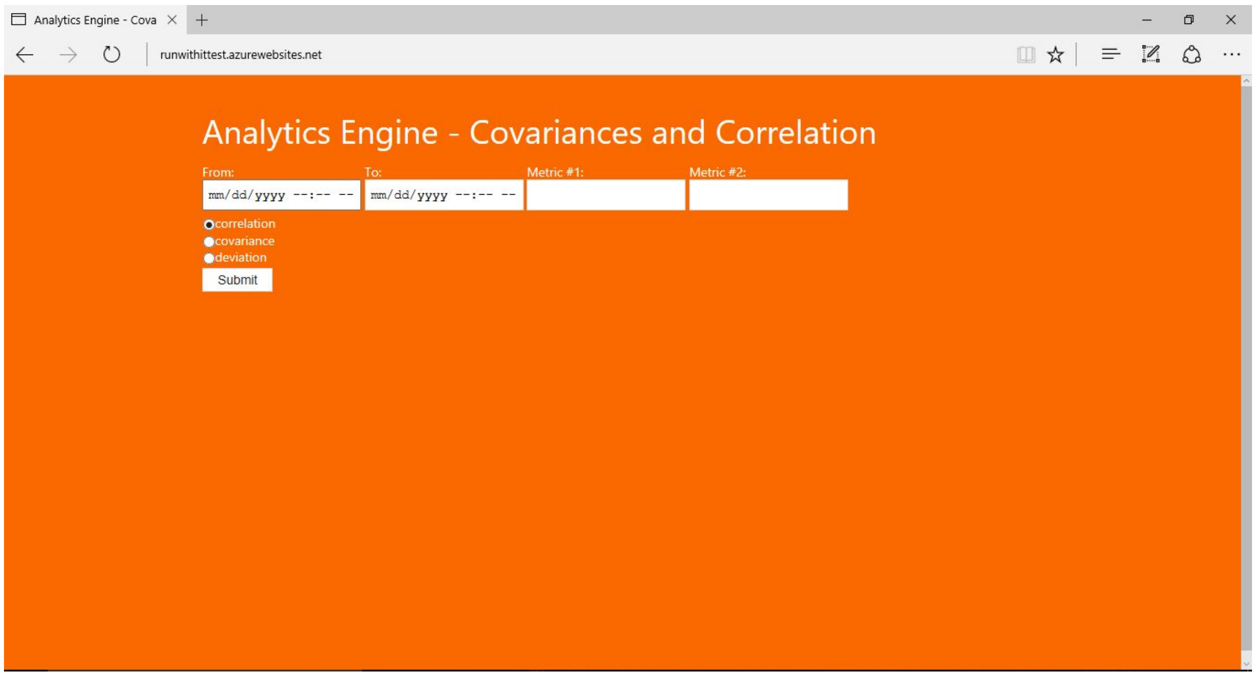
USER MANUAL FOR ANALYTICS ENGINE

This is for the User Interface (UI) interfaces of both the command line interface and the webpage UI.

WEBPAGE User Interface (UI):

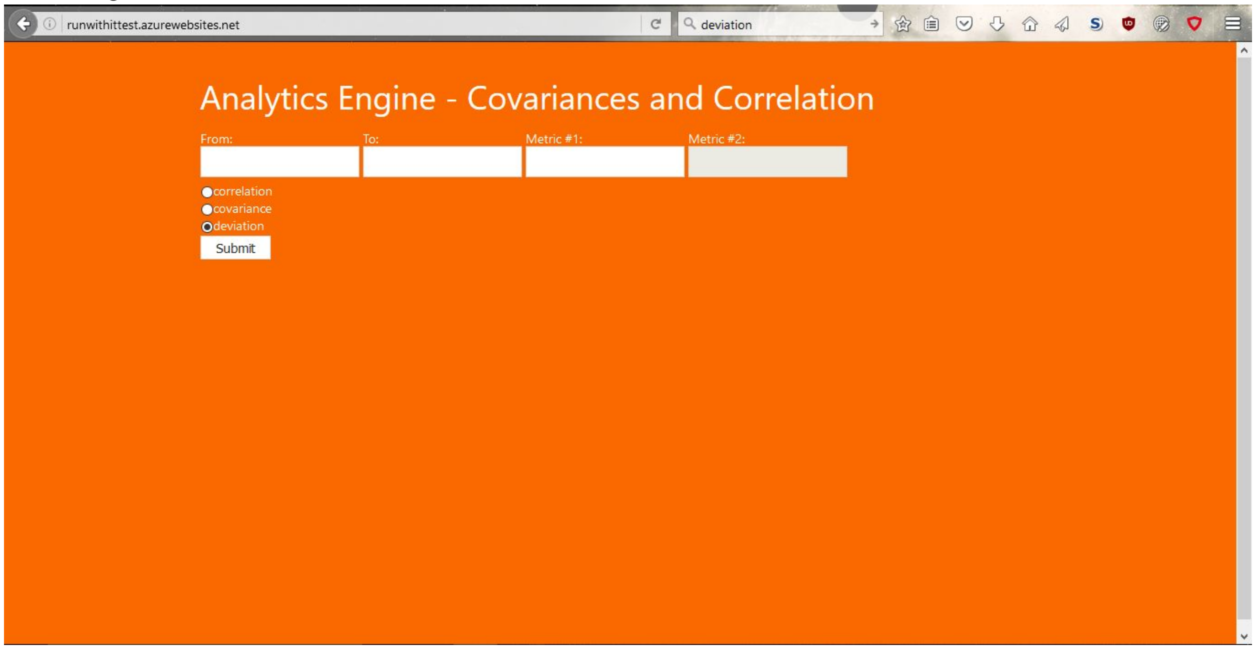
The webpage UI is intended for minor functionality without the using of certain customization for the metric analysis. As well, it will be for teaching new users how the system works and what to look for. It also is another option to test statistics remotely without having to install the core engine on your local computer (machine).

This is the start page of the webpage UI:



This is the display for both covariance and correlation.

Deviation UI: Metric 2 is greyed out as it is not needed as you only need to have to specify the time range of the metric1 and and the metric itself.


runwithittest.azurewebsites.net
deviation
Analytics Engine - Covariances and Correlation
From:
To:
Metric #1:
Metric #2:
correlation
covariance
deviation
Submit

Features:

From: -> This allows the user to specify the start time in which to analyze the metrics.

To: -> This allows the user to specify the end time in which to analyze the metrics.

- Both of the above options can be specified in the line or from a drop down calendar*:
  - Time format is MM/DD/YYYY HH:MM
  - *Drop down calendar for Edge, and formatting in Chrome

Metric1/2: -> This allows the user to enter two different metrics for comparison statistics from the radio buttons.

Radio Buttons: ->

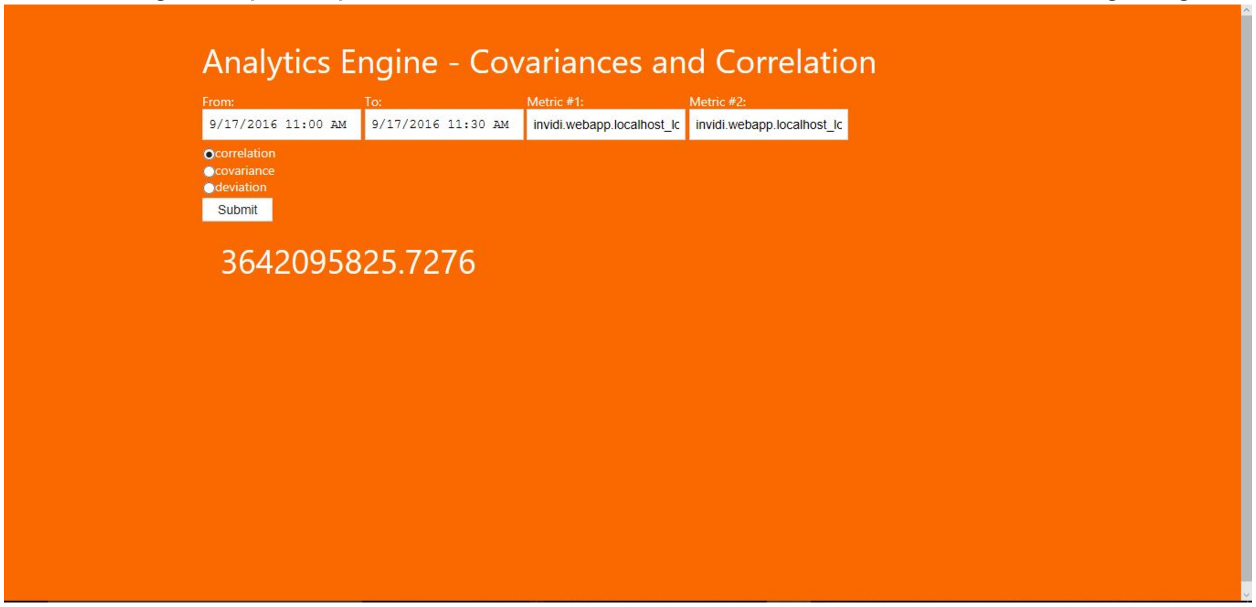
Correlation: Correlation is any of a broad class of statistical relationships involving dependence, though in common usage it most often refers to the extent to which two variables have a linear relationship with each other. Familiar examples of dependent phenomena include the correlation between the physical statures of parents and their offspring, and the correlation between the demand for a product and its price. Correlations are useful because they can indicate a predictive relationship that can be exploited in practice. For example, an electrical utility may produce less power on a mild day based on the correlation between electricity demand and weather. In this example there is a causal relationship, because extreme weather causes people to use more electricity for heating or cooling; however, correlation is not sufficient to demonstrate the presence of such a causal relationship (i.e., correlation does not imply causation).

Covariance: Covariance is a measure of how much two random variables change together. If the greater values of one variable mainly correspond with the greater values of the other variable, and the same holds for the lesser values, i.e., the variables tend to show similar behavior, the covariance is positive. For example, as a balloon is blown up it gets larger in all dimensions. In the opposite case, when the greater values of one variable mainly correspond to the lesser values of the other, i.e., the variables tend to show opposite behavior, the covariance is negative. If a sealed balloon is squashed in one dimension then it will expand in the other two. The sign of the covariance therefore shows the tendency in the linear relationship between the variables. The magnitude of the covariance is not easy to interpret. The normalized version of the covariance, the correlation coefficient, however, shows by its magnitude the strength of the linear relation.

Deviation: deviation is a measure of difference between the observed value of a variable and some other value, often that variable's mean. The sign of the deviation (positive or negative), reports the direction of that difference (the deviation is positive when the observed value exceeds the reference value). The magnitude of the value indicates the size of the difference.

Source: Wikipedia

The resulting example outputs for correlation and covariance will look like the following image:



Analytics Engine - Covariances and Correlation

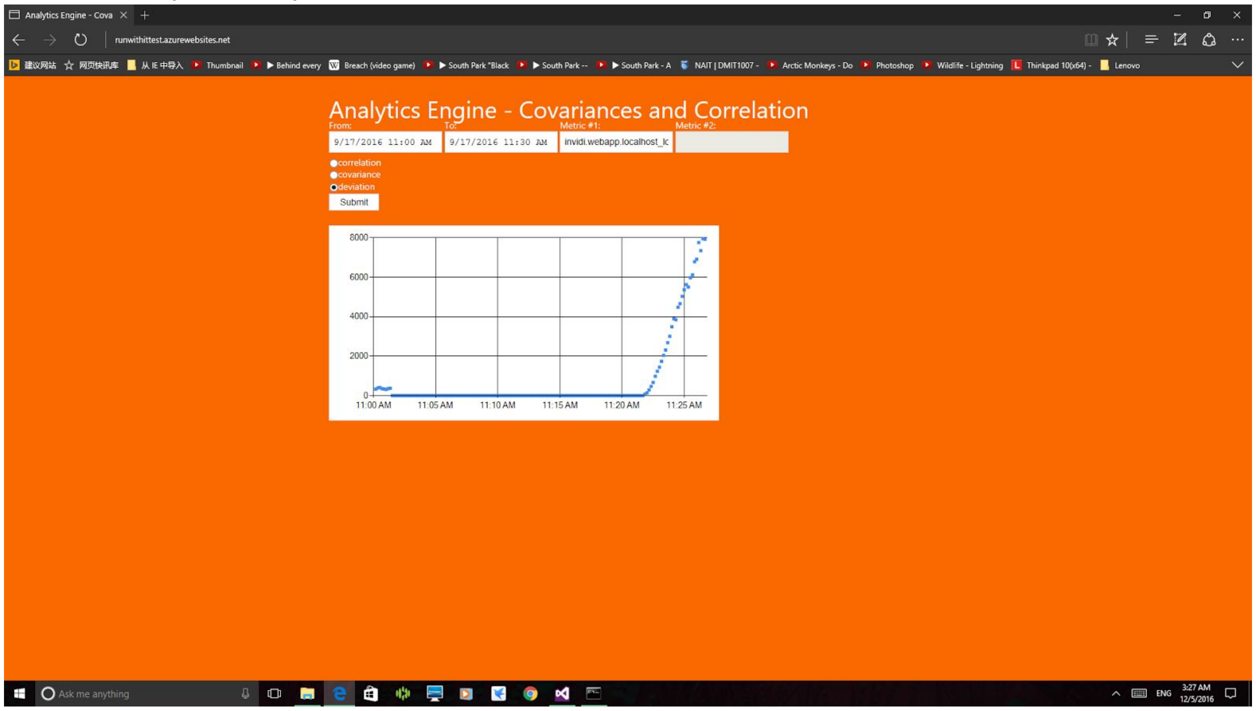
From: 9/17/2016 11:00 AM
To: 9/17/2016 11:30 AM
Metric #1: invidi.webapp.localhost_lc
Metric #2: invidi.webapp.localhost_lc

[✓ correlation]
covariance
deviation

Submit

3642095825.7276

Correlation and covariance will have a single outputed variable like the one above, but deviation will output a graph of timestamps of points that deviate from a default median of a single metric given in Metric1 box.
Deviation output example:



Analytics Engine - Covariances and Correlation

From: 9/17/2016 11:00 AM
To: 9/17/2016 11:30 AM
Metric #1: invidi.webapp.localhost_lc
Metric #2:

correlation
covariance
[✓ deviation]

Submit

Command Line Interface (CLI):

The command line interface is intended for advanced users who understand the data set and want a more powerful and customizable way of getting their data and seeing results. It is a more powerful version in comparison to the web based UI as the commands from earlier can be outfitted with certain options allowing for greater statistical accuracy and better output. An additional metric analysis tool exclusive to the command line interface is Entailment Search explained below. But we will first go through the same functions from the web based UI, but from the command line perspective.

Command Line Options:

NOTE: It does not matter the order in which you place the minor command line functions after you specify a major function, it will always perform the right actions given the postfix for the flags is in the correct format.

Minor Functions:

--metric1:

This allows the user to enter a data stream based on their database.

--metric2:

Same as --metric1, used in comparative analysis in correlation and covariance.

Example input:

- --metric1 IN.stb-sim.dean.RequestTiming.count \
  --metric2 "IN.stb-sim.dean.RequestTiming.count"

--m1_start/--m1_end :

This allows the user to specify the time frame in which to compare or get both metrics 1 and 2. Format is HH:MM_YYYYMMDD

Example input:

- --m1_start 17:00_20160921 \

--m1_end 18:00_20160921

*Optional: If you would like to compare metrics at a different time intervals you can use **--m2_start/--m2_end,** which has the same function and format as the definition above but applied to metric2.

-n:

This flag represents Normalization of metrics. Normalization: It's the process of casting the data to the specific range,like between 0 and 1 or between -1 and +1. Normalization is required when there is big differences in the ranges of different features. This scaling method is useful whenthe data set does not contain outliers.Currently using the normalization formula:

$$z(i) = \frac{(x(i) - min(x))}{max(x) - min(x)}$$

In which z(i) stands for the normalized value of x at position i, x(i), and with min(x) and max(x) representing the min and max values of the given set of metrics.This option is off by default but if you would like to normalize the data before being compared against, just apply this flag to normalize both metrics being analyzed under the major functions covariance, or correlation.

Source: https://docs.google.com/document/d/1x0A1nUz1WWtMCZb5oVzF0SVMY7a_58KQulqQVT8LaVA/edit#

Example Input:

- -node dist/cli.js correlation \
   --metric1 IN.stb-sim.dean.RequestTiming.count \
  --metric2 "IN.stb-sim.dean.RequestTiming.count" \
  --m1_start 17:00_20160921 --m1_end 18:00_20160921 \
  -n

- -node dist/cli.js covariance \
   --metric1 IN.stb-sim.dean.RequestTiming.count \
  --metric2 "IN.stb-sim.dean.RequestTiming.count" \
  --m1_start 17:00_20160921 \
  --m1_end 18:00_20160921\
  -n

Major Functions:

Correlation:

(See above for a description; Radio Buttons: Correlation)

Here is an example of the basic correlation command:

- node dist/cli.js correlation \
  --metric1 IN.stb-sim.dean.RequestTiming.count \
  --metric2 "IN.stb-sim.dean.RequestTiming.count" \
  --m1_start 17:00_20160921 \
  --m1_end 18:00_20160921

Covariance:

(See above for a description; Radio Buttons: Covariance)

Example of covariance command:

- node dist/cli.js covariance \
  --metric1 IN.stb-sim.dean.RequestTiming.count \
  --metric2 "IN.stb-sim.dean.RequestTiming.count" \
   --m1_start 17:00_20160921 \
  --m1_end 18:00_20160921

Deviation:

(See above for a description; Radio Buttons: Deviation)

Example of deviation command:

- node dist/cli.js deviation \
  --metric1 IN.stb-sim.dean.RequestTiming.count \
  --m1_start 17:00_20160921 \
  --m1_end 18:00_20160921

Search**:

This main function applies a combination of both covariance and correlation against all metrics in the system. It outputs the metric name, correlation and covariance against the one entered in the input. Therefore this is useful in finding out what is most closely related in a system against your metric of interest, allowing for finer analysis later.

Example Input:

- search --metric1 IN.stb-sim.dean.RequestTiming.count \
  --metric2 "IN.stb-sim.dean.RequestTiming.count" \
  --m1_start 17:00_20160921 --m1_end 18:00_20160921

Example Output:

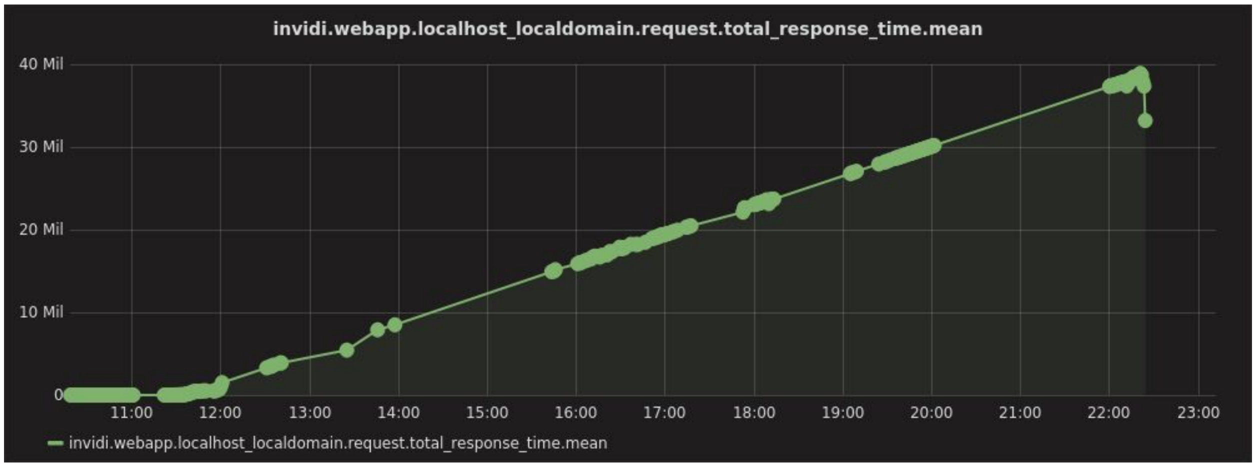
[["metricname",[1,2]], [“metricname2, [1.0, 2.4]], ...[“metricnameN, [Cor_N, Cov_N]]

(First element is correlation, second covariance)

**Takes time to gather all the metrics, so there will be a slight delay from input to starting output.

Entailment Search

Entailment search allows us to acquire possible set of metrics that entails a pattern. For instance, consider a metric *response_time*:



Given the metric *response_time* above, we can acquire a list of metrics that *likely* entail this pattern. In our analytic engine, this can be done by executing the following command:

```
node ./dist/cli.js entailment_search \
--goal-metric response_time \
--goal-metric-time-begin 05:00_20160917 \
--goal-metric-time-end 12:00_20160917 \
--time-begin 00:00_20160917 \
--time-end 23:00_20160917 \
--iteration-count 10000 \
--out /tmp/temp-result.json \
--dashboard-out /tmp/dashboard.json
```

The *response_time* to look for entailments is placed in **goal-metric** option. The pattern of the goal is captured by setting the **goal-metric-time-begin** and **goal-metric-time-end**, in our case, September 17, 2016 from 5am to 12pm. We then search other metrics from 12am to 11pm at the same day. Increasing the range to search from will increase the model accuracy (less false positives). **iteration-count** is set to 1000. Increasing this will increase accuracy of the ai model.

This command line above will generate a grafana dashboard in /tmp/dashboard.json.

Caveats

- At the moment, this is very resource hungry taking 16gb of ram. In the future, we plan to utilize database so the model is saved in disk and cached in ram.
- Slow due to only utilizing single-core. This problem is the perfect for gpu computing considering there are multiple vectors being multiplied all the time, taking advantage of those huge gigaflops(floating operation per second).