Internet Measurement and Data Analysis (7)

Kenjiro Cho

2013-11-13

review of previous class

Class 6 Correlation (11/6)

- Online recommendation systems
- Distance
- Correlation coefficient
- exercise: correlation analysis

today's topics

Class 7 Multivariate analysis

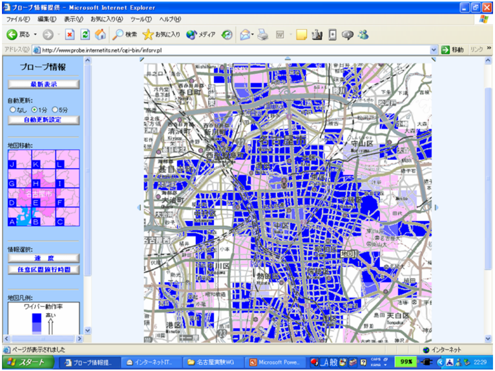
- Data sensing and GeoLocation
- Linear regression
- Principal Component Analysis
- exercise: linear regression

data sensing

- data sensing: collecting data from remote site
- it becomes possible to access various sensor information over the Internet
  - weather information, power consumption, etc.

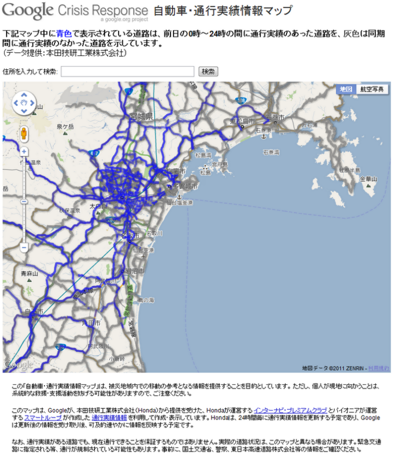
example: Internet vehicle experiment

- by WIDE Project in Nagoya in 2001
  - location, speed, and wiper usage data from 1,570 taxis
  - blue areas indicate high ratio of wiper usage, showing rainfall in detail



Japan Earthquake

- the system is now part of ITS
- usable roads info released 3 days after the quake
  - data provide by HONDA (TOYOTA, NISSAN)



source: google crisis response

GeoLocation Services

- to provide different services according to the user location
- map, navigation, timetable for public transportation
- search for nearby restaurants or other shops (for advertisement)
- possibilities for other services

example: 駅.Locky (Eki.Locky)

- train timetable service by Kawaguchi Lab, Nagoya University
  - popular app from a WiFi GeoLocation research project
- App for iPhone/Android
- automatically select the nearest station and show timetable
  - geo-location by GPS/WiFi
  - also collect WiFi access point info seen by the device
- countdown for the next train
  - can show timetalbe as well
- crowdsourcing: timetable database contributed by users

GPS (Global Positioning System)

- about 30 satellites for GPS
- originally developed for US military use
  - for civilian use, the accuracy was intentionally degraded to about 100m
  - in 2000, the accuracy was improved to about 10m by removing intentional noise
- wide variety of civilian usage
  - car navigation, mobile phones, digital cameras
- location measurement: locate the position by distances from 3 GPS satellites
  - GPS signal includes satellite position and time information
  - distance is calculated by the difference in the time in the signal
  - needs 4 satellites to calibrate the time of the receiver
  - the accuracy improves as more satellites are used
- limitations
  - needs to see satellites
  - initialization time to obtain initial positioning
- improvements: combine with accelerometers and gyro sensors

geo-location using access points

- a communication device knows its associated access point
  - an access point also knows associated devices
  - a device can receive signals from non-associated access points
- there exit services to get location information from access points
- can be used indoors
  - other approaches: sonic signals, visible lights
- can be combined with GPS to improve accuracy

measurement metrics of the Internet

measurement metrics

- link capacity, throughput
- delay
- jitter
- packet loss rate

methodologies

- active measurement: injects measurement packets (e.g., ping)
- passive measurement: monitors network without interfering in traffic
  - monitor at 2 locations and compare
  - infer from observations (e.g., behavior of TCP)
  - collect measurements inside a transport mechanism

delay measurement

- delay components
  - delay = propagation delay + queueing delay + other overhead
  - if not congested, delay is close to propagation deley
- methods
  - round-trip delay
  - one-way delay requires clock synchronization

  - average delay
  - max delay: e.g., voice communication requires $< 400ms$
  - jitter: variations in delay

some delay numbers

- packet transmission time (so called wire-speed)
  - 1500 bytes at 10Mbps: 1.2msec
  - 1500 bytes at 100Mbps: 120usec
  - 1500 bytes at 1Gbps: 12usec
  - 1500 bytes at 10Gbps: 1.2usec
- speed of light in fiber: about 200,000 km/s
  - 100km round-trip: 1 msec
  - 20,000km round-trip: 200msec
- satellite round-trip delay
  - LEO (Low-Earth Orbit): 200 msec
  - GEO (Geostationary Orbit): 600msec

packet loss measurement

packet loss rate

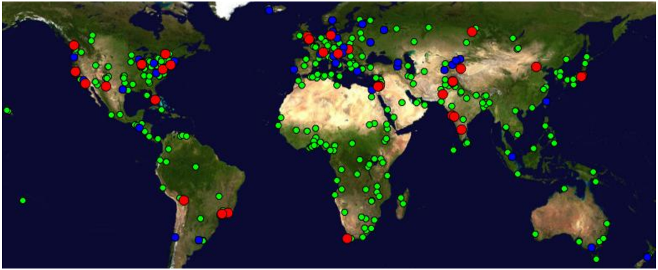
- loss rate is enough if packet loss is random...
- in reality,
  - bursty loss: e.g., buffer overflow
  - packet size dependency: e.g., bit error rate in wireless transmission

pingER project

- the Internet End-to-end Performance Measurement (IEPM) project by SLAC
- using ping to measure rtt and packet loss around the world
  - http://www-iepm.slac.stanford.edu/pinger/
  - started in 1995
  - over 600 sites in over 125 countries

pingER project monitoring sites

- monitoring (red), beacon (blue), remote (green) sites
  - beacon sites are monitored by all monitors



from pingER web site

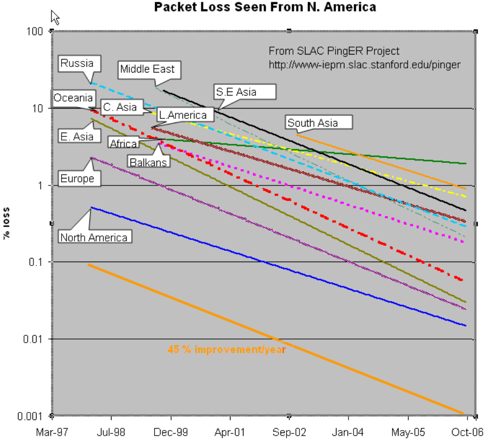
pingER project monitoring sites in east asia

- monitoring (red) and remote (green) sites

from pingER web site

pingER packet loss

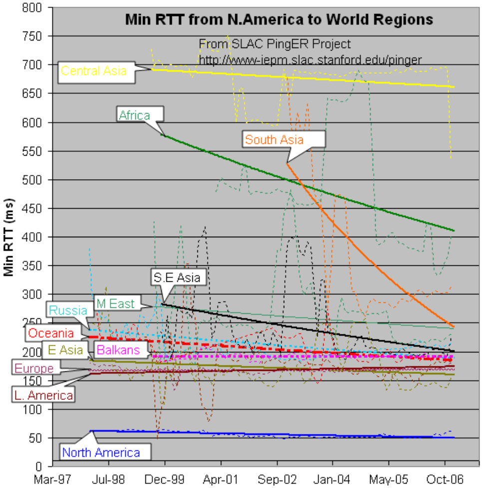
- packet loss observed from N. Ameria
- exponential improvement in 10 years



from pingER web site

pinger minimum rtt

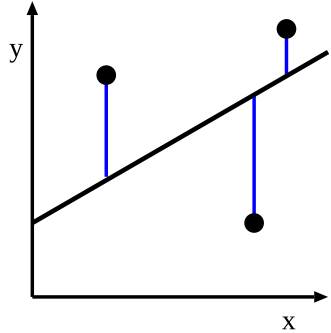
- minimum rtts observed from N. America
- gradual shift from satellite to fiber in S. Asia and Africa

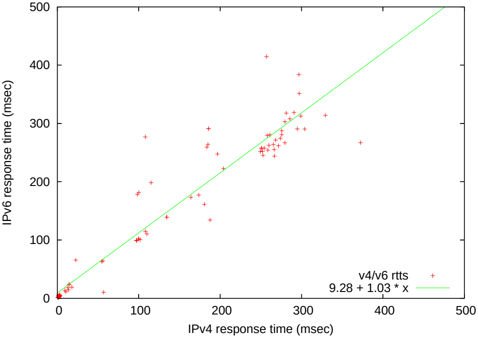


from pingER web site

linear regression

- fitting a straight line to data
  - least square method: minimize the sum of squared errors





least square method

a linear function minimizing squared errors

$$f(x) = b_0 + b_1 x$$

2 regression parameters can be computed by

$$b_1 = \frac{\sum xy - n\bar{x}\bar{y}}{\sum x^2 - n(\bar{x})^2}$$

$$b_0 = \bar{y} - b_1 \bar{x}$$

where

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

$$\sum xy = \sum_{i=1}^{n} x_i y_i \qquad \sum x^2 = \sum_{i=1}^{n} (x_i)^2$$

a derivation of the expressions for regression parameters

The error in the ith observation: $e_i = y_i - (b_0 + b_1 x_i)$
For a sample of n observations, the mean error is

$$\bar{e} = \frac{1}{n}\sum_i e_i = \frac{1}{n}\sum_i (y_i - (b_0 + b_1 x_i)) = \bar{y} - b_0 - b_1\bar{x}$$

Setting the mean error to 0, we obtain: $b_0 = \bar{y} - b_1\bar{x}$
Substituting b_0 in the error expression:
$e_i = y_i - \bar{y} + b_1\bar{x} - b_1 x_i = (y_i - \bar{y}) - b_1(x_i - \bar{x})$
The sum of squared errors, SSE, is

$$SSE = \sum_{i=1}^{n} e_i^2 = \sum_{i=1}^{n}[(y_i - \bar{y})^2 - 2b_1(y_i - \bar{y})(x_i - \bar{x}) + b_1^2(x_i - \bar{x})^2]$$

$$\begin{aligned}\frac{SSE}{n} &= \frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2 - 2b_1\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})(x_i - \bar{x}) + b_1^2\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 \\ &= \sigma_y^2 - 2b_1\sigma_{xy}^2 + b_1^2\sigma_x^2\end{aligned}$$

The value of b_1, which gives the minimum SSE, can be obtained by differentiating this equation with respect to b_1 and equating the result to 0:

$$\frac{1}{n}\frac{d(SSE)}{db_1} = -2\sigma_{xy}^2 + 2b_1\sigma_x^2 = 0$$

That is: $b_1 = \frac{\sigma_{xy}^2}{\sigma_x^2} = \frac{\sum xy - n\bar{x}\bar{y}}{\sum x^2 - n(\bar{x})^2}$

principal component analysis; PCA

purpose of PCA

- convert a set of possibly correlated variables into a smaller set of uncorrelated variables

PCA can be solved by eigenvalue decomposition of a covariance matrix

applications of PCA

- demensionality reduction
  - sort principal components by contribution ratio, components with small contribution ratio can be ignored
- principal component labeling
  - find means of produced principal components

notes:

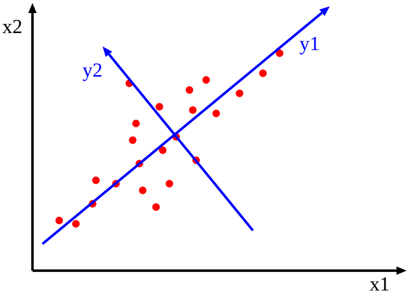
- PCA just extracts components with large variance
  - not simple if axes are not in the same unit
- a convenient method to automatically analyze complex relationship, but it does not explain the complex relationship

PCA: intuitive explanation

a view of cordinate transformation using a 2D graph

- draw the first axis (the 1st PCA axis) that goes through the centroid, along the direction of the maximal variability
- draw the 2nd axis that goes through the centroid, is orthogonal to the 1st axis, along the direction of the 2nd maximal variability
- draw the subsequent axes in the same manner

For example, "height" and "weight" can be mapped to "body size" and "slimness". we can add "sitting height" and "chest measurement" in a similar manner



PCA (appendix)

principal components can be found as the eigenvectors of a covariance matrix.
let X be a d-demensional random variable. we want to find a $d \times d$ orthogonal transformation matrix P that converts X to its principal components Y.

$$\mathsf{Y} = \mathsf{P}^\top \mathsf{X}$$

solve this equation, assuming $cov(\mathsf{Y})$ being a diagonal matrix (components are independent), and P being an orthogonal matrix. $(\mathsf{P}^{-1} = \mathsf{P}^\top)$
the covariance matrix of Y is

$$\begin{aligned} cov(\mathsf{Y}) &= \mathsf{E}[\mathsf{Y}\mathsf{Y}^\top] = \mathsf{E}[(\mathsf{P}^\top\mathsf{X})(\mathsf{P}^\top\mathsf{X})^\top] = \mathsf{E}[(\mathsf{P}^\top\mathsf{X})(\mathsf{X}^\top\mathsf{P})] \\ &= \mathsf{P}^\top\mathsf{E}[\mathsf{X}\mathsf{X}^\top]\mathsf{P} = \mathsf{P}^\top cov(\mathsf{X})\mathsf{P} \end{aligned}$$

thus,

$$\mathsf{P}cov(\mathsf{Y}) = \mathsf{P}\mathsf{P}^\top cov(\mathsf{X})\mathsf{P} = cov(\mathsf{X})\mathsf{P}$$

rewrite P as a $d \times 1$ matrix:

$$\mathsf{P} = [\mathsf{P}_1, \mathsf{P}_2, \ldots, \mathsf{P}_d]$$

also, $cov(\mathsf{Y})$ is a diagonal matrix (components are independent)

$$cov(\mathsf{Y}) = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_d \end{bmatrix}$$

this can be rewritten as

$$[\lambda_1\mathsf{P}_1, \lambda_2\mathsf{P}_2, \ldots, \lambda_d\mathsf{P}_d] = [cov(\mathsf{X})\mathsf{P}_1, cov(\mathsf{X})\mathsf{P}_2, \ldots, cov(\mathsf{X})\mathsf{P}_d]$$

for $\lambda_i\mathsf{P}_i = cov(\mathsf{X})\mathsf{P}_i$, P_i is an eigenvector of the covariance matrix X
thus, we can find a transformation matrix P by finding the eigenvectors.

assignment 1: the finish time distribution of a marathon

- purpose: investigate the distribution of a real-world data set
- data: the finish time records from honolulu marathon 2012
  - http://results.sportstats.ca/res2012/honolulumarathon_m.htm
  - the number of finishers: 24,070
- items to submit
  1. mean, standard deviation and median of the total finishers, male finishers, and female finishers
  2. the distributions of finish time for each group (total, men, and women)
     - plot 3 histograms for 3 groups
     - use 10 minutes for the bin size
     - use the same scale for the axes to compare the 3 plots
  3. CDF plot of the finish time distributions of the 3 group
     - plot 3 groups in a single graph
  4. discuss differences in finish time between male and female. what can you observe from the data?
  5. optional
     - other analysis of your choice (e.g., discussion on differences among age groups)
- submission format: a single PDF file including item 1-5
- submission method: upload the PDF file through SFC-SFS
- submission due: 2013-11-07

honolulu marathon data set

data format

```
      Chip     Pace                                                   Gender    Category       @10km  @21.1  @3
Place Time     /mi        #    Name                City         ST CNT Plce/Tot Plc/Tot Category Split1 Split2
---- -------- ---- ----- ------------------- ------------ -- --- -------- ------- ------ ----- ------- ---
   1 02:12:31 5:04     6 Kipsang, Wilson     Iten            KEN  1/12690  1/16   MElite 31:40 1:07:07 1:3
   2 02:13:08 5:05     7 Geneti, Markos      Addis Ababa     ETH  2/12690  2/16   MElite 31:39 1:07:02 1:3
   3 02:14:15 5:08    11 Kimutai, Kiplimo    Eldoret         KEN  3/12690  3/16   MElite 31:40 1:07:02 1:3
   4 02:14:55 5:09     2 Ivuti, Patrick      Kangundo        KEN  4/12690  4/16   MElite 31:40 1:07:02 1:3
   5 02:15:17 5:10    12 Arile, Julius       Kepenguria      KEN  5/12690  5/16   MElite 31:39 1:07:02 1:3
   6 02:15:53 5:11     9 Bouramdane, Abderr  Champs De Cou   MAR  6/12690  6/16   MElite 31:40 1:07:01 1:3
   7 02:18:27 5:17     8 Manza, Nicholas     Ngong Hills     KEN  7/12690  7/16   MElite 31:39 1:07:01 1:3
   8 02:19:46 5:20     1 Chelimo, Nicholas   Ngong Hills     KEN  8/12690  8/16   MElite 31:40 1:07:02 1:3
   9 02:25:23 5:33 20850 Harada, Taku        Nagoya-Shi   AI JPN  9/12690  1/1238 M25-29 31:54 1:09:52 1:4
  10 02:27:12 5:37 25474 Hagawa, Eiichi      Matsumoto    NA JPN 10/12690  1/1501 M30-34 32:46 1:12:21 1:4
...
```

- Chip Time: finish time
- Category: MElite, WElite, M15-19, M20-24, ..., W15-29, W20-24, ...
  - note some runners have "No Age" for Category
- Country: 3-letter country code: e.g., JPN, USA
  - note some runners have "UK" for country-code
- check the number of the total finishers when you extract the finishers

item 1: computing mean, standard deviation and median

- round off to minute (slightly different from using seconds)
- exclude "No Age" for the male and female groups

	n	mean	stddev	median
all	24,070	369.1	94.2	357
men	12,532	350.5	93.2	338
women	11,537	389.3	91.0	381

example script to extract data

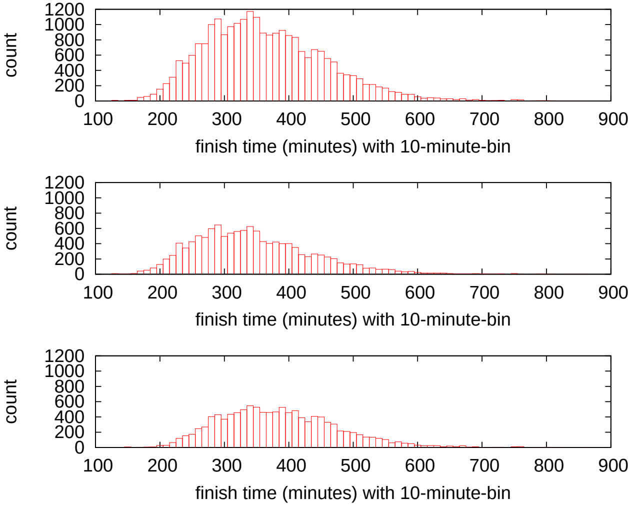
```
# regular expression to read chiptime and category from honolulu.htm
re = /\s*\d+\s+(\d{2}:\d{2}:\d{2})\s+.*((?:[MW](?:Elite|\d{2}\-\d{2})|No Age))/

filename = ARGV[0]

open(filename, 'r') do |io|
  io.each_line do |line|
    if re.match(line)
      puts "#{$1}\t#{$2}"
    end
  end
end
```

item 2: histograms for 3 groups

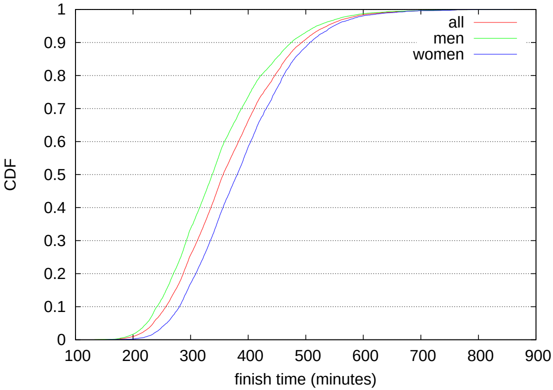
- plot 3 histograms for 3 groups
- use 10 minutes for the bin size
- use the same scale for the axes to compare the 3 plots



finish time histograms total(top) men(middle) women(bottom)

item 3: CDF plot of the finish time distributions of the 3 group

- plot 3 groups in a single graph

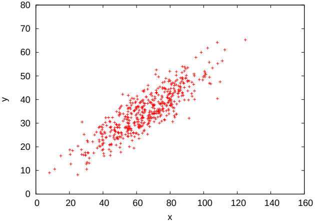


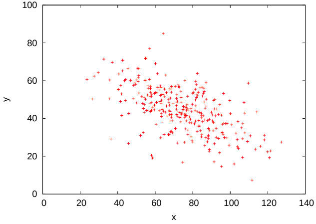
previous exercise: computing correlation coefficient

- compute correlation coefficient using the sample data sets
  - correlation-data-1.txt, correlation-data-2.txt

correlation coefficient

$$\rho_{xy} = \frac{\sigma_{xy}^2}{\sigma_x \sigma_y} = \frac{\sum_{i=1}^{n} x_i y_i - \frac{(\sum_{i=1}^{n} x_i)(\sum_{i=1}^{n} y_i)}{n}}{\sqrt{(\sum_{i=1}^{n} x_i^2 - \frac{(\sum_{i=1}^{n} x_i)^2}{n})(\sum_{i=1}^{n} y_i^2 - \frac{(\sum_{i=1}^{n} y_i)^2}{n})}}$$





data-1:r=0.87 (left), data-2:r=-0.60 (right)

script to compute correlation coefficient

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

# regular expression for matching 2 floating numbers
re = /([-+]?\d+(?:\.\d+)?)\s+([-+]?\d+(?:\.\d+)?)/

sum_x = 0.0 # sum of x
sum_y = 0.0 # sum of y
sum_xx = 0.0 # sum of x^2
sum_yy = 0.0 # sum of y^2
sum_xy = 0.0 # sum of xy
n = 0 # the number of data

ARGF.each_line do |line|
    if re.match(line)
      x = $1.to_f
      y = $2.to_f
      sum_x += x
      sum_y += y
      sum_xx += x**2
      sum_yy += y**2
      sum_xy += x * y
      n += 1
    end
end

r = (sum_xy - sum_x * sum_y / n) /
  Math.sqrt((sum_xx - sum_x**2 / n) * (sum_yy - sum_y**2 / n))

printf "n:%d r:%.3f\n", n, r
```

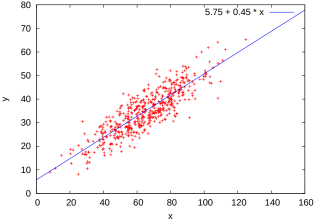
today's exercise: linear regression

- linear regression by the least square method
- use the data for the previous exercise
  - correlation-data-1.txt, correlation-data-2.txt

$$f(x) = b_0 + b_1 x$$

$$b_1 = \frac{\sum xy - n\bar{x}\bar{y}}{\sum x^2 - n(\bar{x})^2}$$

$$b_0 = \bar{y} - b_1 \bar{x}$$



data-1:r=0.87 (left), data-2:r=-0.60 (right)

script for linear regression

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

# regular expression for matching 2 floating numbers
re = /([-+]?\d+(?:\.\d+)?)\s+([-+]?\d+(?:\.\d+)?)/

sum_x = sum_y = sum_xx = sum_xy = 0.0
n = 0
ARGF.each_line do |line|
    if re.match(line)
      x = $1.to_f
      y = $2.to_f

      sum_x += x
      sum_y += y
      sum_xx += x**2
      sum_xy += x * y
      n += 1
    end
end

mean_x = Float(sum_x) / n
mean_y = Float(sum_y) / n
b1 = (sum_xy - n * mean_x * mean_y) / (sum_xx - n * mean_x**2)
b0 = mean_y - b1 * mean_x

printf "b0:%.3f b1:%.3f\n", b0, b1
```

adding the least squares line to scatter plot

```
set xrange [0:160]
set yrange [0:80]

set xlabel "x"
set ylabel "y"

plot "correlation-data-1.txt" notitle with points, \
5.75 + 0.45 * x lt 3
```

summary

Class 7 Multivariate analysis

- Data sensing and GeoLocation
- Linear regression
- Principal Component Analysis
- exercise: linear regression

next class

Class 8 Time-series analysis (11/27)

- Internet and time
- Network Time Protocol
- Time series analysis
- exercise: time-series analysis
- **assignment 2**