Visualization on Big Data / Basic of profiling

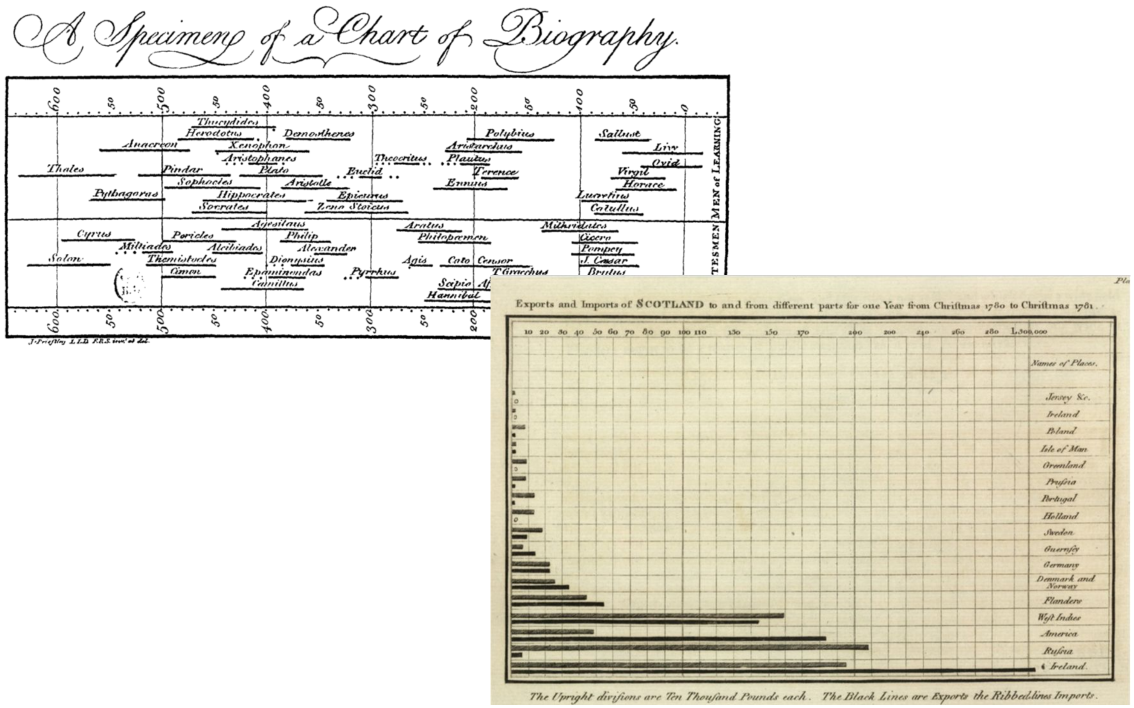
Lecture 7
November 22nd, 2017

Jonghyun Bae (jonghbae@snu.ac.kr)
Computer Science and Engineering
Seoul National University

***Slide credits**: Ji Lee (데이터 분석 시각화 분석), Nathan Yau (How to Spot Visualization Lies)*

Begin of visualization (1)

- **William Playfair in 1786**
  - Founder of graphical methods of statistics
  - Bar charts, graphics



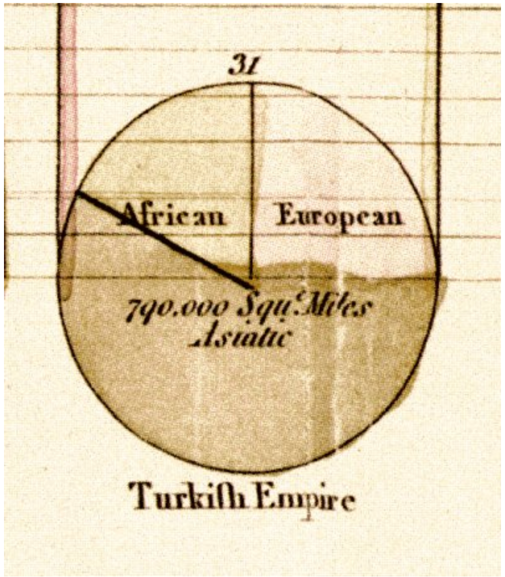


Image from https://en.wikipedia.org/wiki/William_Playfair

Begin of visualization (2)

- **Combination of various fields**
  - Computer engineering, Statistics, Graphic design, Human-Computer Interaction

- **It feels like we're all suffering from information overload of data glut. And the good news is there might be an easy solution to that, and that's using out eyes more.**
  - David McCandless (at TEDGlobal 2010)

Outline

- **Data visualization**
- Characteristics of data and graph
- Visualization on big data
- How to visualize with Spark
- Basic of profiling
- Types of profiler

Purpose of visualization

- **The representation and presentation of data to facilitate understanding**
  - Save a time
  - Have a clear purpose
  - Include only the relevant content
  - Encode data/information appropriately

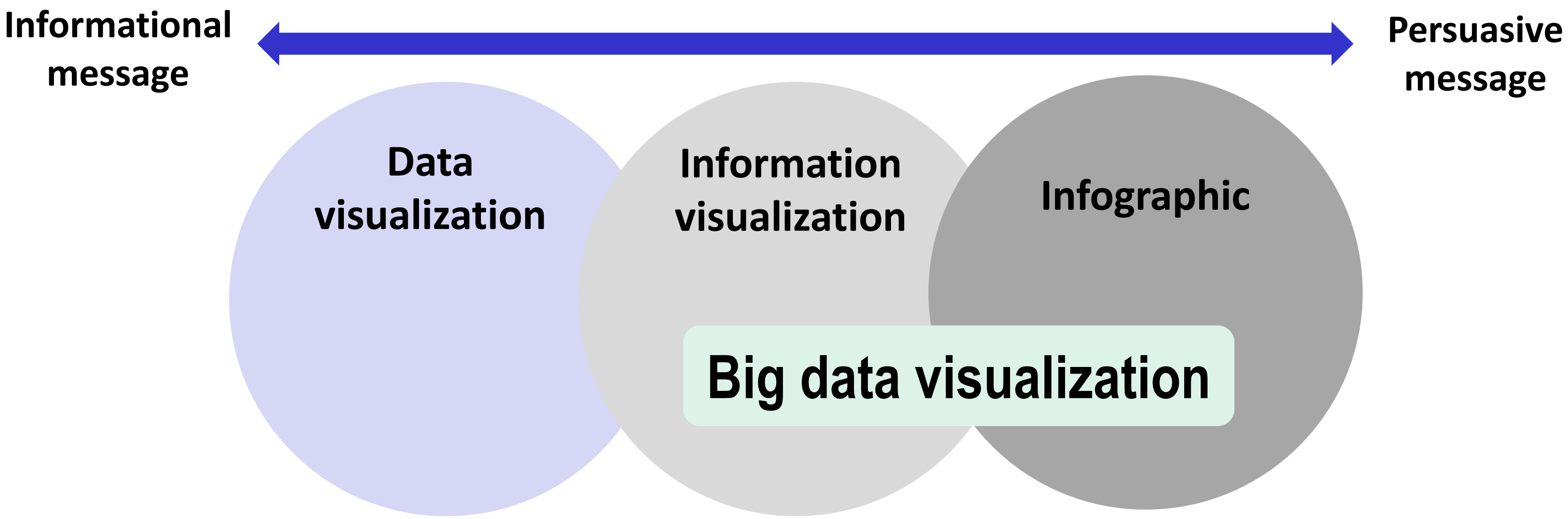
Classification of visualization (1)

- **Data visualization**
  - Research area of visual representation of data
  - To communicate information proactively and effectively using graphic meanings

- **Information visualization**
  - To visualize large quantities of quantitative information
  - Intuitively deliver abstract information for users to view, explore, and understand

Classification of visualization (2)

- **Infographic**
  - A graphical message that represents important information in a single graphical representation that makes it easy for people viewing it to understand the information.
  - Used in symbols, maps, technical documents, etc. that need to explain complex information quickly and clearly



Principles

- **Trustworthy**
- **Accessible**
  - Understanding
- **Elegant**
  - Eliminate arbitrary
  - Thoroughness
  - Style
  - Decoration (additive, not negative)

Visualization methodology

- Ben Fry’s seven-steps

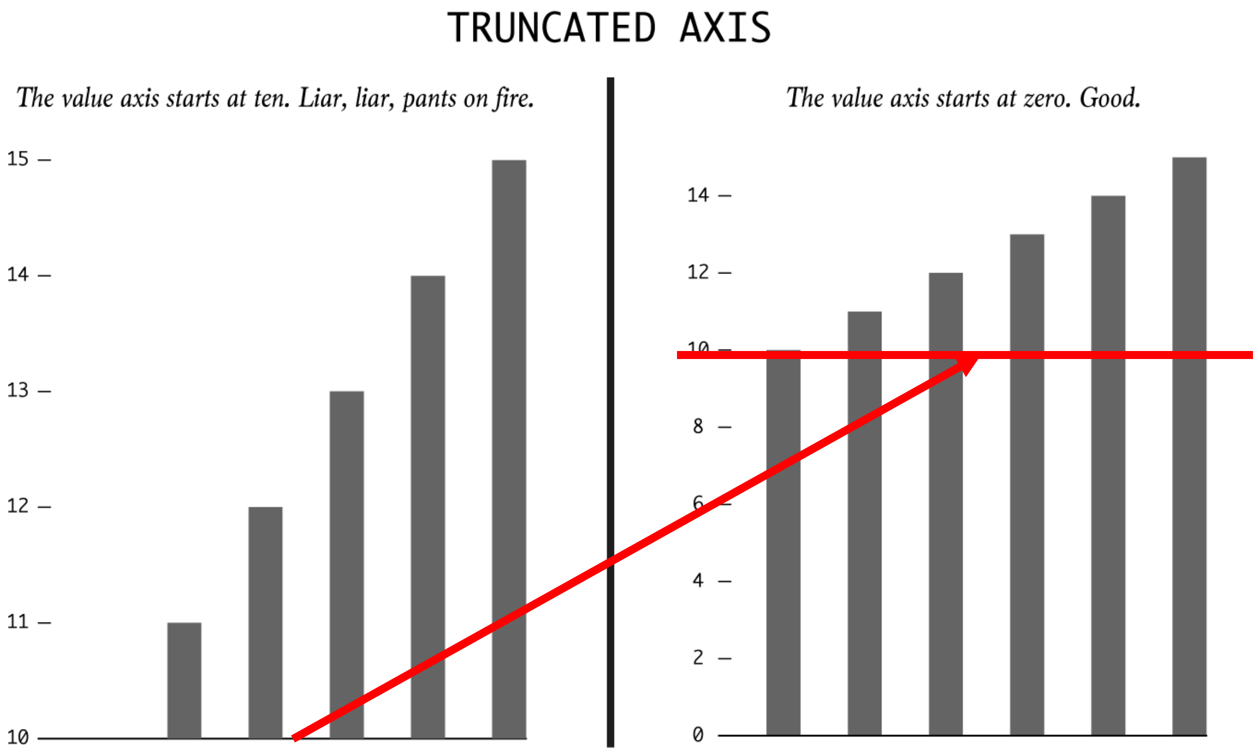
Stage	Description
Acquire	Obtain the data (file, disk, over network)
Parse	Provide some structure for the data’s meaning, and order them into categories
Filter	Remove all but the data of interest
Mine	Apply methods from statistics or data mining as a way to discern patterns or place the data
Represent	Choose a basic visual model, such as a bar graph, list, tree, etc.
Refine	Improve the basic representation to be clearer and more visually engaged
Interact	Add methods for manipulating or controlling what features are visible

Visualization tools

	Description
General purpose	Excel, CVS / JSON, Google chart API, D3 (Data-Driven Documents), Visual.ly
Interactive GUI control	Crossfilter, Tangle
Mapping	Modest Maps, Leaflet, Polymaps, OpenLayers, Kartograph, CartoDB
Expert	Processing, NodeBo, R, python, Weka, Gephi

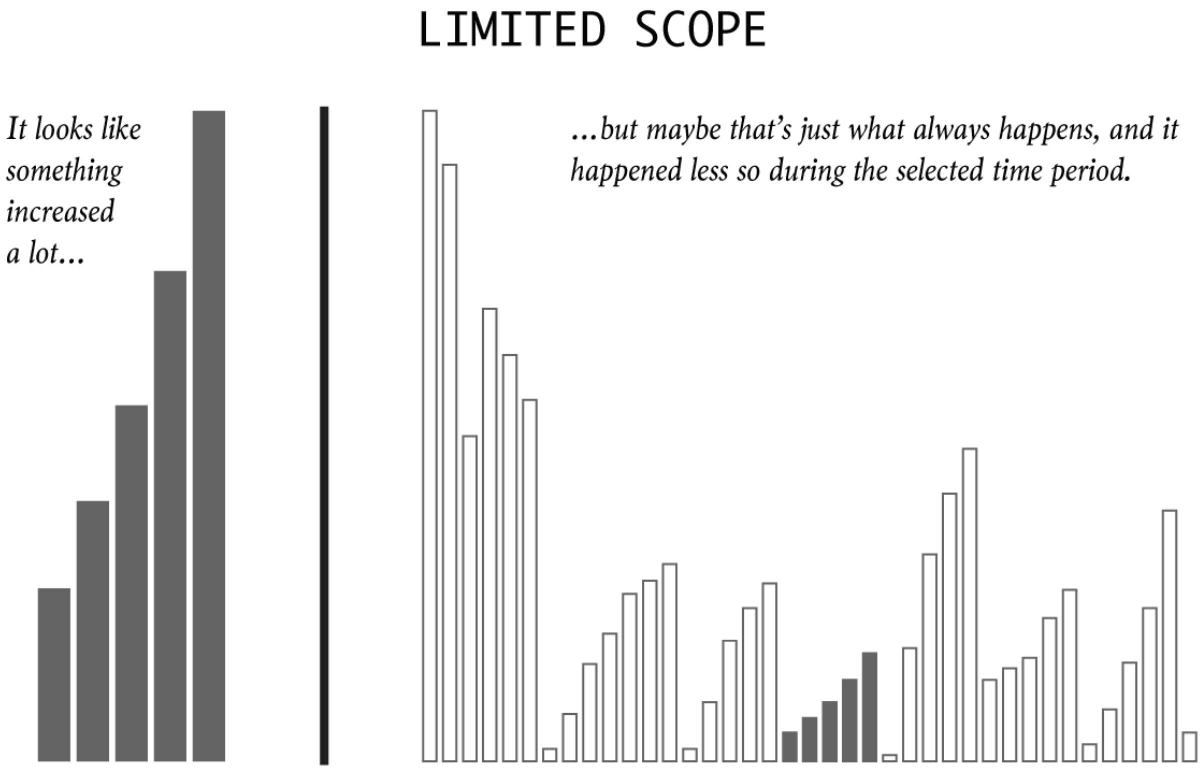
Pitfalls (1)

- **Truncated axis**
  - Make the length shorter using the same data by truncating the value axis



Pitfalls (2)

- **Limited scope**
  - Easy to cherry-pick dates and timeframes to fit a specific narrative

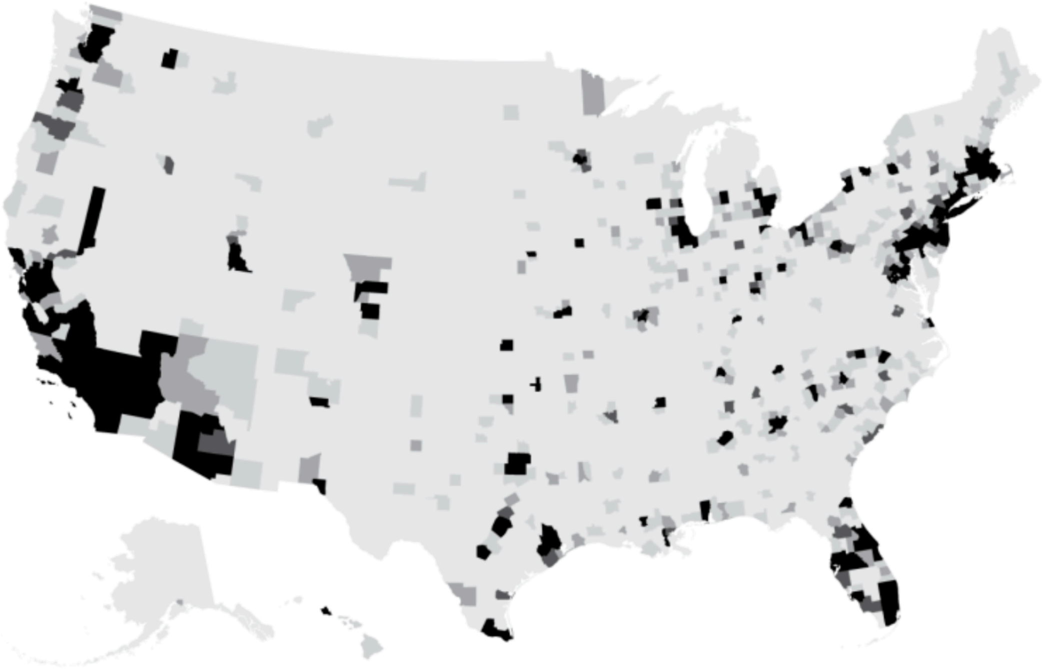


Pitfalls (3)

- **Seeing only in absolutes**
  - Use relative (or normalized) data in some case

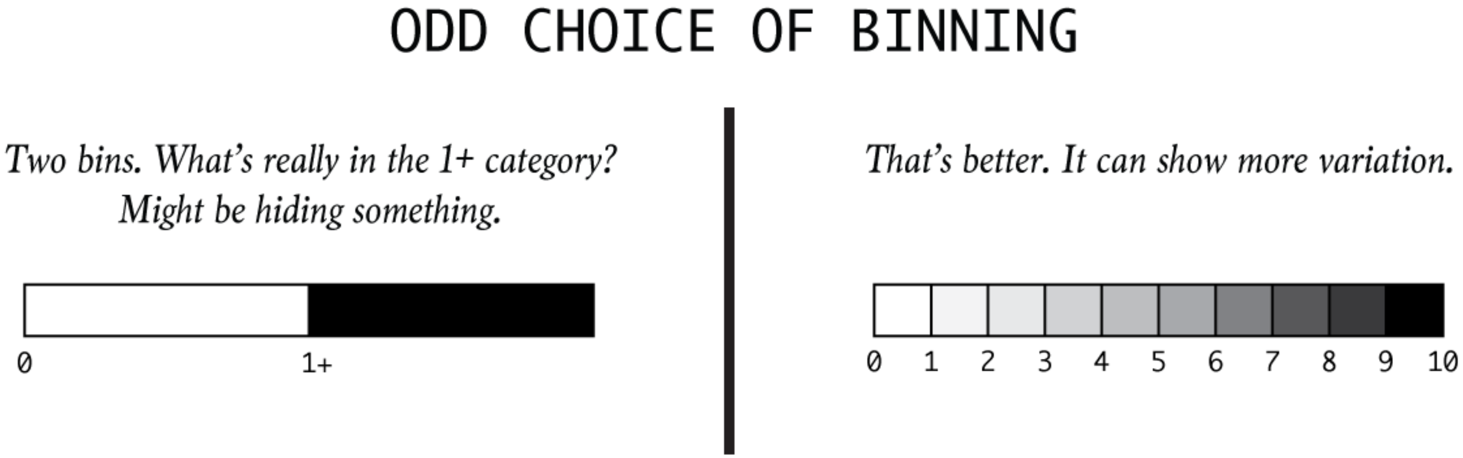
SEEING ONLY IN ABSOLUTES

This is just population. When comparing across places, categories, or groups, you must compare fairly and consider relative values.



Pitfalls (4)

- **Odd choice of binning**
  - Complexity is often what makes things worth looking at.
  - Do not oversimplification


ODD CHOICE OF BINNING
Two bins. What's really in the 1+ category? Might be hiding something.
0
1+
That's better. It can show more variation.
0
1
2
3
4
5
6
7
8
9
10

Outline

- Data visualization
- **Characteristics of data and graph**
- Visualization on big data
- How to visualize with Spark
- Basic of profiling
- Types of profiler

Data types

- **Data types**
  - Static, dynamic
- **Dataset types**
  - Tables, networks, fields, geometry, trees
- **Attribute types**
  - Categorical, ordered marking

Category of graphs (1)

- **Time-series**
  - A single variable is captured over a period of time
- **Ranking**
  - Categorical subdivisions are ranked in ascending or descending order
- **Part-to-whole**
  - Categorical subdivisions are measured as a ratio to the whole

Category of graphs (2)

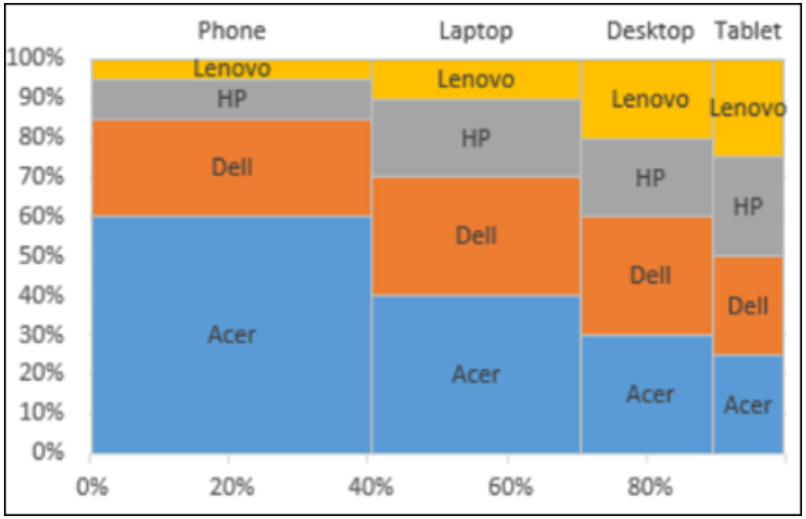
- **Frequency distribution**
  - Shows the number of observations of a particular variable for given interval
- **Correlation**
  - Comparison between observations represented by two variables to determine if they tend to move in the same of opposite directions
- **Nominal comparison**
  - Comparing categorical subdivisions in no particular order

Method for visualization

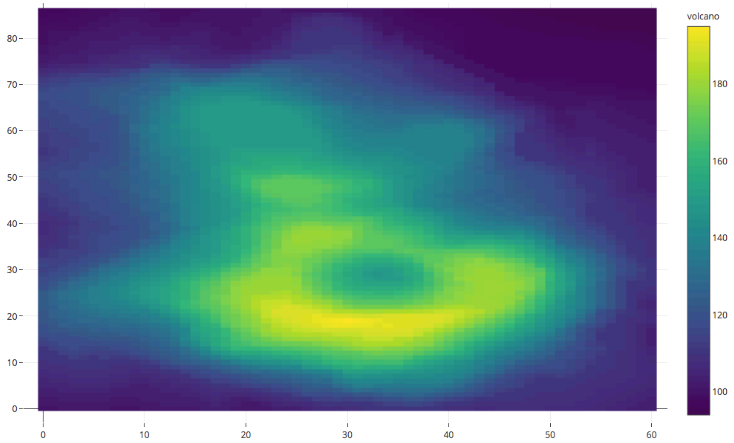
Category	Time series	Line chart
	Ranking	Bar chart (with ordering)
	Part-to-whole	Donut chart, Pie chart, **Marimekko chart**, Stacked bar chart, Sunburst diagram, **Treemap**
	Frequency distribution	Histogram, Pie chart, Stem-and-leaf plot, **Heatmap**
	Correlation	Scatter plot
	Nominal comparison	Dot plot

Graph types (1)

- **Marimekko chart**
  - Encode two quantitative variables: one using the height and one using the width of the bars

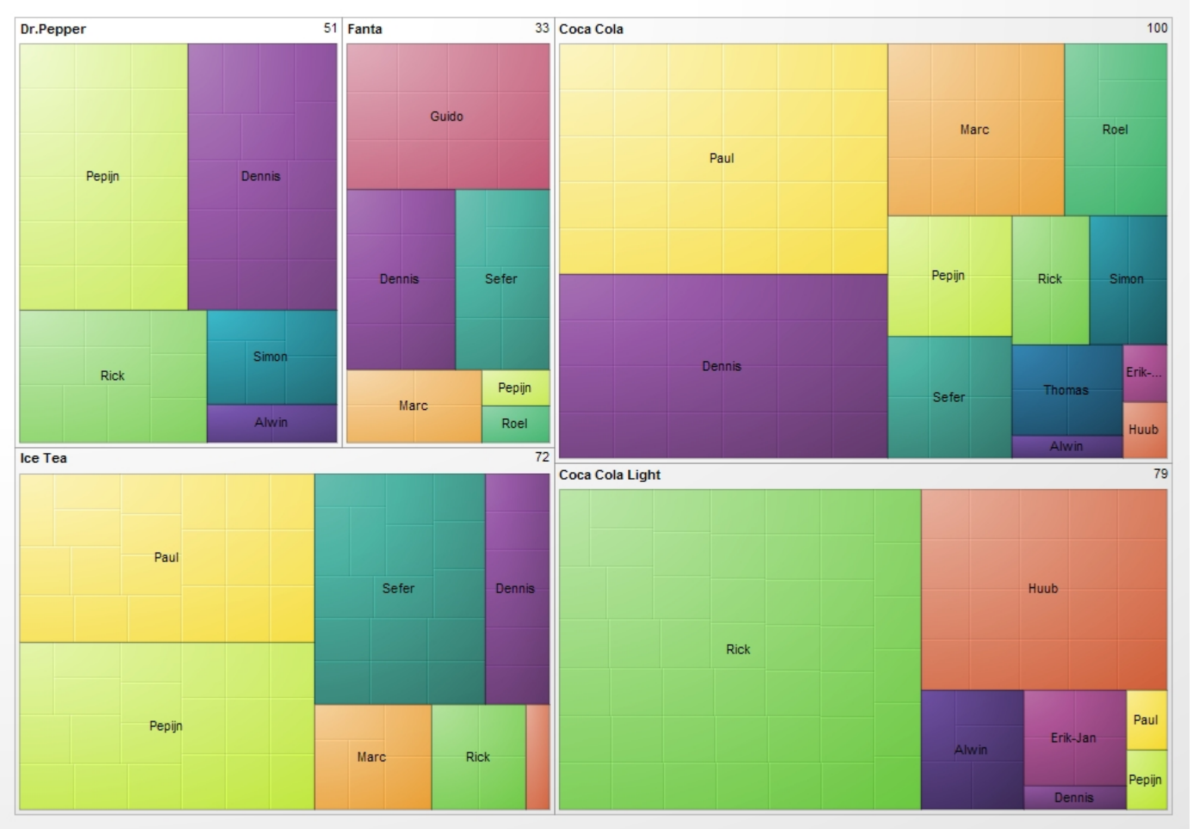


- **Heat map**
  - Individual values contained in a matrix are represented as colors



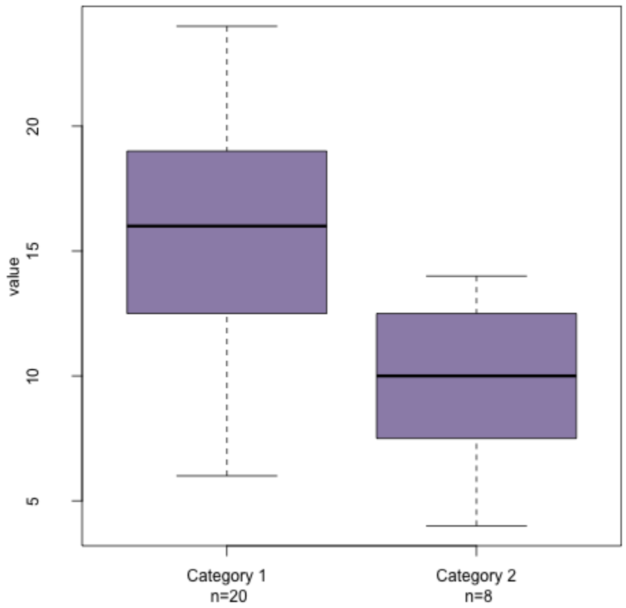
Graph types (2)

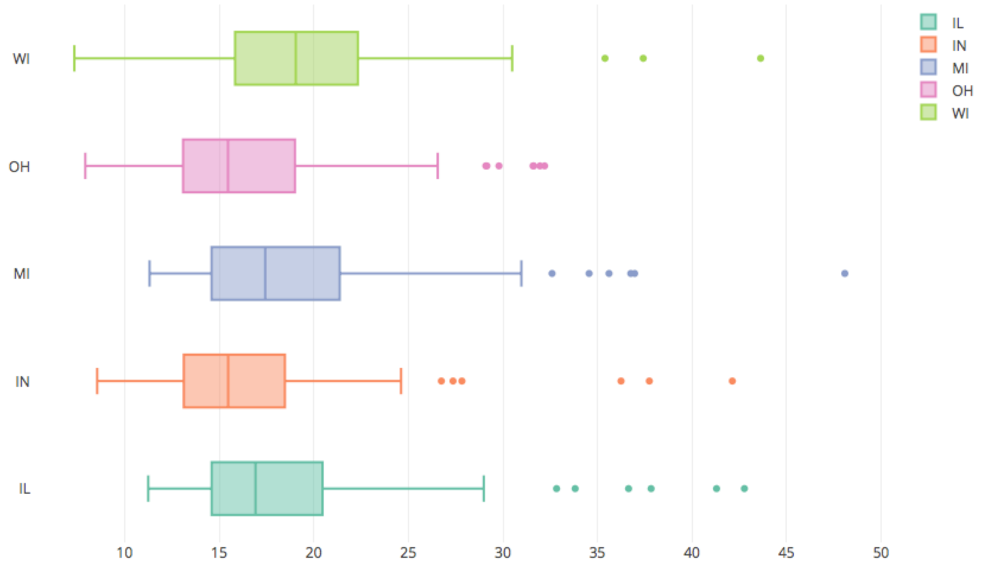
- **Treemapping**
  - Display hierarchical data as a set of nested rectangles
  - Branch of the tree is given a rectangle with smaller rectangles representing sub-branches



Graph types (3)

- **Box and whisker**
  - Groups of numerical data through their quartiles
  - Variability outside the upper and lower quartiles
  - Outliers may be plotted as individual points





Outline

- Data visualization
- Characteristics of data and graph
- **Visualization on big data**
- How to visualize with Spark
- Basic of profiling
- Types of profiler

Purpose of big data visualization

- **Analyzes phenomena, patterns, structures, changes, and correlations that appear constantly to identify future problems and find problems**

- **Can be used to collect two or more pieces of information as meaningful or messageful information**

- **Visualize and deliver big data analysis results for easy understanding**
  - Information visualization

Difficulty of big data visualization

- **Handling large volumes**
  - Sampling, regression and summary
- **Hard to real time computation**
  - Streaming technique
- **Different audience and data**

Efficient data reduction (1)

- **Sampling**
  - Selection of a subset of individuals from within a statistical population to estimate characteristics
  - Clustering whole dataset and get subset of each cluster

- **Regression**
  - Estimating the relationships among variables
  - Widely used for prediction and forecasting

Efficient data reduction (2)

- **Summary**
  - Summarize a set of observations, in order to communicate the largest amount of information as simply as possible
  - Standard deviation, range, interquartile range, mean absolute difference, etc.

Streaming computation

- **Using data which is generated continuously by thousands of data sources**
  - Mobile or web applications, ecommerce purchases, inform from social networks

- **Difference between batch processing**
  - Queries or processing over data within a rolling time window, or on just the most recent data record
  - Individual records or micro batches consisting of a few records
  - Requires latency in the order of seconds or milliseconds

Efficient graph types for big data (1)

- **Correlation matrix**

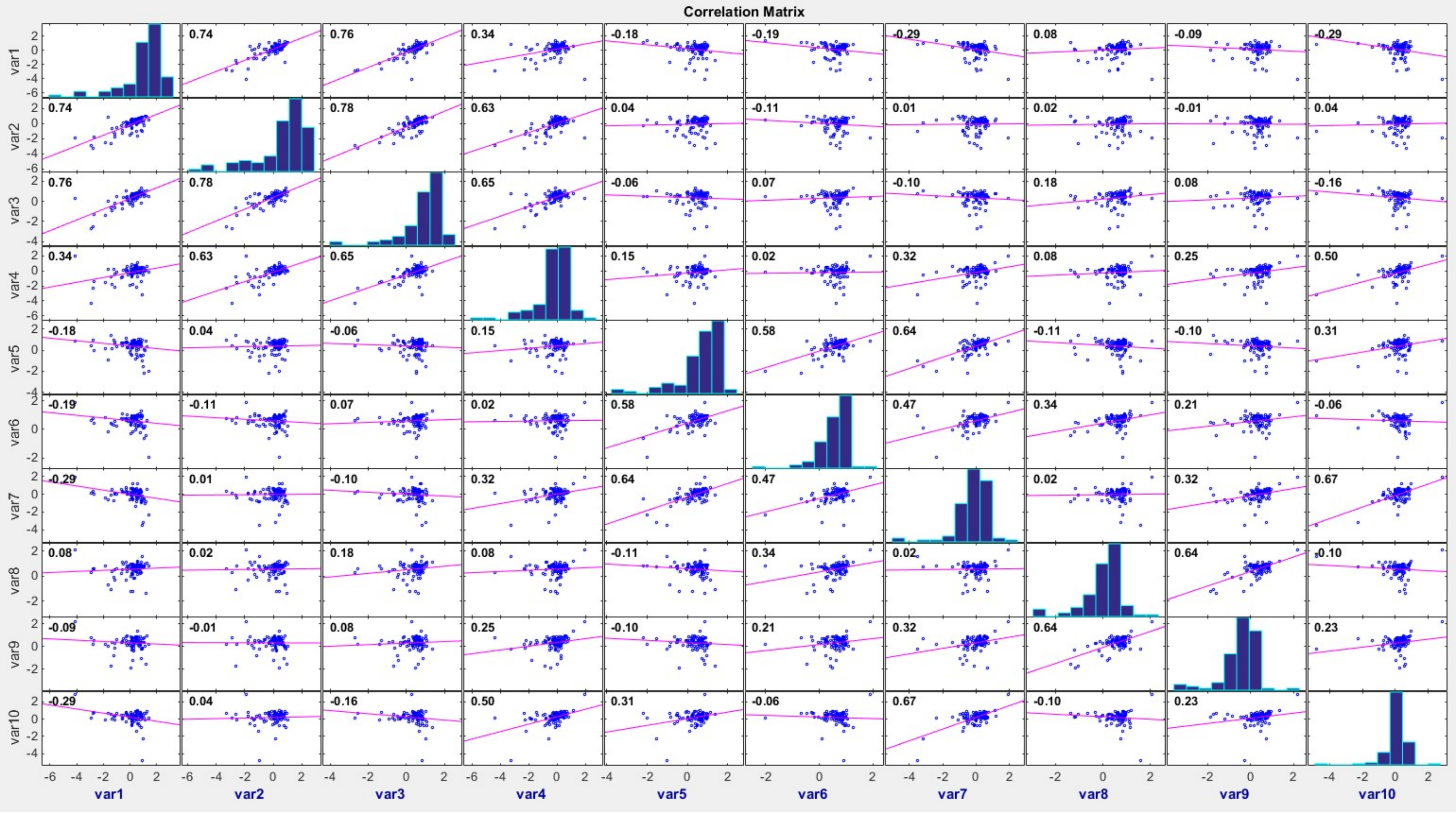
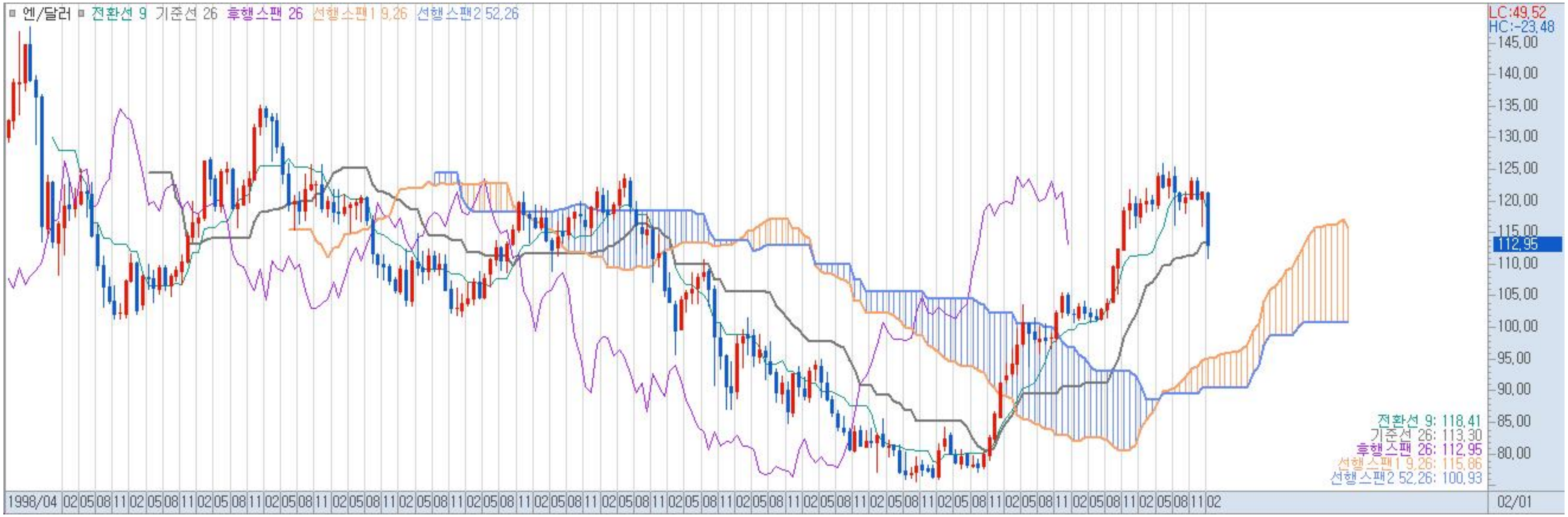


Image from https://kr.mathworks.com/matlabcentral/answers/248227-plot-only-upper-part-half-of-the-correlation-matrix?requestedDomain=www.mathworks.com

Efficient graph types for big data (2)

- Time-series (Forecasting)



Time-series

Forecasting

Image from http://thinkaboutcapital.blogspot.kr/2016/02/blog-post_21.html

Outline

- Data visualization
- Characteristics of data and graph
- Visualization on big data
- **How to visualize with Spark**
- Basic of profiling
- Types of profiler

Limitation of Spark visualization

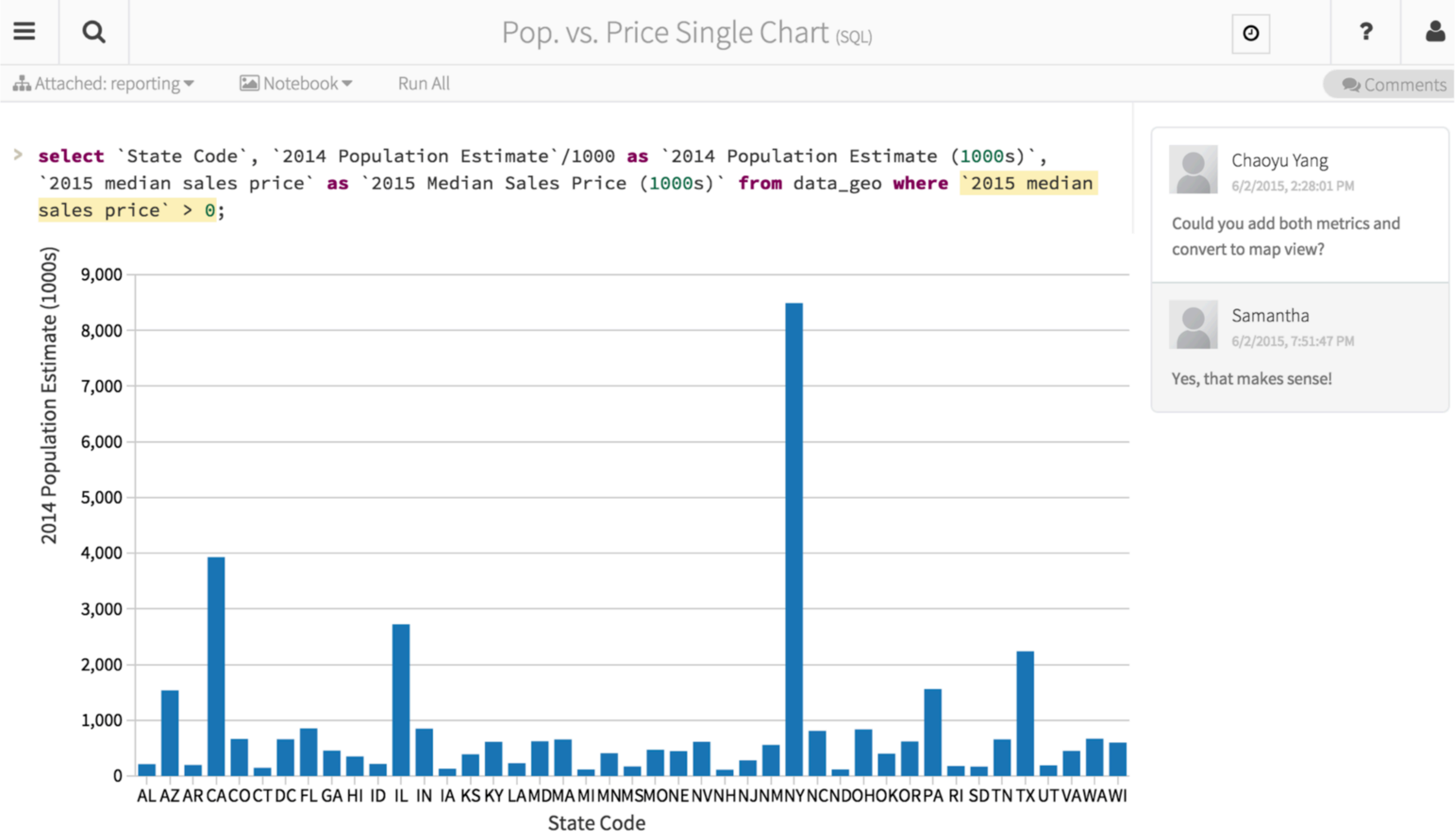
- Currently apache spark does not support its own visualization tool
- It is necessary to convert Spark's operation result to another graphic tool
- Or use a tool that automatically converts and visualizes data

Databricks Unified Analytics Platform (1)

- Started by developers of Apache Spark
- Run on AWS for cloud infrastructure
- Optimizes I/O performance and fully-managed cloud platform

Databricks Unified Analytics Platform (2)

- **https://databricks.com**



Pop. vs. Price Single Chart (SQL)
Attached: reporting
Notebook
Run All
Comments
2014 Population Estimate (1000s)
9,000
8,000
7,000
6,000
5,000
4,000
3,000
2,000
1,000
0
AL AZ AR CA CO CT DC FL GA HI ID IL IN IA KS KY LA MD MA MI MN MS MO NE NV NH NJ NM NY NC ND OH OK OR PA RI SD TN TX UT VA WA WI
State Code

```
> select `State Code`, `2014 Population Estimate`/1000 as `2014 Population Estimate (1000s)`,
  `2015 median sales price` as `2015 Median Sales Price (1000s)` from data_geo where `2015 median
  sales price` > 0;
```

Chaoyu Yang
6/2/2015, 2:28:01 PM

Could you add both metrics and convert to map view?

Samantha
6/2/2015, 7:51:47 PM

Yes, that makes sense!

Apache Zeppelin (1)

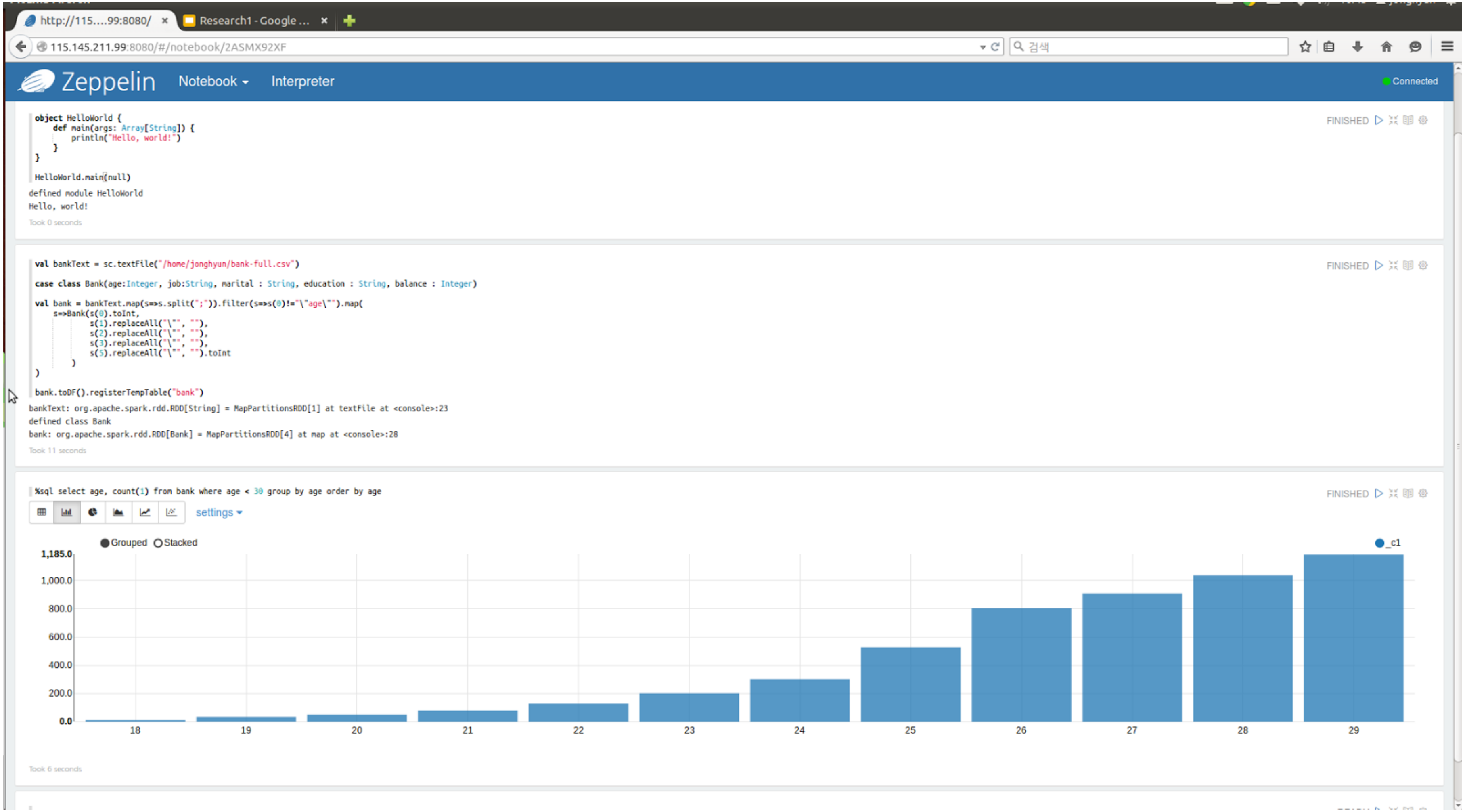
- **Web-based notebook that enables data-driven, interactive data analytics**

- **Multiple language backend**
  - Interpreter concept to be plugged into Zeppelin
  - python, R, PostgreSQL, cassandra, Google BigQuery

- **Multi-user support with LDAP**

Apache Zeppelin (2)

- **http://zeppelin.apache.org**



```
object HelloWorld {
    def main(args: Array[String]) {
        println("Hello, world!")
    }
}

HelloWorld.main(null)
defined module HelloWorld
Hello, world!
Took 0 seconds
```

```
val bankText = sc.textFile("/home/jonghyun/bank-full.csv")

case class Bank(age:Integer, job:String, marital : String, education : String, balance : Integer)

val bank = bankText.map(s=>s.split(";")).filter(s=>s(0)!="\"age\"").map(
    s=>Bank(s(0).toInt,
            s(1).replaceAll("\"", ""),
            s(2).replaceAll("\"", ""),
            s(3).replaceAll("\"", ""),
            s(5).replaceAll("\"", "").toInt
        )
)

bank.toDF().registerTempTable("bank")
bankText: org.apache.spark.rdd.RDD[String] = MapPartitionsRDD[1] at textFile at <console>:23
defined class Bank
bank: org.apache.spark.rdd.RDD[Bank] = MapPartitionsRDD[4] at map at <console>:28
Took 11 seconds
```

```
%sql select age, count(1) from bank where age < 30 group by age order by age
```

Took 6 seconds

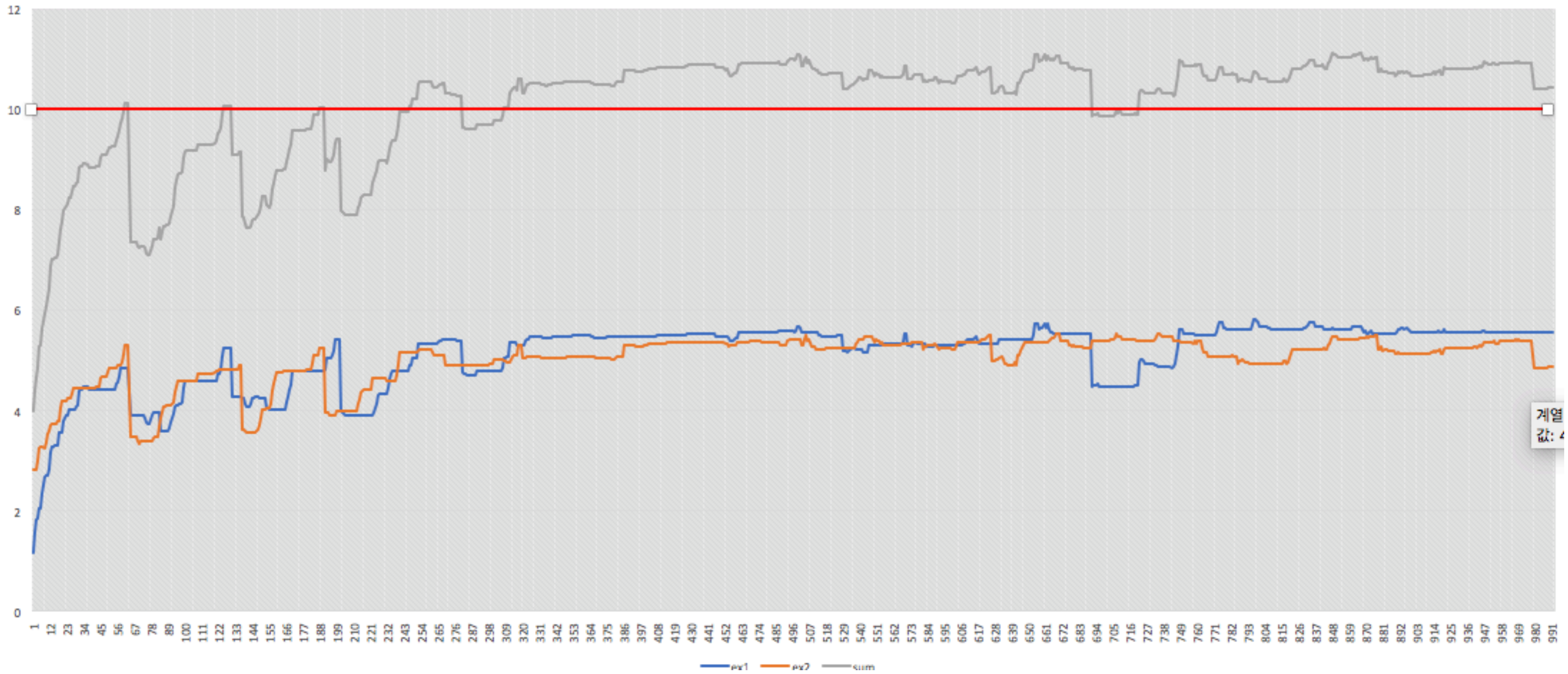
Outline

- Data visualization
- Characteristics of data and graph
- Visualization on big data
- How to visualize with Spark
- **Basic of profiling**
- Types of profiler

Profiling

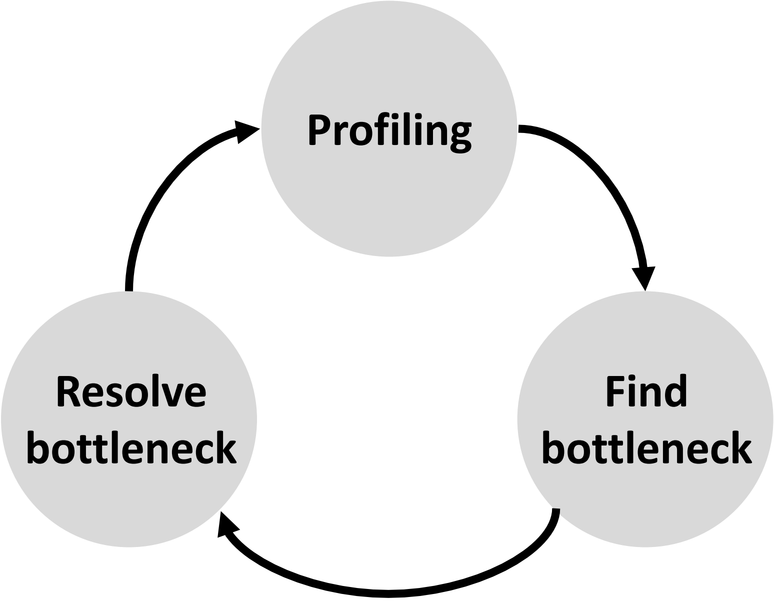
- **Form of dynamic program analysis that measures**
  - Space, time complexity, frequency and duration of function calls

- **Serve to aid program optimization**



Importance of profiling

- **Fine performance bottleneck**
  - Amdahl’s law: After resolving one performance bottleneck, the performance bottleneck reappears in the unresolved area



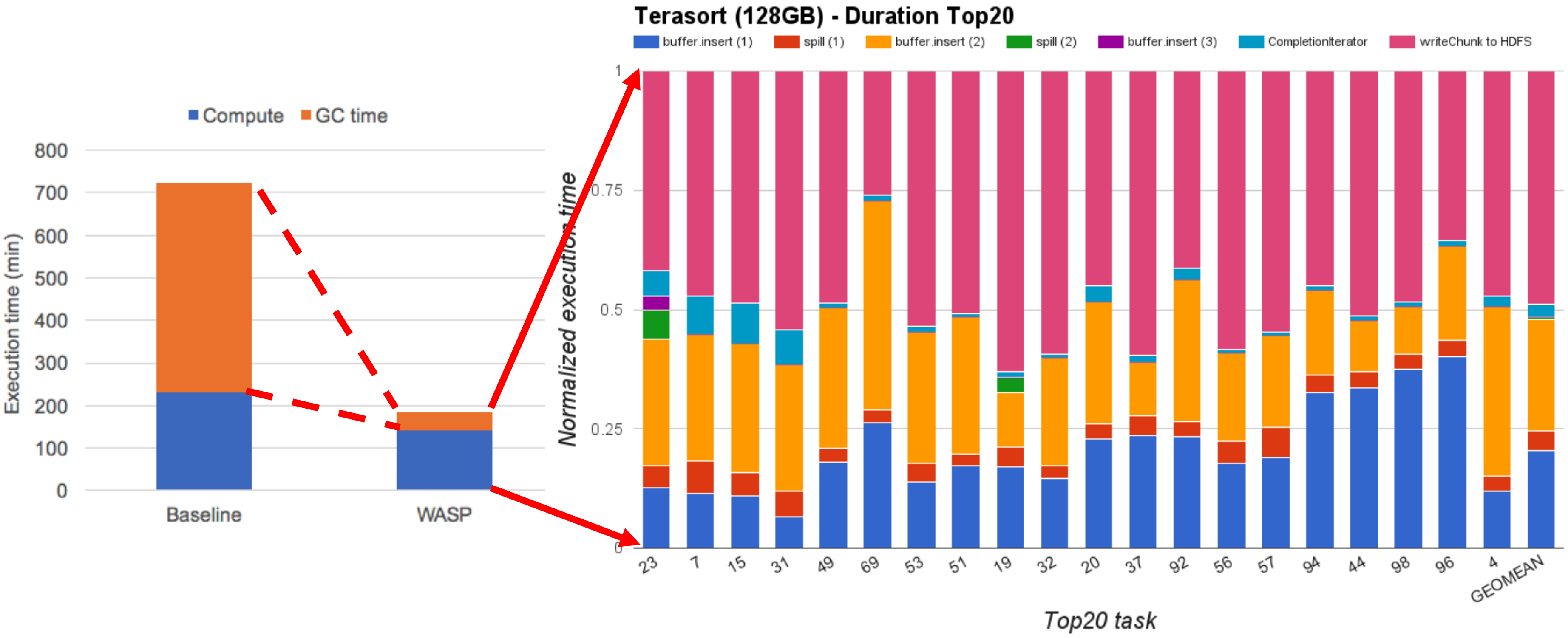
- **So that is the reason why we always profiling our programs**

Profiling factor

	Target	Factor	Index
Hardware	CPU	Clock, cores	Usage, idle time (%)
	Memory	Total size	Space usage (%)
	Storage (I/O)	I/O latency, throughput	I/O wait
Software	O/S	Type, version	Swapping, paging, lock
	Middleware	Instances, configuration	Resource usage
	Application	Algorithm, data structure	Execution time

Example of profiling

- **GC was performance bottleneck**
  - Then what is next performance bottleneck? (maybe pink region)



Outline

- Data visualization
- Characteristics of data and graph
- Visualization on big data
- How to visualize with Spark
- Basic of profiling
- **Types of profiler**

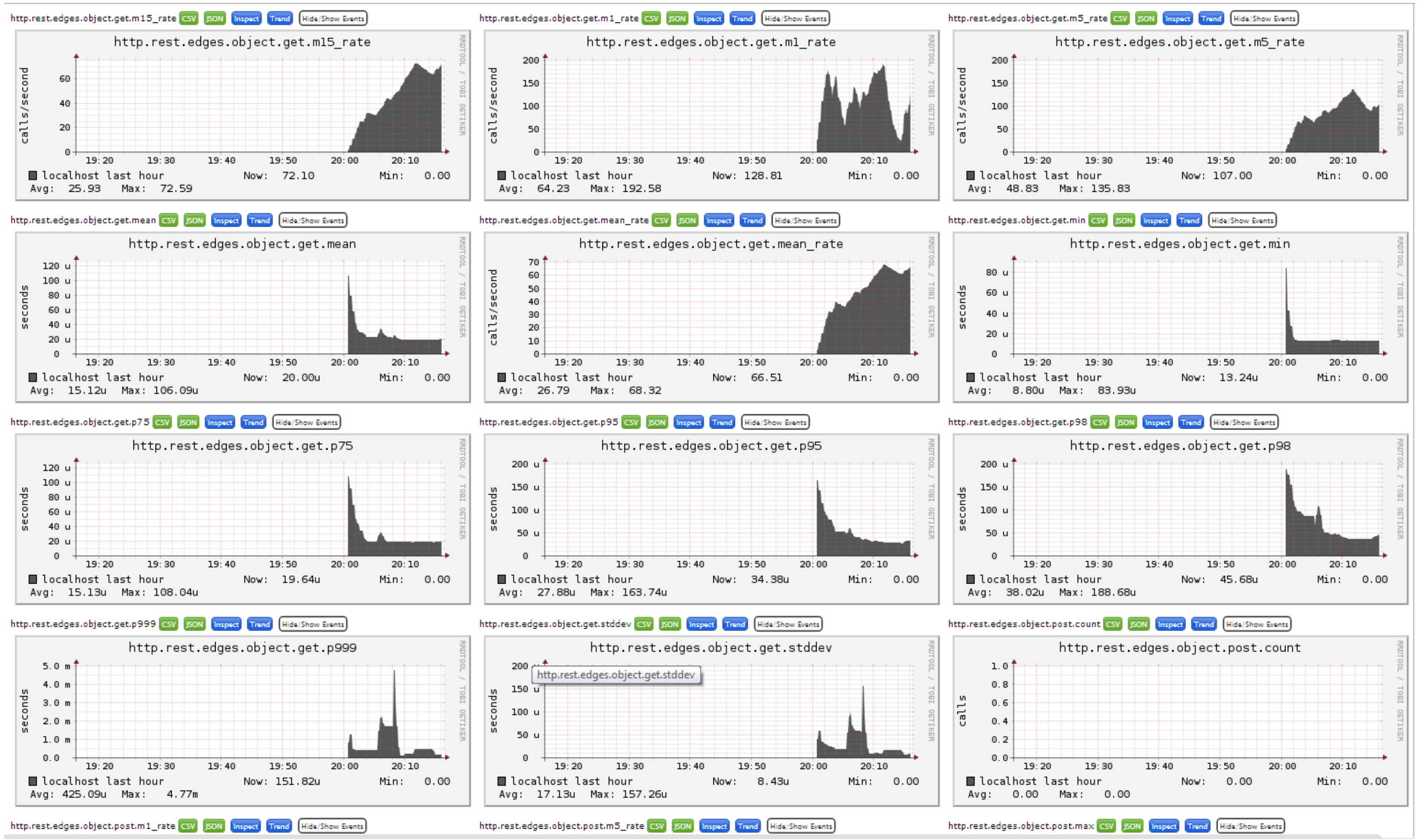
Ganglia monitoring system

- **Scalable distributed monitoring system**

- **Main responsibilities**
  - Monitor changes in host state
  - Announce relevant changes
  - Listen to the state of all other ganglia nodes via a unicast or multicast channel
  - Answer requests for an XML description of the cluster state

Example of Ganglia

- Easy to monitoring from multiple sources
- User defined performance factors can be added


http.rest.edges.object.get.m15_rate CSV JSON Inspect Trend Hide/Show Events
http.rest.edges.object.get.m15_rate
calls/second
localhost last hour Now: 72.10 Min: 0.00
Avg: 25.93 Max: 72.59
http.rest.edges.object.get.m1_rate CSV JSON Inspect Trend Hide/Show Events
http.rest.edges.object.get.m1_rate
calls/second
localhost last hour Now: 128.81 Min: 0.00
Avg: 64.23 Max: 192.58
http.rest.edges.object.get.m5_rate CSV JSON Inspect Trend Hide/Show Events
http.rest.edges.object.get.m5_rate
calls/second
localhost last hour Now: 107.00 Min: 0.00
Avg: 48.83 Max: 135.83
http.rest.edges.object.get.mean CSV JSON Inspect Trend Hide/Show Events
http.rest.edges.object.get.mean
seconds
localhost last hour Now: 20.00u Min: 0.00
Avg: 15.12u Max: 106.09u
http.rest.edges.object.get.mean_rate CSV JSON Inspect Trend Hide/Show Events
http.rest.edges.object.get.mean_rate
calls/second
localhost last hour Now: 66.51 Min: 0.00
Avg: 26.79 Max: 68.32
http.rest.edges.object.get.min CSV JSON Inspect Trend Hide/Show Events
http.rest.edges.object.get.min
seconds
localhost last hour Now: 13.24u Min: 0.00
Avg: 8.80u Max: 83.93u
http.rest.edges.object.get.p75 CSV JSON Inspect Trend Hide/Show Events
http.rest.edges.object.get.p75
seconds
localhost last hour Now: 19.64u Min: 0.00
Avg: 15.13u Max: 108.04u
http.rest.edges.object.get.p95 CSV JSON Inspect Trend Hide/Show Events
http.rest.edges.object.get.p95
seconds
localhost last hour Now: 34.38u Min: 0.00
Avg: 27.88u Max: 163.74u
http.rest.edges.object.get.p98 CSV JSON Inspect Trend Hide/Show Events
http.rest.edges.object.get.p98
seconds
localhost last hour Now: 45.68u Min: 0.00
Avg: 38.02u Max: 188.68u
http.rest.edges.object.get.p999 CSV JSON Inspect Trend Hide/Show Events
http.rest.edges.object.get.p999
seconds
localhost last hour Now: 151.82u Min: 0.00
Avg: 425.09u Max: 4.77m
http.rest.edges.object.get.stddev CSV JSON Inspect Trend Hide/Show Events
http.rest.edges.object.get.stddev
http.rest.edges.object.get.stddev
seconds
localhost last hour Now: 8.43u Min: 0.00
Avg: 17.13u Max: 157.26u
http.rest.edges.object.post.count CSV JSON Inspect Trend Hide/Show Events
http.rest.edges.object.post.count
calls
localhost last hour Now: 0.00 Min: 0.00
Avg: 0.00 Max: 0.00
19:20 19:30 19:40 19:50 20:00 20:10
RRDTOOL / TOBI OETIKER
http.rest.edges.object.post.m1_rate CSV JSON Inspect Trend Hide/Show Events
http.rest.edges.object.post.m5_rate CSV JSON Inspect Trend Hide/Show Events
http.rest.edges.object.post.max CSV JSON Inspect Trend Hide/Show Events

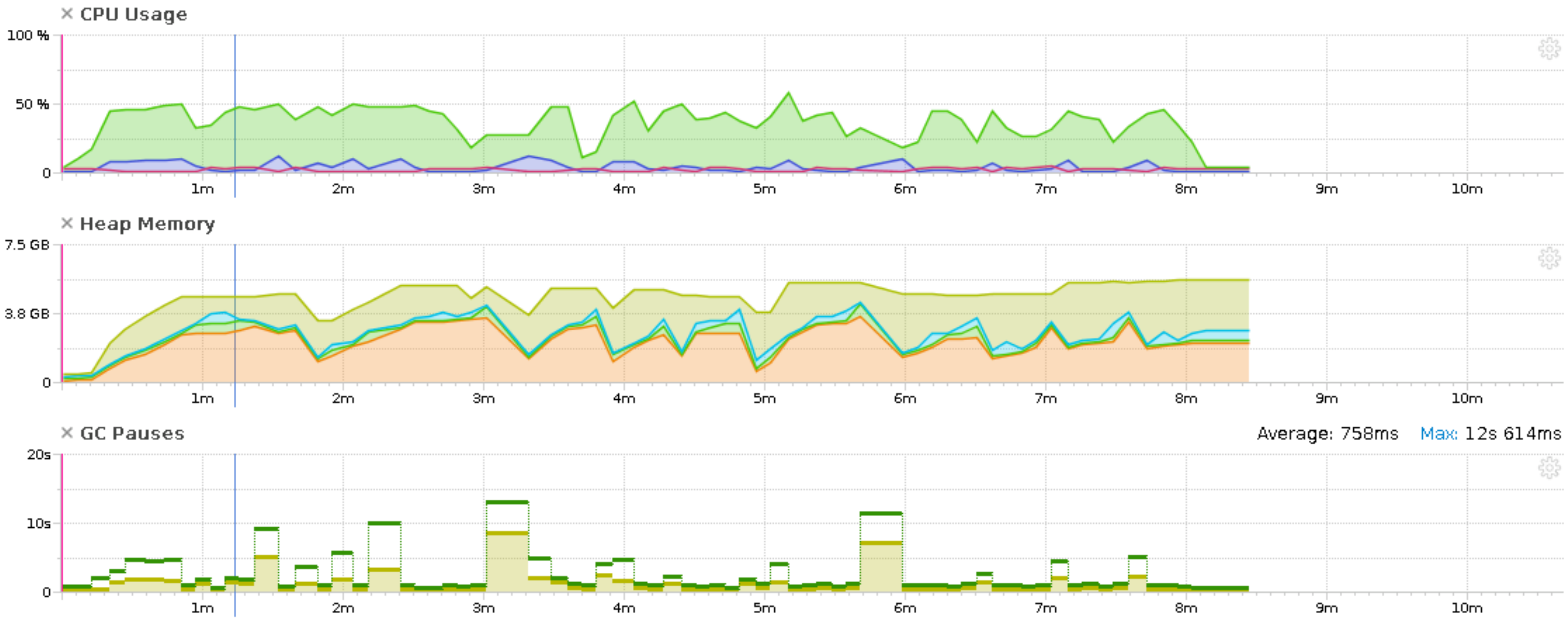
YourKit java profiler

- Commercial Java profiling tool that allows to generate CPU and memory profiles of running applications

- Support thread-level function-call tree

Example of YourKit

- **Real-time monitoring about running application**



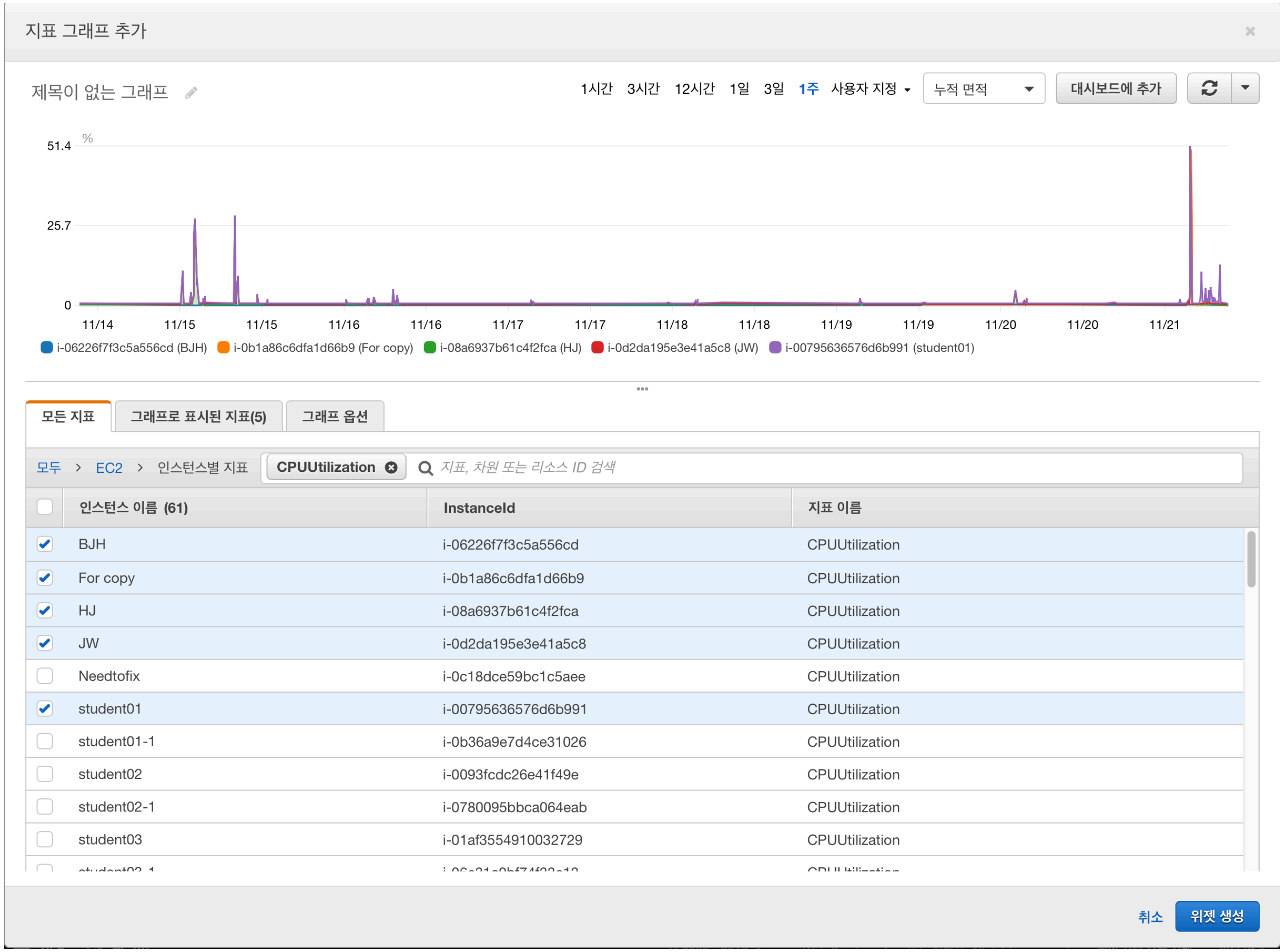
Amazon CloudWatch

- Monitoring service for AWS cloud resources and the applications run on AWS

- View metrics for CPU utilization, data transfer from Amazon ED2 instances

Example of Amazon CloudWatch

- **CPU utilization of instances in BDE3 class**



□	인스턴스 이름 (61)	InstanceId	지표 이름
☑	BJH	i-06226f7f3c5a556cd	CPUUtilization
☑	For copy	i-0b1a86c6dfa1d66b9	CPUUtilization
☑	HJ	i-08a6937b61c4f2fca	CPUUtilization
☑	JW	i-0d2da195e3e41a5c8	CPUUtilization
□	Needtofix	i-0c18dce59bc1c5aee	CPUUtilization
☑	student01	i-00795636576d6b991	CPUUtilization
□	student01-1	i-0b36a9e7d4ce31026	CPUUtilization
□	student02	i-0093fcdc26e41f49e	CPUUtilization
□	student02-1	i-0780095bbca064eab	CPUUtilization
□	student03	i-01af3554910032729	CPUUtilization
□	student03-1	[illegible]	CPUUtilization