Re-examining Marginal use of Curiosity Rover

Kevin Lee
USC ‘22
Agenda

- Who am I
- What is the Curiosity rover
- How I used Marginal Data Analysis to improve efficiency of the rover
Intro

- University of Southern California - B.S in Computer Science/Physics
- Expected graduation: May of 2022
Google DSC Leader

- Developer Student Club founder/leader at USC.
- Working together to build solutions for local problems in the community.
- Access to Google Developer resources.
Curiosity Rover

- 8 years old (launched 2011)
- 2586 Sols on Mars
- 13.42 miles traveled
- Over 600k photos taken
Discoveries

- Persistent evidence of liquid water
- Suitable home for Life (organic molecules, nitrogen, sulfur, oxygen, carbon)
- Methane discovery (2019)
  - Methane can be produced by living organisms or by chemical reactions between rock and water
- And much more....
Goal

- Develop a tool that analyzes the marginal use of the Curiosity rover actual vs. estimated from EVR files.

Sol 2490 left Mastcam image of SAM inlet covers.
Where things started

- EVR (Event Reports)
  - Chill_get (chill_get_evrs -t LST -b Sol-2452M00:00:00 -e Sol-2453M00:00:00 -m)
  - Spazz/Elasticsearch used
import os, sys, json, re
from math import floor
import elasticsearch, urllib3
from elasticsearch import helpers

pwd = os.getcwd()

sys.path.insert(0, '{}/msl-datalytics/src/'.format(pwd))

from spazz import *
import timeit

start = timeit.time()

# from msldatalytics.src.spazz import *
# from spazz import *

es = elasticsearch.Elasticsearch('https://nasa.gov', sniff_on_start=False)
Grabbing EVR files

```python
numMarginDatabaseErrors = 0
numMissingActuallsErrors = 0
numMultipleActuallsErrors = 0
# initialize Spazz for a future query
spazzObj = spazz({'beginTime': "Sol-" + str(starting_Sol) + "00:00:00", 'timeType': "LST"})
#initialize the query
# the "not" line should remove all instances of sub_00000
# This query is essentially a frame work for the elasticsearch to base off from. It continuously
# match this query.
query = {
    "query": {
        "match": {}
    }
}
```
Lots of useless info
Level of priorities

1) Complexity
2) Reliability
3) Maintainability
1. Complexity

Key Factor

- Efficiency
- Ease to read/understand
- Ability to pick up fairly easily
2. Reliability

- Probability of failure-free software
- TDD (Test Driven Development)

The probability of failure is given as

\[
F(t) = 1 - R(t) = \int_{0}^{t} f(s) \, ds
\]

and the failure rate (FR) indicating the conditional probability that a failure per unit time occurs in an interval \([t, t + \Delta t]\) given that a failure has not occurred before time \(t\) is given by

\[
FR = \frac{F(t+\Delta t) - F(t)}{\Delta R(t)}.
\]
3. Maintainability

- Important because it is approx. 75% of the cost related to project.

Marginal Analysis Tool

The task is to develop a tool that analyzes the marginal use of the Curiosity rover actual vs. estimated from EVR files. If possible, James wants to look deeper into the individual sub-master processes that happens within each block for better efficiency. Eventually, make it simple enough to be a command to feed in intervals and generates maintainable and document thoroughly for someone else to pick up.

Getting Started

1. Copy the repository in a desired location.

   git clone https://github.jpl.nasa.gov/kevin/MMG

Prerequisites

Python 2 is required to run this program. Have internet access to download packages if necessary.

I am running Python 2.7.10

Run this to find out what python version you are running

python -c "import sys; print(sys.version)"

Required packages:

https://github.jpl.nasa.gov/MMG/tree/data/lyrics

https://github.jpl.nasa.gov/MMG/tree/data/extract

https://github.jpl.nasa.gov/MMG/tree/data/lyrics

Space is built from real-data repository that contains shared resources that utilize the Analytics cloud infrastructure to query/index, and analyze MS. Data.
Complications

An error has occurred

The application failed to start (exited with code 1).

DEPRECATION: Python 2.7 will reach the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 will no longer be maintained.

Error in value quale[3L](cond):
  SyntaxError: invalid syntax (classes.py, line 26)

Detailed traceback:
  File "<string>" , line 1, in <module>
Calls: local ... tryCatch -> tryCatchList -> tryCatchOne -> <Anonymous>
Execution halted
Making it in a Web-Dashboard style

- Graphs created using javascript.
- Learning D3.js
  - Document Object Model (DOM)
  - Scalable Vector Graphics (SVG)
  - JavaScript
  - Geometry
  - Color Spaces
  - Data Structures
  - Standard Model

Example graph made

```javascript
var shots = d3.select("svg")
  .selectAll("g")
  .data(data)
  .enter()
  .append("g")
    .attr("class", "shots")
    .attr("transform", function(d){
      return "translate(" + 10 * d.converted_y + "," + 10 * d.converted_x + ")";
    })
    .on("mouseover", function(d){
      d3.select(this).raise()
      .append("text")
      .attr("class", "playername")
      .text(d.player)
    })
    .on("mouseout", function(d){
      d3.selectAll("text.playername").remove();
    })
```
Achievements

- Hands on experience
  - Reading Code
  - Proper Documentation
- Skills acquired
  - JavaScript
  - HTML
  - D3.js
    - DOM
    - SVG
- Understanding of software process
Questions

USC ‘22
Computer Science Candidate