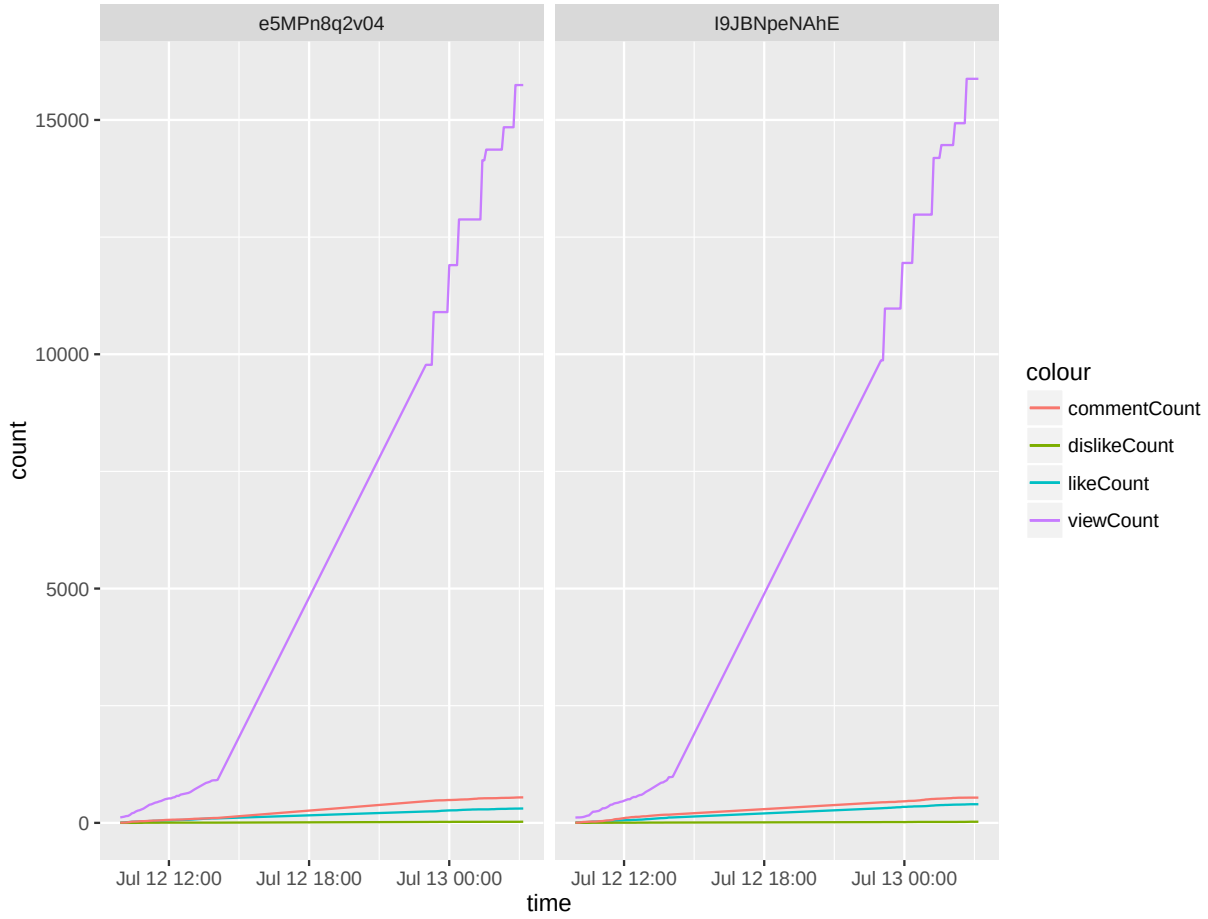
Here I've used `get_video_data()` from the YouTubeR package to track the change in video statistics. What the function does is it downloads the data related to a video in a give period of time. Here I'm tracking videos posted in Times Now's News hour debate on 12nd July. Here I've successfully obtained the change in viewCount, likeCount, dislikeCount and commentCount for two videos posted in the channel on above mentioned date in 5 mins interval from 10 A.M..

Here what the data looks like:

```
##      items.id
## 1 e5MPn8q2v04
## 2 I9JBNpeNAhE
## 3 e5MPn8q2v04
## 4 I9JBNpeNAhE
##                                                                                items.snippet.title
## 1      Kashmir Clashes Due to Terrorist 'Burhan Wani's' Death: The Newshour Debate (11th July 2016)
## 2 Hafiz Saeed & Syed Salahuddin Hold Memorial for Burhan Wani: The Newshour Debate (11th July 2016)
## 3      Kashmir Clashes Due to Terrorist 'Burhan Wani's' Death: The Newshour Debate (11th July 2016)
## 4 Hafiz Saeed & Syed Salahuddin Hold Memorial for Burhan Wani: The Newshour Debate (11th July 2016)
##   items.snippet.channelTitle items.statistics.viewCount
## 1        The Newshour Debate                       116
## 2        The Newshour Debate                       113
## 3        The Newshour Debate                       124
## 4        The Newshour Debate                       116
##   items.statistics.likeCount items.statistics.dislikeCount
## 1                          7                             0
## 2                          7                             1
## 3                          9                             0
## 4                          8                             1
##   items.statistics.commentCount                time
## 1                             5 2016-07-12 09:55:01
## 2                             5 2016-07-12 09:55:02
## 3                             6 2016-07-12 10:00:03
## 4                             6 2016-07-12 10:00:03
```

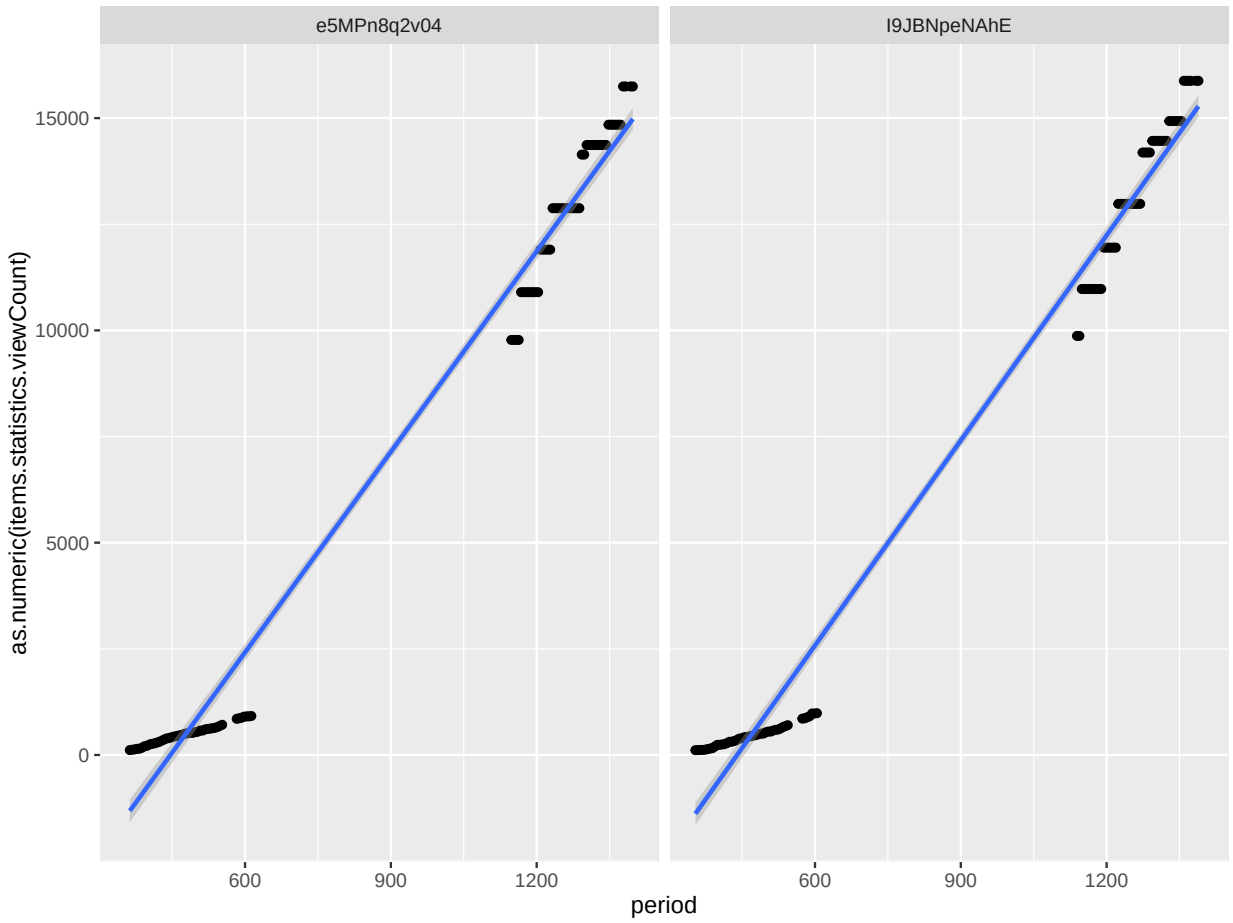
Visualization:

Here, viewCount, likeCount, dislikeCount and commentCount has been plotted against time. The x-axis and y-axis represents time and count respectively. It shows change in count over time.



Fitting a linear model:

Here, we are fitting a LM on viewCount w.r.t time. **Time=Current time-Published.At** . Model does not fits the data well because of non-linearity, but it definitely shows the way forward.



It's just the beginning

There are so many things can be done with this. This is hard to list them. Soon I'm going to share some of the applications based on this.

Reminder:

My computer was turned off during 2PM to 11PM on that day that's why there a jump in the graph at the same time. This can be considered as a reminder that in order to obtain data continuously **one must turn on a computer for a cetain period of time**. It is true for internet connectivity as well depending on the requirement of the program.

This is just a simple example of where crontab can be used. There are many more things can be done using crontab, shiny and R. I'm going to present some examples related this topic and shiny soon.

The source code for this is available in here and a slider version of this presentation is available here.