Internet Measurement and Data Analysis (7)

Kenjiro Cho

2011-11-09

review of previous class

Class 6 Measuring the characteristics of the Internet

- delay, packet loss, jitter
- correlation and multivariate analysis
- principal component analysis
- exercise: correlation analysis

today's topics

Class 7 Measuring the diversity and complexity of the Internet

- sampling
- statistical analysis
- histogram
- exercise: histogram, CDF

complex systems

complex systems science

- a system with interfering components that as a whole exhibits complex behavior not obvious from the individual components
- the real world is full of complex systems
- difficult to analyze by traditional methods based on reductionism
  - need to understand a complex system as is, without decomposition
- many studies started in 1990's
  - few remaining problems that can be solved with reductionism
  - analysis and simulations enabled by computers

complexity of the Internet

complexity of topology (network science)

- scale-free: the degree distribution of nodes follows a power-law
  - many small-degree nodes and a small number of large-degree nodes
  - highest-degree nodes greatly exceed the average degree
- small-world:
  - compact: the average distance between 2 nodes is short
  - clusters: nodes are highly clustered

traffic behavior (time-series analysis)

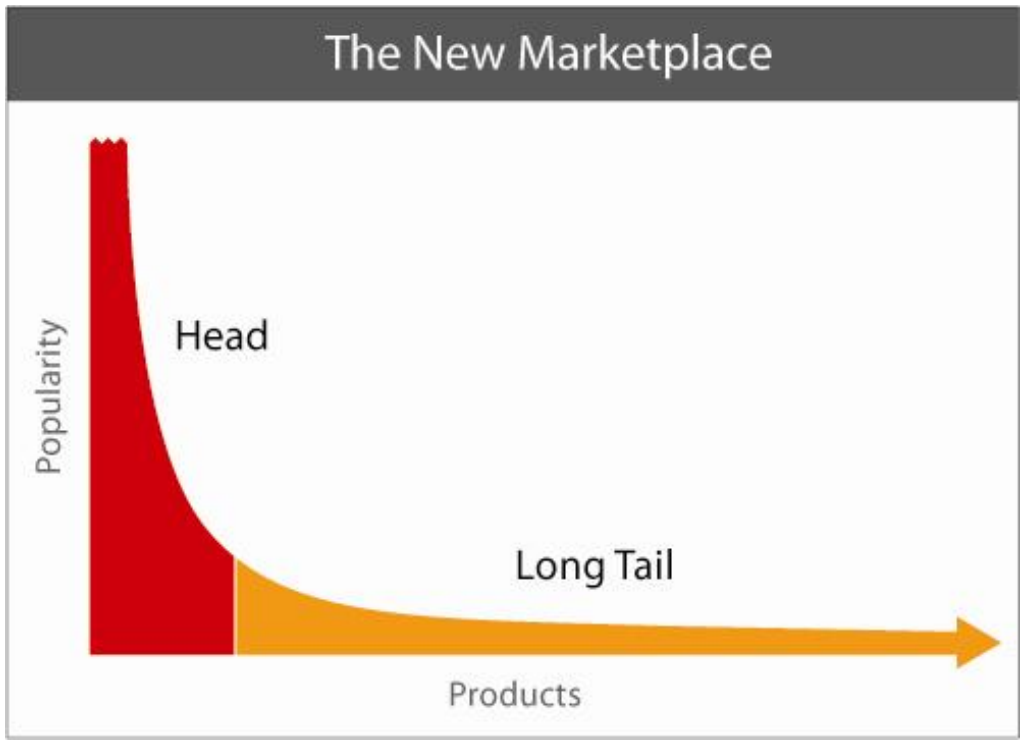
- self-similarity
- long-range dependence

long tail

a business model for online retail services

- head: a small number of bestseller items: for real stores
- tail: diverse low-sales items: covered by online stores

it is now widely used for diverse niche market

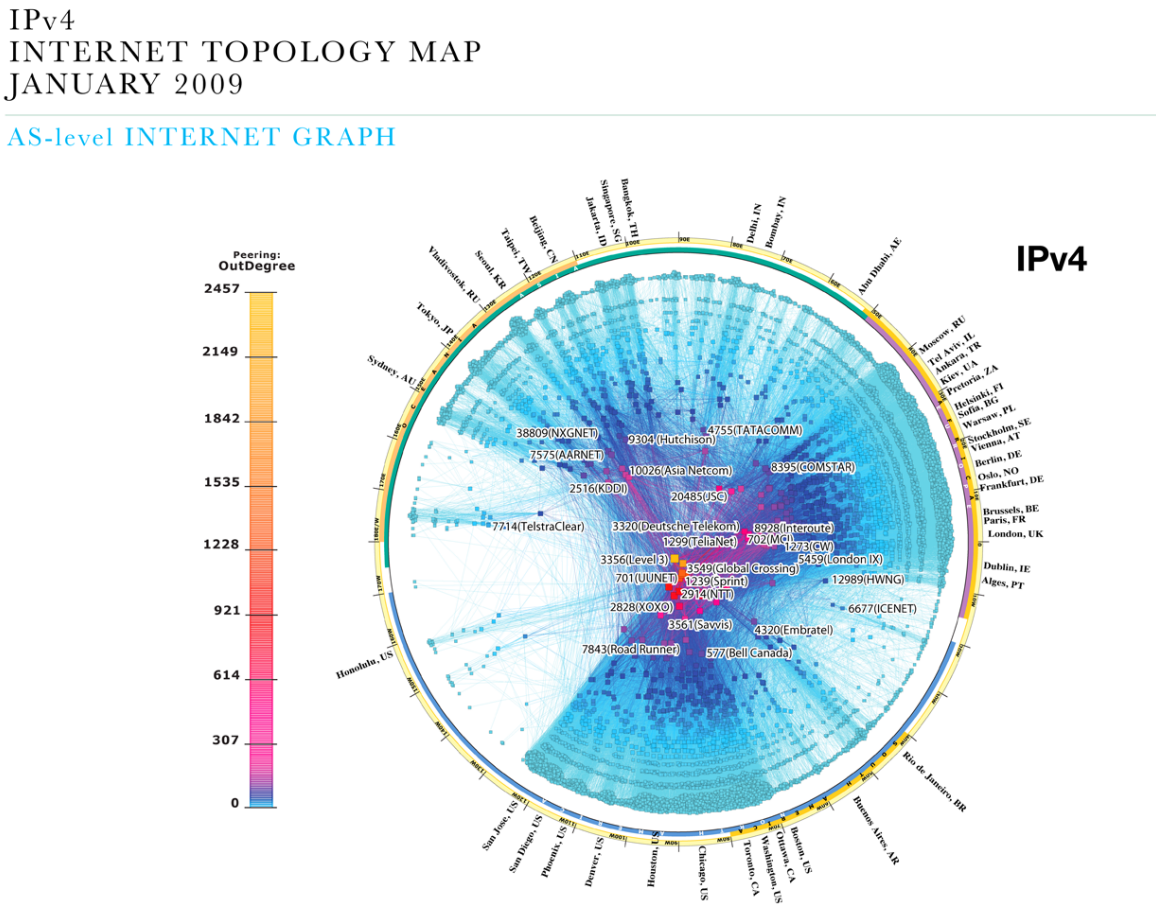


source: `http://longtail.com/`

example: AS structure of the Internet

CAIDA AS CORE MAP 2009/03

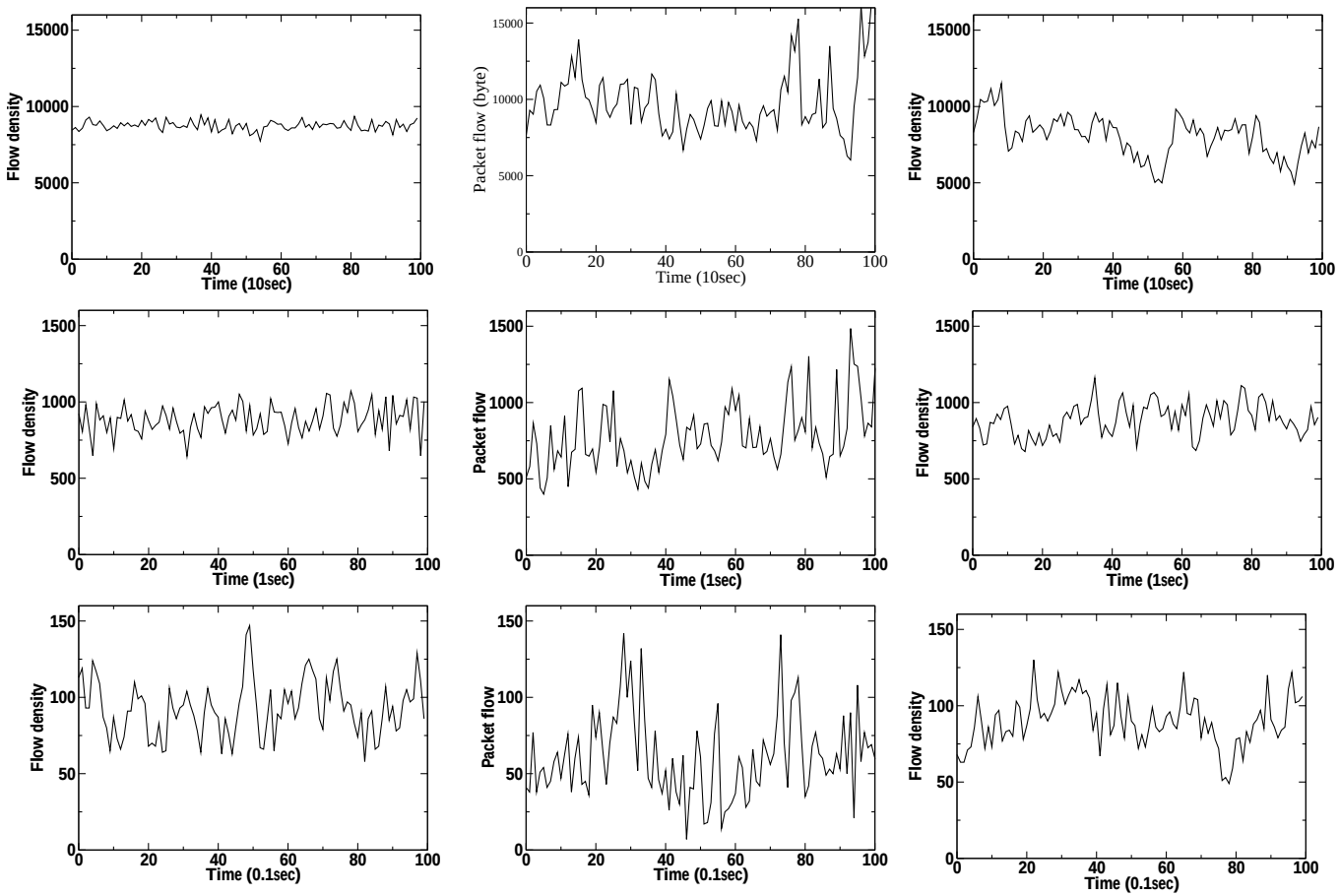
- visualization of AS topology using skitter/ark data
- longitude of AS (registered location), out-degree of AS



http://www.caida.org/research/topology/as_core_network/

self-similarity in network traffic

- exponential model (left), real traffic (middle), self-similar model (right)
- time scale: 10sec (top), 1 sec (middle), 0.1 sec (bottom)



diversity in Internet data

- different behavior can be observed from different locations
  - country, region, time
  - enterprise/university/home, backbone/access network

typical network does not exist!

- how to measure and describe diversity
- methods for sampling

sampling

- investigating the whole population (census): not realistic in most cases
- sampling is needed

sampling for the Internet

- observation points
- time, duration
- packet, flow

packet sampling methods

- counter-based 1/N sampling (deterministic)
  - simple to implement, widely used
  - possible synchronization with targets of measurement
- probabilistic 1/N sampling
  - probabilistically select packets (or other objects)
- sampling by time
  - example: take the first minute every hour
- flow-based sampling
  - probabilistically sample new flows
  - observe all packets belonging to a selected flow
  - advantage: able to analyze flow behaviors
- many other sampling methods

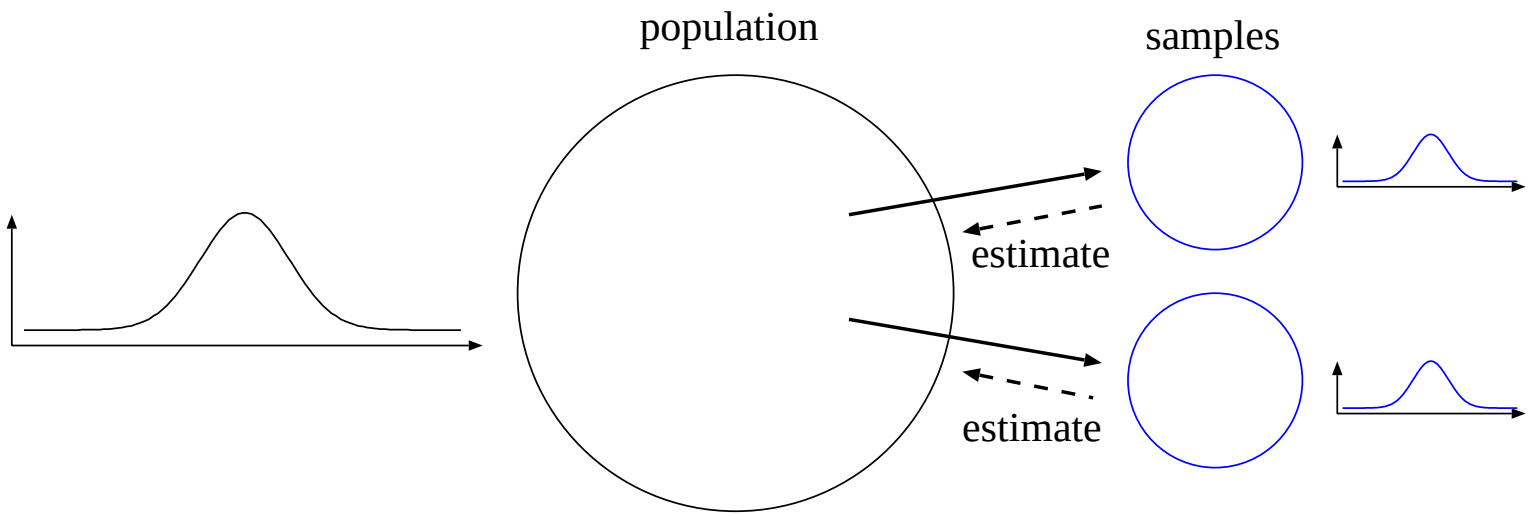
sampling: sample and population

summary statistics and statistical inference

- summary statistics: numbers that summarize properties of data (e.g., mean and standard deviation)
- statistical inference: makes inferences about the population based on samples using statistical methods

population: whole data (difficult or impossible to obtain for most cases)

- need to infer properties of the population from samples
- variables: properties of the population (fixed)
- statistics: inferred values based on samples (varying)



expected value

the expected value $E(X)$ of stochastic variable X: mean

- discrete model

$$E(X) = \mu = \sum_{i=1}^{n} x_i p_i$$

- continuous model

$$E(X) = \mu = \int_{-\infty}^{\infty} x f(x) dx$$

properties of expected values

- $E(c) = c$
- $E(X + c) = E(X) + c$
- $E(cX) = cE(X)$
- $E(X + Y) = E(X) + E(Y)$

sample mean

- sample mean: $\bar{x}$

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

- sample variance: s^2

$$s^2 = \frac{1}{n-1} \sum_{i=1}^{n} (x_i - \bar{x})^2$$

- sample standard deviation: s
- note: divide sum of squares by $(n-1)$, not by n
  - degree of freedom: the number of independent variables in the sum of squares is $(n-1)$ because of $\bar{x}$

law of large numbers and central limit theorem

law of large numbers

- as the number of samples increases, the sample mean converges to the population mean

central limit theorem

- the mean of a sufficiently large number of samples is approximately normally distributed, regardless of the original distribution. $N(\mu, \sigma^2/n)$
- when the population is normally distributed, it can be applied even when n is small

standard error

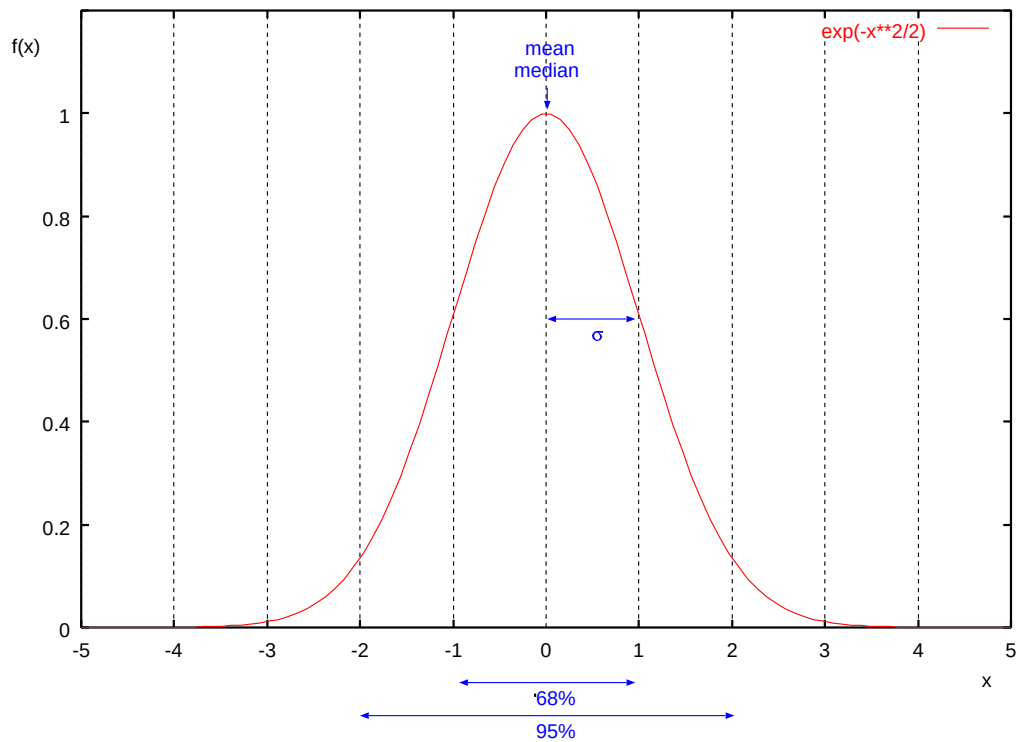
standard error: standard deviation of sample mean (SE)

$$SE = \sigma/\sqrt{n}$$

- you can improve the precision by increasing the number of samples n
  - standard error becomes smaller but with only $1/\sqrt{n}$
- the distribution of sample mean from a normal distribution $N(\mu, \sigma)$ will be a normal distribution with mean μ and standard deviation SE $= \sigma/\sqrt{n}$

normal distribution

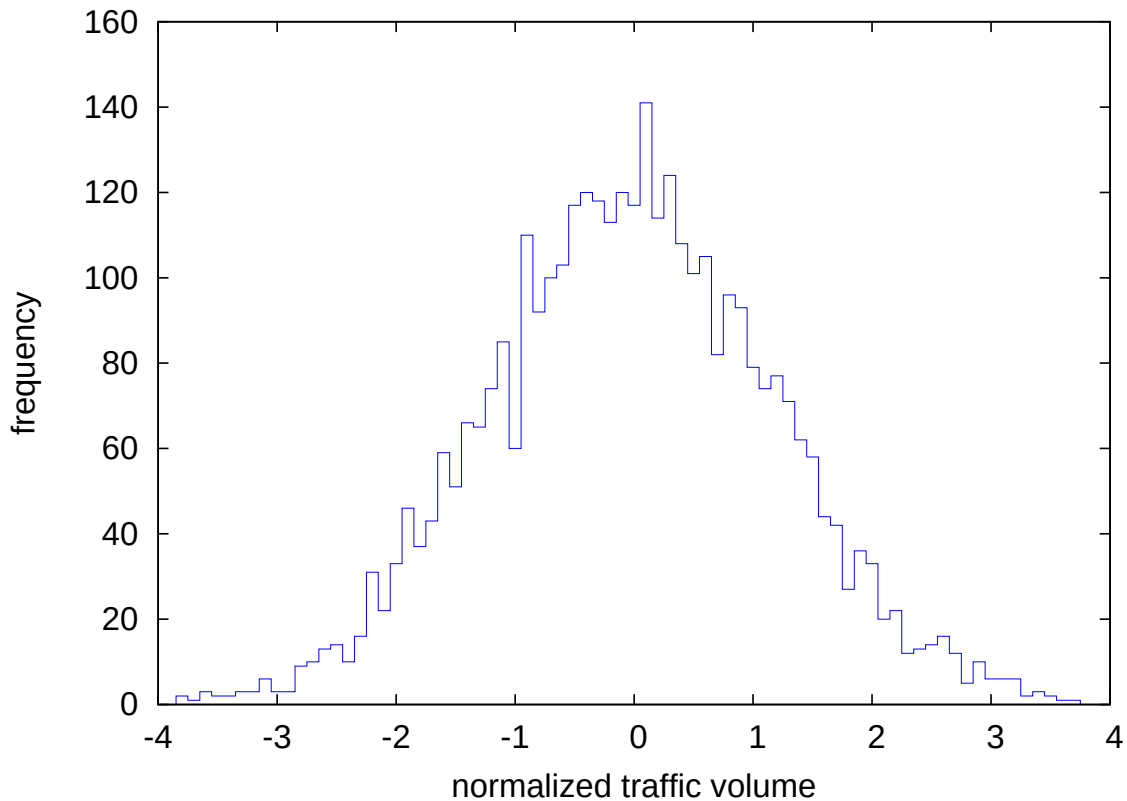
- also known as gaussian distribution
- defined by 2 parameters: μ:mean, σ^2:variance
- sum of random variables follows normal distribution
- standard normal distribution: $\mu = 0, \sigma = 1$
- in normal distribution
  - 68% within $(mean - stddev, mean + stddev)$
  - 95% within $(mean - 2 * stddev, mean + 2 * stddev)$



histogram (1/2)

to see distribution of the data set

- split the data into equal-sized bins by value
- count the frequency of each bin
- X axis: variable, Y axis: frequency



histogram (2/2)

with histograms

- you can identify
  - center (i.e., the location) of the data
  - spread (i.e., the scale) of the data
  - skewness of the data
  - presence of outliers
  - presence of multiple modes in the data

limitations of histograms

- needs appropriate bin size
  - too small: each bin doesn't have enough samples (e.g., empty bins)
  - too large: only few regions available
  - difficult for highly skewed distribution
- enough samples needed

methods to select bin size

Sturges' formula: the number of bins k, the number of samples n

$$k = \log_2 n + 1$$

Scott's choice: bin width h, standard deviation σ, the number of samples n

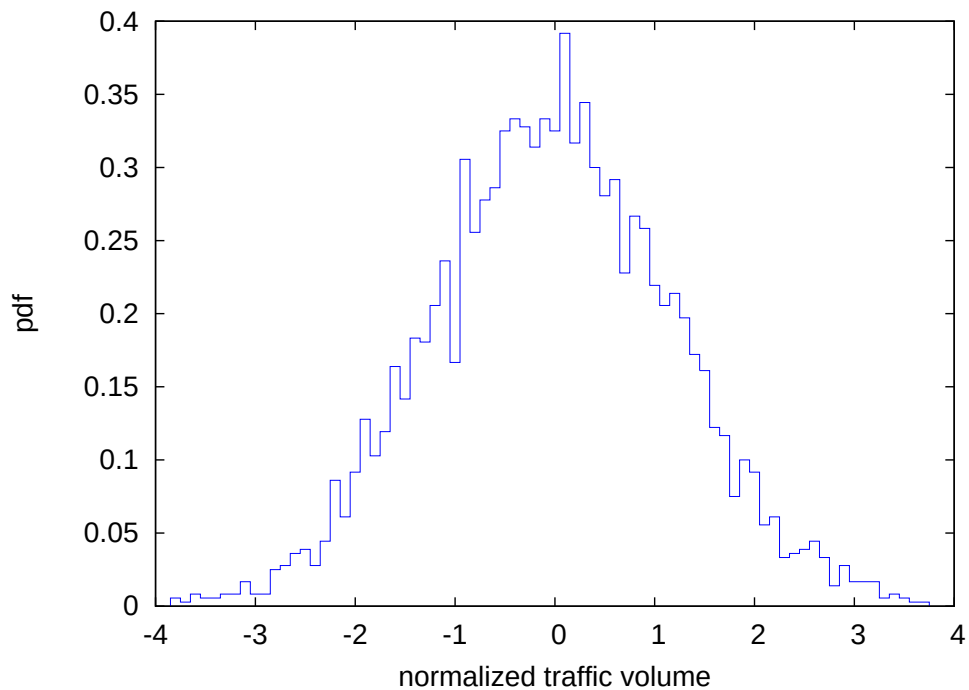
$$h = \frac{3.5\sigma}{n^{1/3}}$$

- these are just guidelines. you need to find an appropriate size depending on the distribution and the meaning of variables.

probability density function; pdf

- normalize the frequency (count) to make the sum of the area under the histogram to be 1
  - divide the count by (the total number of observations * the bin width)
- probability density function: probability of observing x

$$f(x) = P[X = x]$$



cumulative distribution function; cdf

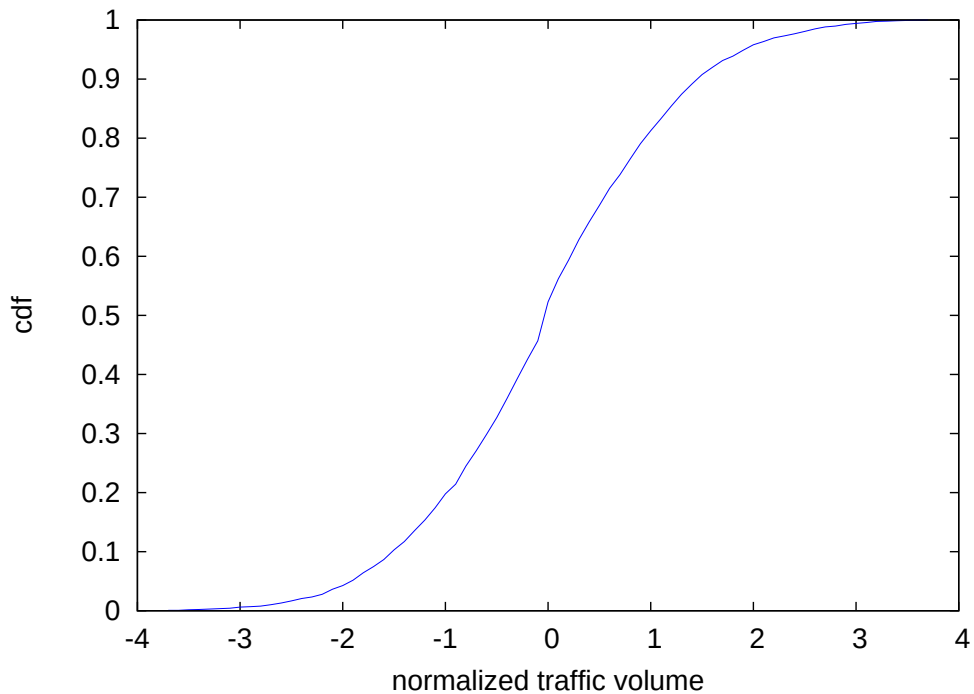
- density function: probability of observing x

$$f(x) = P[X = x]$$

- cumulative distribution function: probability of observing x or less

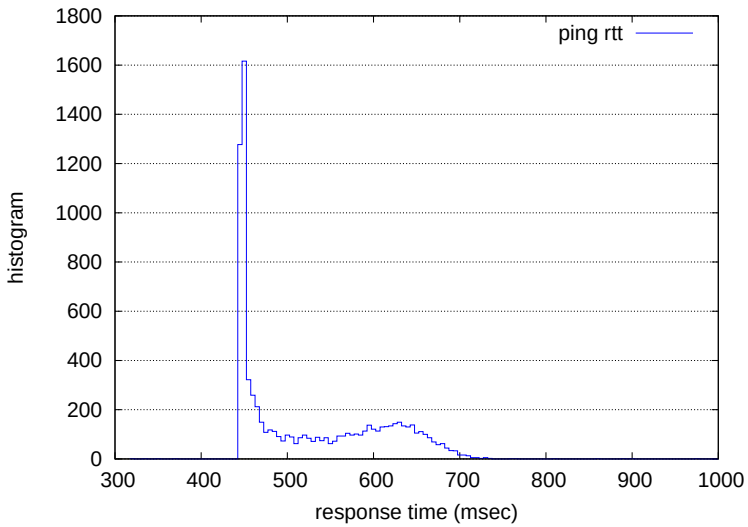
$$F(x) = P[X <= x]$$

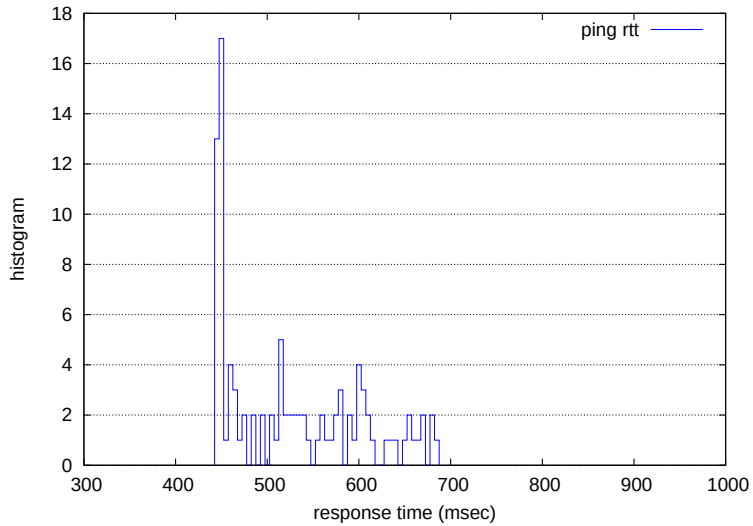
- better than histogram when distribution is highly skewed, sample count is not enough, or outliers are not negligible

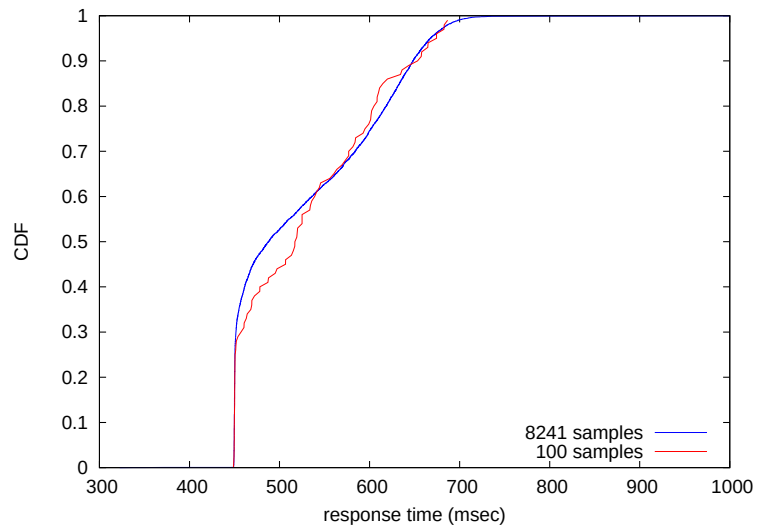


histogram vs cdf

- no need to worry about bin size or sample count for cdf







original data (left), 100 samples (right), cdfs (bottom)

complementary cumulative distribution function (ccdf)

in power-law distribution, the tail of distribution is often of interest

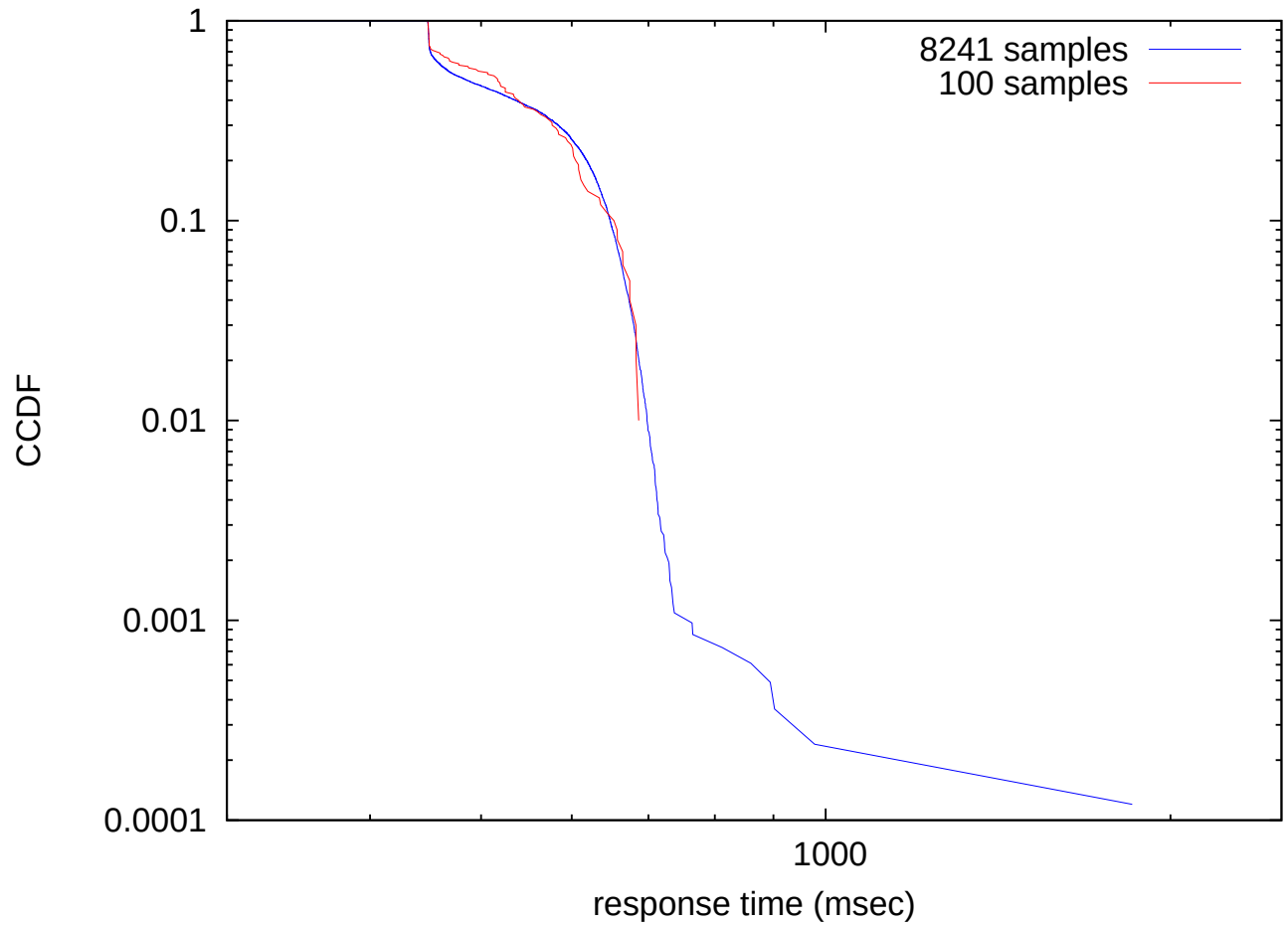
ccdf: probability of observing x or more

$$F(x) = 1 - P[X <= x]$$

- plot ccdf in log-log scale
  - to see the tail of the distribution or scaling property

example CCDF

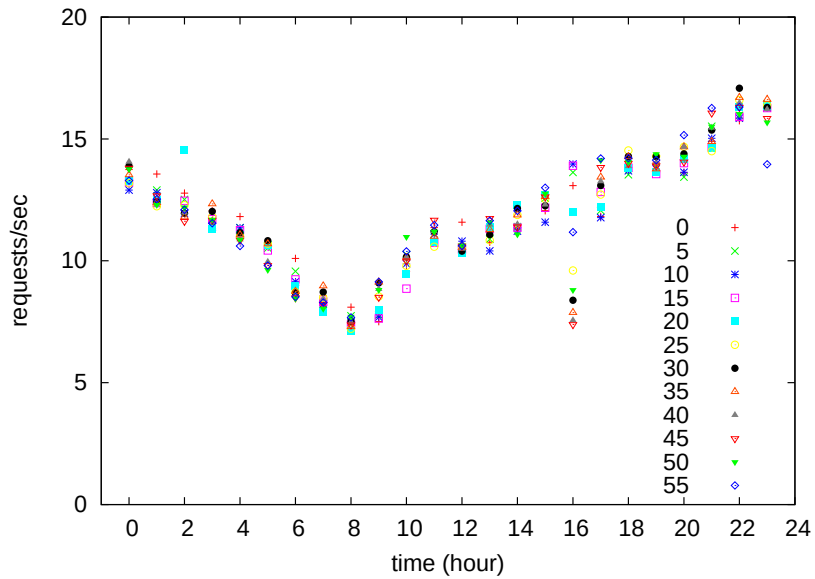
- comparing the tails of the original data and sampled data



previous exercise: correlation

request-table in Class 4:

- use the 5-minute bin output from the previous class
- focus on the request counts
- use 12 5-minute bins for an hour as 12 samples per hour



previous exercise: correlation (cont'd)

an hourly request table

- row: hourly data (0 .. 23)
- column: hour samples(00 05 10 ... 55) mean stddev

correlation computation

- use 12 5-minute bins as 12 time-series
- compute correlation coefficient among time slots (columns)

```
#hour  00     05     10     ...     55    mean     stddev
 0    4123   3963   3871    ...    3987   4046.8    102.3
 1    4068   3871   3838    ...    3760   3774.9    106.2
 2    3833   3755   3580    ...    3628   3703.6    219.0
 3    3614   3433   3418    ...    3462   3515.5     86.2
...
22    4724   4790   4757    ...    4893   4882.2    113.4
23    4922   4932   4889    ...    4188   4818.9    203.8
```

previous exercise: correlation matrix

compute correlation matrix for the 12 time-series data

correlation matrix

	00	05	10	15	20	25	30	35	40	45	50	55
00	1.000	0.982	0.983	0.974	0.953	0.917	0.872	0.849	0.850	0.841	0.862	0.870
05	0.982	1.000	0.992	0.982	0.961	0.914	0.871	0.845	0.848	0.841	0.874	0.899
10	0.983	0.992	1.000	0.983	0.944	0.904	0.855	0.828	0.825	0.814	0.848	0.882
15	0.974	0.982	0.983	1.000	0.963	0.917	0.864	0.846	0.841	0.831	0.864	0.901
20	0.953	0.961	0.944	0.963	1.000	0.951	0.910	0.893	0.893	0.876	0.902	0.912
25	0.917	0.914	0.904	0.917	0.951	1.000	0.985	0.981	0.972	0.956	0.965	0.939
30	0.872	0.871	0.855	0.864	0.910	0.985	1.000	0.994	0.992	0.982	0.975	0.932
35	0.849	0.845	0.828	0.846	0.893	0.981	0.994	1.000	0.992	0.982	0.972	0.917
40	0.850	0.848	0.825	0.841	0.893	0.972	0.992	0.992	1.000	0.993	0.986	0.931
45	0.841	0.841	0.814	0.831	0.876	0.956	0.982	0.982	0.993	1.000	0.987	0.937
50	0.862	0.874	0.848	0.864	0.902	0.965	0.975	0.972	0.986	0.987	1.000	0.959
55	0.870	0.899	0.882	0.901	0.912	0.939	0.932	0.917	0.931	0.937	0.959	1.000

previous exercise: sample code (1/2)

```
#!/usr/bin/env ruby

# read request-table.txt
re = /(\d+)\s+(\d+)\s+(\d+)\s+(\d+)\s+(\d+)\s+(\d+)\s+(\d+)\s+(\d+)\s+(\d+)\s+(\d+)\s+(\d+)\s+(\d+)\s+(\d

hourly = Array.new(24){ Array.new(12) }
ARGF.each_line do |line|
  if re.match(line)
    for min in 0 .. 11
      hourly[$1.to_i][min] = Regexp.last_match(min + 2).to_i
    end
  end
end

means = Array.new(12)
for min in 0 .. 11
  mean = 0
  for hour in 0 .. 23
    mean += hourly[hour][min]
  end
  means[min] = Float(mean) / 24
end
```

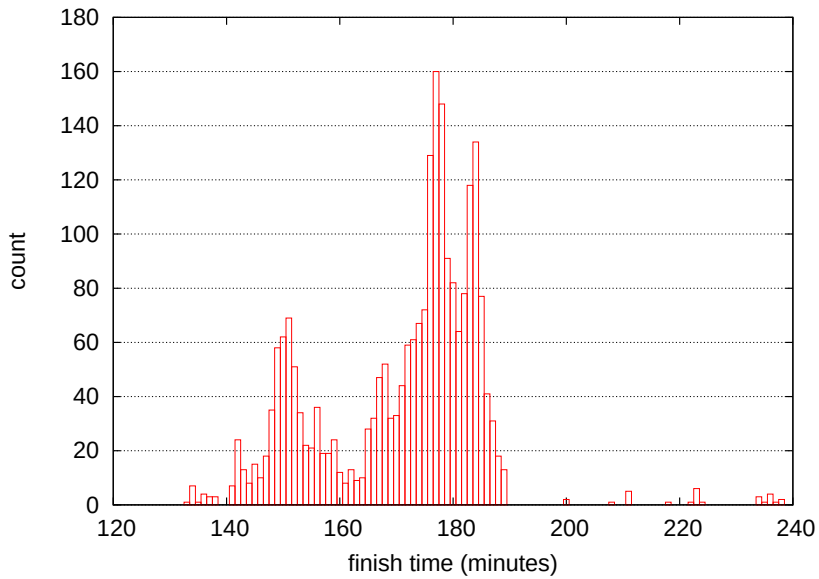
previous exercise: sample code (2/2)

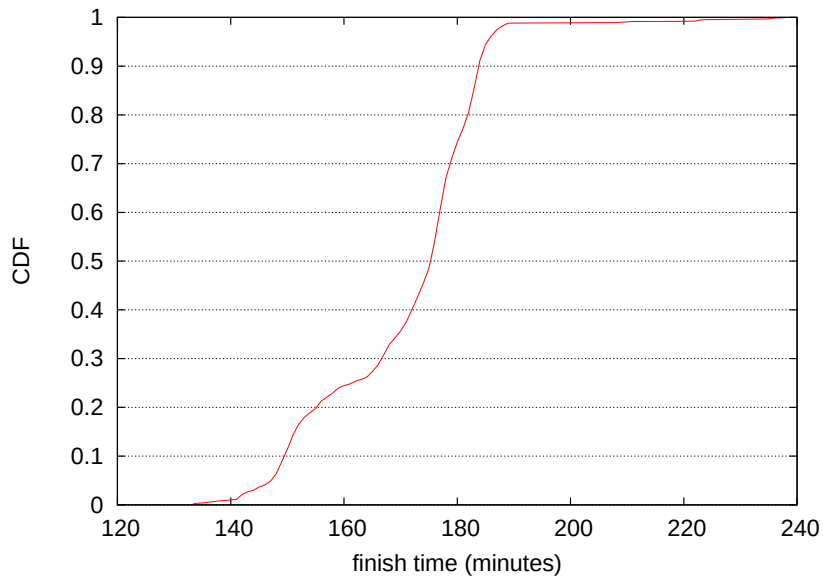
```
cc_matrix = Array.new(12){ Array.new(12) }
for m0 in 0 .. 11
  for min in 0 .. 11
    cov = 0
    sum_dx2 = sum_dy2 = 0
    for hour in 0 .. 23
      x = hourly[hour][m0]
      y = hourly[hour][min]
      cov += (x - means[m0]) * (y - means[min])
      sum_dx2 += (x - means[m0])**2
      sum_dy2 += (y - means[min])**2
    end
    cc_matrix[m0][min] = Float(cov) / Math.sqrt(sum_dx2 * sum_dy2)
  end
end

for m0 in 0 .. 11
  for min in 0 .. 11
    printf "%.3f ", cc_matrix[m0][min]
  end
  print "\n"
end
```

exercise: histogram and CDF

- distribution of finish time of a city marathon (from Class 2)
- plot a CDF this time





exercise: histogram and CDF (cont'd)

- distribution of finish time of a city marathon (from Class 2)
- plot a CDF this time

original:

```
# Minutes Count
133 1
134 7
135 1
136 4
137 3
138 3
141 7
142 24
...
```

add cumulative count:

```
# Minutes Count CumulativeCount
133 1 1
134 7 8
135 1 9
136 4 13
137 3 16
138 3 19
141 7 26
142 24 50
```

answers: assignment 1

- assignment: compute mean and standard deviation of traffic, plot the results
  - similar to today's exercise but for traffic (not for request counts)
- to understand programming of statistical procedures and graph plotting
- data: use the 5-min bin outputs from the previous exercise
- items to submit
  1. traffic data table
  2. graph 1: a plot of 12 samples per hour for 24 hours
  3. graph 2: a plot of mean and standard deviation of traffic for 24 hours
- the results should look similar to ones for the request counts
  - you need to adjust the number of digits for table outputs
  - use "Mbps" for the unit of traffic in the graphs

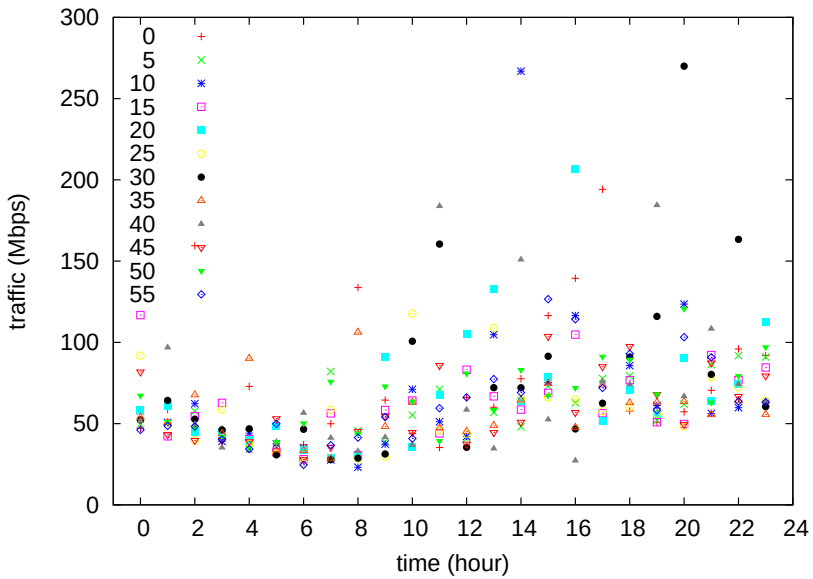
answers: traffic data table

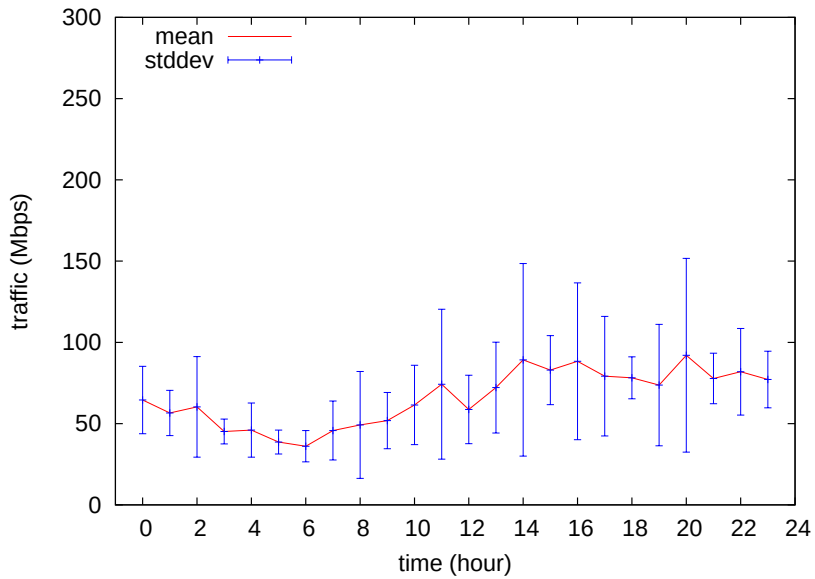
the table should look like:

```
#hour        00           05   ...           mean         stddev
 0   1766135158   1857342919   ...   2420355075.6    777770181.4
 1   2202831446   2322940598   ...   2120850506.4    521760120.1
 2   5980871926   2158091698   ...   2261318711.4   1161290997.4
 3   1741001140   1609648229   ...   1692879169.6    286988721.2
...
```

answers: graphs for traffic

- graph 1: 12 samples per hour, for 24 hours
- graph 2: mean and standard deviation for 24 hours





summary

Class 7 Measuring the diversity and complexity of the Internet

- sampling
- statistical analysis
- histogram
- exercise: histogram, CDF

next class

Class 8 Distributions (11/16)

- normal distribution and other distributions
- confidence intervals
- statistical tests
- exercise: generating distributions, confidence intervals
- **assignment 2**