Assignment 3: Critical Color Change (100pts)

Goal of this assignment:

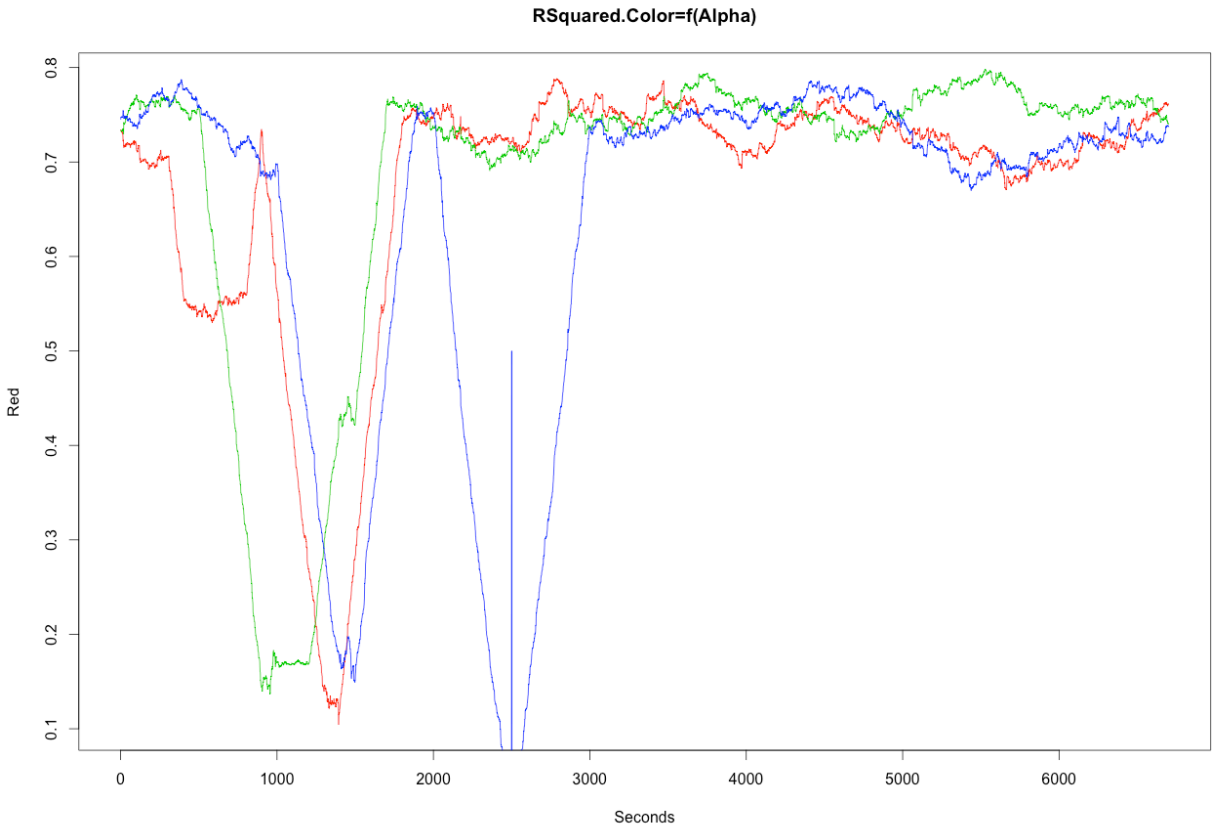
- Working with linear regression
- Working with thread

You will be receiving 3 colors and one Alpha in a stream. The goal of this exercise is to find the critical change in color in a video lasting 2 hours. Each frame being received every second. The colors are correlated with Alpha.

Task 1(10 pts): With the library of your choice, read the csv file
You can potentially use the following code:

```
import pandas as pd
import numpy as np
df = pd.read_csv('C:\\dataanalytics\\asssignment-session2.csv')
```

Task 2(20 pts): For each color, calculate the R-squared with a rolling window of 500 seconds. You will do a regression linear between one given color and Alpha on a 500 second-vector. Therefore, during the first 500 seconds, you will not have any regression linear data. The chart below starts at 0 (being +500 seconds) and represents the R-squared for the 3 colors.



Task 3(20pts): Find the approximate seconds you encounter problems of correlation between the colors and the alpha.

Task 4(30pts): The goal is to detect bad correlation in real time. Therefore you need to parallelize the linear regression for the three colors to be faster. You should find the same results as the task 3.

```
totalThreads=Number.of.colors

def ParallelRegression(threadnum):
    for i in range(threadnum):
       res[:,i] = getRquared(sparse.linalg.spsolve(…))

threads = []
for threadnum in range(totalThreads):
    t = threading.Thread(target=ParallelRegression, args=(threadnum))
    threads.append(t)
    t.start()

for threadnum in range(totalThreads): threads[threadnum].join()
```

Task 5(20pts): Instead of using R squared, can you use the lower/upper bounds of the confidence interval to find the same results? Can you implement the threaded version?