Practical Data Visualization

Visualization can be a valuable tool for analysing and understanding data (and the "reality" that we are trying to measure). It can be much richer than numeric metrics, providing valuable insights. It can also distort, camouflage, or hide information. This talk will provide some real-life examples of how to use and mis-use visualization.
Why do I analyze data?

Debugging performance
   If I know the factor that is limiting performance
   I know where to focus efforts to improve it

Predicting behavior of future workloads
   If I can describe the factors that affect
   performance then I can predict whether a
   workload will perform adequately

examples:
   batch throughput, transaction processing responsiveness,
   gaming, audio recording or playback, desktop interactivity
What am I trying to accomplish?

I am trying to
- remove noise from the data
- find a signal in the data
- solve a mystery
- explore a problem space
- gain insights into how and why a system is behaving
- not lie to myself
What this talk is not

Edward Tufte's books:
- The Visual Display of Quantitative Information
- Envisioning Information
- Visual Explanations

http://www.edwardtufte.com/tufte/books_vdqii

These books are highly recommended. This talk is much more humble....
Data Example

Task Switch Time
Which is Better?

<table>
<thead>
<tr>
<th></th>
<th>min</th>
<th>avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>ts_11</td>
<td>34</td>
<td>69</td>
</tr>
<tr>
<td>ts_12</td>
<td>35</td>
<td>68</td>
</tr>
</tbody>
</table>

(hint: smaller is better)
Which is Better?

<table>
<thead>
<tr>
<th></th>
<th>min</th>
<th>avg</th>
<th>max</th>
</tr>
</thead>
<tbody>
<tr>
<td>ts_11</td>
<td>34</td>
<td>69</td>
<td>147</td>
</tr>
<tr>
<td>ts_12</td>
<td>35</td>
<td>68</td>
<td>159</td>
</tr>
</tbody>
</table>

(conflicting metrics)
Which is Better?

<table>
<thead>
<tr>
<th></th>
<th>min</th>
<th>avg</th>
<th>max</th>
<th>std dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>ts_11</td>
<td>34</td>
<td>69</td>
<td>147</td>
<td>21.9</td>
</tr>
<tr>
<td>ts_12</td>
<td>35</td>
<td>68</td>
<td>159</td>
<td>28.0</td>
</tr>
</tbody>
</table>

(conflicting metrics)
standard deviation - What is it?

**Intuitive description**

A measure of the width of a frequency distribution.

A smaller standard deviation means the distribution of the data points is narrower.

A smaller standard deviation means the distribution of the data points is closer to the mean.
standard deviation – What is it?

If the data was a perfect “normal distribution”, it would be described by the curves on the previous slide.

The starting point for many statistics analysis is “assume a normal distribution”.
Which is Better?

<table>
<thead>
<tr>
<th></th>
<th>min</th>
<th>avg</th>
<th>max</th>
<th>std dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>ts_11</td>
<td>34</td>
<td>69</td>
<td>147</td>
<td>21.9</td>
</tr>
<tr>
<td>ts_12</td>
<td>35</td>
<td>68</td>
<td>159</td>
<td>28.0</td>
</tr>
</tbody>
</table>
red: ts_11  blue: ts_12
(dashed lines are the averages)
Answer
It depends.
Answer

It depends.

Real Answer:

“Average is a meaningless metric”
“Standard Deviation is a meaningless metric”
Answer

Trick question. The two sets of data, ts_11 and ts_12, are two different runs of the same test.

Real Answer:

The graph explains difference between the two test runs better than the raw metrics.
Which is Better?

<table>
<thead>
<tr>
<th></th>
<th>min</th>
<th>avg</th>
<th>max</th>
</tr>
</thead>
<tbody>
<tr>
<td>ts_15</td>
<td>22</td>
<td>40</td>
<td>94</td>
</tr>
<tr>
<td>ts_18</td>
<td>28</td>
<td>36</td>
<td>80</td>
</tr>
</tbody>
</table>
Which is Better?

<table>
<thead>
<tr>
<th></th>
<th>min</th>
<th>avg</th>
<th>max</th>
<th>std dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>ts_15</td>
<td>22</td>
<td>40</td>
<td>94</td>
<td>1.2</td>
</tr>
<tr>
<td>ts_18</td>
<td>28</td>
<td>36</td>
<td>80</td>
<td>9.6</td>
</tr>
</tbody>
</table>
red: ts_15  blue: ts_18
(dashed lines are the averages)
Answer

It depends.

What is more important for your use case?

Throughput -- average
Consistency / determinancy -- std dev, maximum
Worst case -- maximum
Answer

Real Answer:

The graph explains difference between the two test runs better than the raw metrics.
“Average is a meaningless metric”
“Standard Deviation is a meaningless metric”
Average is a meaningless metric

If the different data data sets have different distributions

For hard real-time metrics, average is always meaningless (maximum is critical)
Standard Deviation is a meaningless metric

If the data is not a normal distribution
Meaningless Metrics

But Average and Standard Deviation can still be used as flags to get your attention or provide insights.
ASCII graphs can also be useful

Not sexy, but do not underestimate their power
Migration Algorithm 1

# producer cpu map:
#
# 0 0101010101010101010101010101010101010101010101010101010101010101010101
# 70 0101010101010101010101010101010101010101010101010101010101010101010101
# 140 0101010101010101010101010101010101010101010101010101010101010101010101
# 210 0101010101010101010101010101010101010101010101010101010101010101010101
# 280 0101010101010101010101010101010101010101010101010101010101010101010101
# 350 0101010101010101010101010101010101010101010101010101010101010101010101
# 420 0101010101010101010101010101010101010101010101010101010101010101010101
# 490 0101010101010101010101010101010101010101010101010101010101010101010101
# 560 0101010101010101010101010101010101010101010101010101010101010101010101
# 630 0101010101010101010101010101010101010101010101010101010101010101010101
# 700 0101010101010101010101010101010101010101010101010101010101010101010101
#
# consumer cpu map:
#
# 0 1010101010101010101010101010101010101010101010101010101010101010101010
# 70 1010101010101010101010101010101010101010101010101010101010101010101010
# 140 1010101010101010101010101010101010101010101010101010101010101010101010
# 210 1010101010101010101010101010101010101010101010101010101010101010101010
# 280 1010101010101010101010101010101010101010101010101010101010101010101010
# 350 1010101010101010101010101010101010101010101010101010101010101010101010
# 420 1010101010101010101010101010101010101010101010101010101010101010101010
# 490 1010101010101010101010101010101010101010101010101010101010101010101010
# 560 1010101010101010101010101010101010101010101010101010101010101010101010
# 630 1010101010101010101010101010101010101010101010101010101010101010101010
# 700 1010101010101010101010101010101010101010101010101010101010101010101010
Excessive migration is usually not good.
Migration Algorithm 2

producer and consumer always on same cpu, instead of always on the other cpu

```
#             -- producer ------------
# consumer    cpu 0     cpu 1
# --------    ----------  ----------
#  cpu 0        5168       0
#  cpu 1           0       4832
```
Migration Algorithm 2

producer and consumer always on same cpu, instead of always on the other cpu

# -- producer ------------
# consumer      cpu 0       cpu 1
# --------     ----------  ----------
#  cpu 0             5168           0
#  cpu 1                0        4832

How often does migration occur?
One migration per test run?
One migration per message?
Migration Algorithm 2

# producer cpu map:
#
#       0 0010101010000000000100111111111111111111111111111111111111111111111111
#      70 1111111111111111111111111111111111111111111111111111111111111111111111
#     140 0000000000000111111111111111111111111111111111111111111111111111111111
#     210 1111111111111111111111111111111111111111111111111111111111111111111111
#     280 1111111111111111111111111111111111111111111111111111111111111111111111
#     350 0000000000000000000000000000000000000000000000000000000000000000000000
#     420 0011111111111111111111111111111111111111111111111111111111111111111111
#     490 1111111111111111111111111111111111111111111111111111111111111111111111
#     560 0000000000000000000000000000000000000000000000000000000000010011111111
#     630 1111111111111111111111111111111111111111111111111111111111111111111111
#     700 1111111111111111111111111111111111111111111111111111111111111111111111
#
# consumer cpu map:
#
#       0 0010101010000000000100111111111111111111111111111111111111111111111111
#      70 1111111111111111111111111111111111111111111111111111111111111111111111
#     140 0000000000000111111111111111111111111111111111111111111111111111111111
#     210 1111111111111111111111111111111111111111111111111111111111111111111111
#     280 1111111111111111111111111111111111111111111111111111111111111111111111
#     350 0000000000000000000000000000000000000000000000000000000000000000000000
#     420 0011111111111111111111111111111111111111111111111111111111111111111111
#     490 1111111111111111111111111111111111111111111111111111111111111111111111
#     560 0000000000000000000000000000000000000000000000000000000000010011111111
#     630 1111111111111111111111111111111111111111111111111111111111111111111111
#     700 1111111111111111111111111111111111111111111111111111111111111111111111
Migration Algorithm 2

# producer cpu map:
#
#      0 001010101010000000000010011111111111111111
#     70 11111111111111111111111111111111111111
#    140 00000000000001111111111111111111111111
#
# consumer cpu map:
#
#      0 001010101010000000000010011111111111111111
#     70 11111111111111111111111111111111111111
#    140 00000000000001111111111111111111111111

Occasional migration, but producer and consumer always on same cpu
Lines vs Points

Which presentation provides a more informative graph?
blue: insertions
blue: insertions
Lines vs. Points

Lines can emphasize trends, changes in direction
Lines can hide detail

Points can expose detail
Data Example

Scheduler task migrations per second
Sampling Period

Number of events sampled per data point graphed
trace_05
sample duration: 00100 msec

migrations / second

elapsed time (seconds)
trace_05
sample duration: 01000 msec
trace_05
sample duration: 02000 msec

migrations / second

elapsed time (seconds)
trace_05
sample duration: 06000 msec

migrations / second

elapsed time (seconds)
trace_05
sample duration: 10000 msec
trace_05
sample duration: 20000 msec

migrations / second

elapsed time (seconds)
trace_05
sample duration: 40000 msec

[Graph showing the number of migrations per second over time, with a peak around the 100-second mark.]
Lines vs Points

another example
trace_05
sample duration: 00100 msec
trace_05
sample duration: 00333 msec
trace_05
sample duration: 02000 msec
trace_05
sample duration: 10000 msec
trace_05
sample duration: 20000 msec

migrations / second

elapsed time (seconds)
trace_05
sample duration: 40000 msec
Rescale

Change scale to reveal detail
trace_05
sample duration: 00100 msec
trace_05
sample duration: 00100 msec
trace_05
sample duration: 00333 msec

migrations / second vs elapsed time (seconds)
trace_05
sample duration: 01000 msec

![Graph showing migrations per second over elapsed time (seconds).]
trace_05
sample duration: 02000 msec

[Diagram showing migrations per second over elapsed time in seconds]
trace_05
sample duration: 06000 msec

The graph shows the number of migrations per second over time. The y-axis represents the number of migrations per second, while the x-axis represents the elapsed time in seconds. The graph starts with a slight increase in the number of migrations, followed by a steady state, and then a sharp decrease towards the end of the sample duration.
trace_05
sample duration: 10000 msec
trace_05
sample duration: 20000 msec
trace_05
sample duration: 40000 msec

![Graph showing the number of migrations per second over elapsed time. The graph shows a downward trend indicating a decrease in migrations as the elapsed time increases. The y-axis represents the number of migrations per second, ranging from 480 to 320. The x-axis represents the elapsed time in seconds, ranging from 0 to 80. The graph has a blue line indicating the trend.]
Lines vs Points

again...
trace_05
sample duration: 00100 msec

migrations / second

elapsed time (seconds)
trace_05
sample duration: 00100 msec
trace_05
sample duration: 00333 msec
trace_05
sample duration: 01000 msec

migrations / second

elapsed time (seconds)
trace_05
sample duration: 10000 msec
Transformation between domains

May reveal different information

In this example:

time domain to histogram transformation shows significance of bi-modal nature of data more clearly....
trace_05
sample duration: 01000 msec
Scatterplot

Graph two metrics for each sample

Requires logging of each sample instead of updating counter(s) for each sample.
  More detailed information
  More overhead in data collection
Rescale

Individual samples: visibility vs details

data: cost of do_local_timer()
trace_irqs_off_106_f-34_c-0_1_S-0
cpu 0
Rescale

Change scale to expose detail (2)
trace_irqs_off_106_f-34_c-0_1_S-0

cpu 0
Additional Dimensions

Reveal additional information (intensity)

3-d (shown)
2-d with color gradient (not shown)
Finding trees in the forest

(1) Subset the data:
   - all samples
   - samples filtered by scheduling policy of interrupted task
   - segment multi-modal distribution into several (nearly) normal distributions
Finding trees in the forest

(2) Transformation between domains and another way of showing intensity

Puts focus on two different concepts:
- # instructions vs # i-cache stall cycles
- frequency of # i-cache stall cycles
trace_irqs_off_121_f-34_c-0_1_S-1
cpu 0 check_preempt_curr_idle
Comparing multiple tests

Metric: maximum IRQ disabled time

1 types of test for each kernel configuration

4 variants of kernel configuration
  data for cpu 0
  data for cpu 1
matplotlib box plot

box is:  25%, 50%, 75%
        star inside box is the average

whisker end is most extreme value within
1.5 * (75% - 25%), each outlier would also be shown

Numbers above each box are:
  max
  median
  average
  min
matplotlib box plot

An example plot showing outliers is available at:

http://matplotlib.org/pyplots/boxplot_demo_06.hires.png
Comparing multiple dimensions

Metric: maximum IRQ disabled time

11 types of test for each kernel configuration

15 variants of kernel configuration
max IRQs disabled time
11 test cases
max IRQs disabled time
11 test cases

config_1  config_2  config_3  config_4  config_5

config_6  config_7  config_8  config_9  config_10

config_11 config_12 config_13 config_14 config_15
Final Thoughts

The simple answer is sometimes correct only through chance, dig deeper

Be creative in visualization

Visualization can hide or expose information

Ensure there is a physical meaning underlying the metrics and the visualization of the metrics
THE END

Thank you for your attention...
Questions?
How to get a copy of the slides

1) leave a business card with me

2) frank.rowand@am.sony.com