

# **PROGRAMMING**

## Lecture 18

Dept. of Computer Engineering  
Hanbat National University

# OUTLINE

---

- Data analysis
- Data plotting

# DATA ANALYSIS

---

## PROBLEM 1: ANALYSIS OF USD-KRW EXCHANGE RATES

Text files, "1994.txt", "1995. txt", ..... , and "2009.txt", in a folder, "data" contain the daily USD-KRW exchange rates for 16 years from 1994 to 2009, respectively . The format of each file is:

```
.....  
.....  
2009/05/11    0. 00080110  
.....
```

The exchange rate is given as USD per KRW.

(Continued)

---

For data analysis, read all files (16 files) into a **single long list of tuples**, each consisting of two integers that represent a date and an exchange rate (USD/KRW), respectively, as shown below:

[ ....., (20090511, 1248.28), .....]

Notice that the **exchange rate** is **inversed**.

---

## Pseudo code

For every year in 1994 - 2009 ,

1. Open the file for each year.
2. Read the exchange rate data of each day.
3. Re-format the data.
4. Append the data to a list.

**Type in the first code** given in the **hand-out** to implement a program for solving this problem.

---

## Steps 1 and 2 : Open files and read data.

```
years = range(1994, 2010)
```

```
data = [ ]
```

```
for yr in years:
```

```
    fname = "data/ %d.txt" % yr
```

```
    f = open(fname, "r")
```

step 1

```
        for line in f:
```

```
            do Step 2
```

step 2

---

## Step 3: Re-format the data

How to convert date to an integer

"2009/5/11"  $\Rightarrow$  20090511

```
yr, mh, dy = date.split("/")
```

```
date = 10000 * int(yr) + 100 * int(mh) + int(dy)
```

How to inverse the exchange rate

$$\text{KRW/USD} = 1/(\text{USD/KRW})$$

---

## PROBLEM 2: SUMMARY REPORT

Report the maximum and minimum exchange rates over the 16 years together with the date these occur. Also report the average exchange rate for every year.

Minimum: (19950705, 755)

Maximum: (19971223, 1960)

.....

1997 952

1998 1397

.....

**Incrementally add new functions** to your implemented version of the first program in order to obtain a solution for PROBLEM 2. You should also **replace** the function **main()** of the **first program** with a new one.



---

## Pseudo code

For all year in 1994 – 2009,

1. Compute the **average exchange rate** for **each year**.
2. Find the **minimum exchange rate** and the **date** on which it occurs.
3. Find the **maximum exchange rate** and the **date** on which it occurs.
4. Print the **minimum** and **maximum exchange rates** for the **entire years** and their corresponding dates.
5. Print the **average exchange rates** for the **years**.

---

## Step1: Finding the yearly average exchange rates

```
def average(data, yr):  
    sum = 0  
    count = 0  
    start = yr * 10000  
    end = (yr + 1) * 10000  
    for d, v in data:  
        if start < d < end:  
            sum += v  
            count += 1  
    return sum / count
```

---

### PROBLEM 3: MIN./MAX. MONTHLY RATES:

Modify your program(for Problem 2) to print the maximum and minimum exchange rates of every month for all years as shown below:

```
.....  
.....  
1998: 1519/1806 1519/1705 .....1191/1239  
.....  
.....
```

(Continued)

---

**Remove** functions, `average(., ., .)`, `find_min(.)`, and `find_max(.)` from your program. Also **replace** the function **`main()`** with a **new one**.

# DATA PLOTTING

---

## PROBLEM 4: DATA PLOTTING

**Modify** your **program** for solving problem 3 to plot the monthly maximum and minimum exchange rate data. You should **replace** the function **main()** of your program with a **new one**, and put the following statements at the **beginning** of the **new program**, which will be used for **plotting** the **data**, as explained in the **next few slides**:

```
import cs1media
x_step = 5
min_y = 700
max_y = 2000
y_step = 0.5:
```

# Data plotting with cs1media

---



Background grid

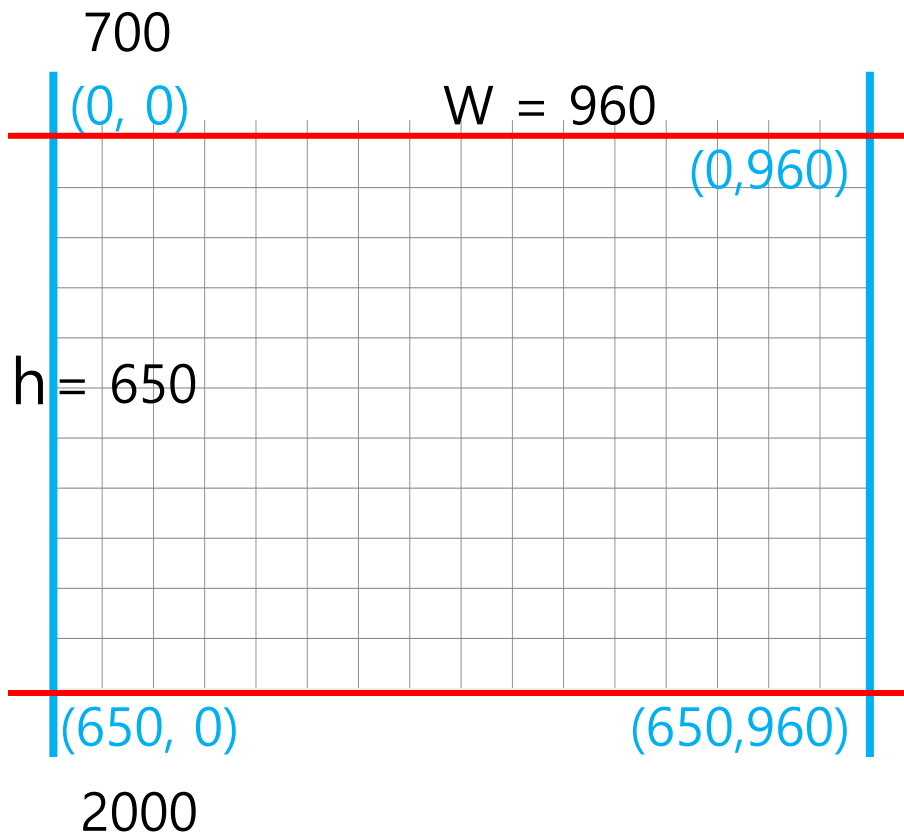
Plotted data

---

## Pseudo code

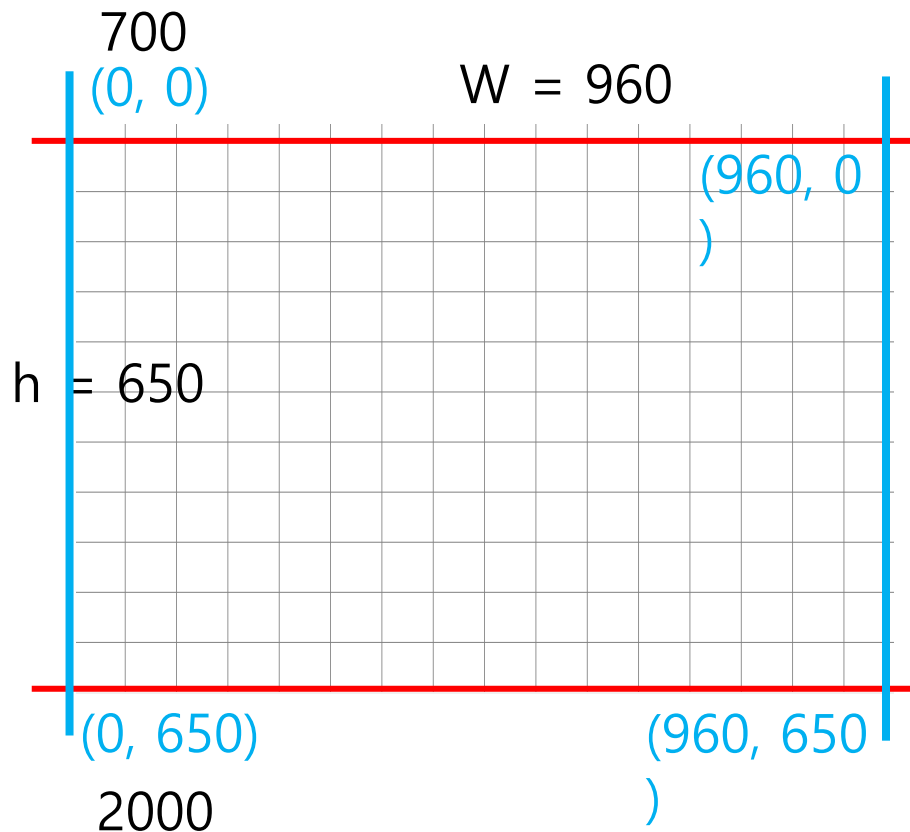
1. Draw a background grid.
2. For every year, do the following:
  - 2.1 Find the monthly maximum and minimum rates.
  - 2.2 Plot the data prepared in step 2.1.

## Step1: Drawing a background grid



```
x_step = 5  
min_y = 700  
max_y = 2000  
y_step = 0.5  
w = len(years) * 12 * x_step  
h = int((max_y - min_y) * y_step)
```





**#vertical line for every january**

for yr in years:

$x = (\text{yr} - \text{years}[0]) * 12 * x\_step$

for y in range(h):

$p.set(x, y, \text{cs1media.Color.gray})$

**# horizontal line per 100 Won**

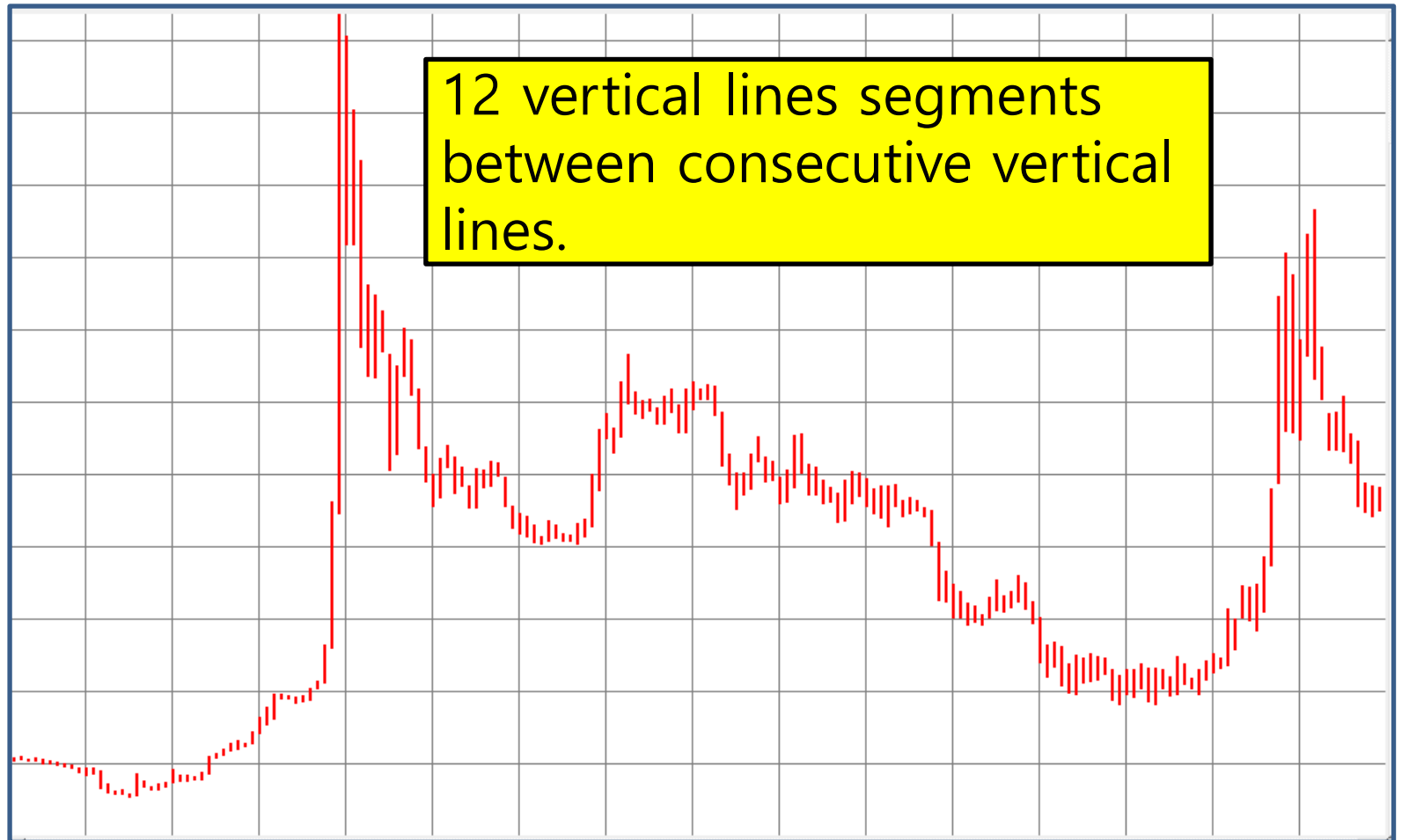
for won in range(min\_y, max\_y, 100):

$y = \text{int}((\text{won} - \text{min\_y}) * y\_step)$

for x in range(w):

$p.set(x, y, \text{cs1media.Color.gray})$

## Step 2.2: Plotting the data



---

```
# plot exchange rate
```

```
for yr in years:
```

```
    minmax = find_minmax(yr)
```

```
    for m in range(12):
```

```
        x = ((yr - years[0]) * 12 + m) * x_step
```

```
        y1 = int((minmax[m][0] - min_y) * y_step)
```

```
        y2 = int((minmax[m][1] - min_y) * y_step)
```

```
        for y in range(h - y2, h - y1 + 1):
```

Why?

```
            p.set(x, y, cs1media.Color.red)
```

```
            p.set(x+1, y, cs1media.Color.red)
```

Creating  
two endpoints

Minimum

Maximum

Why two lines ?

---

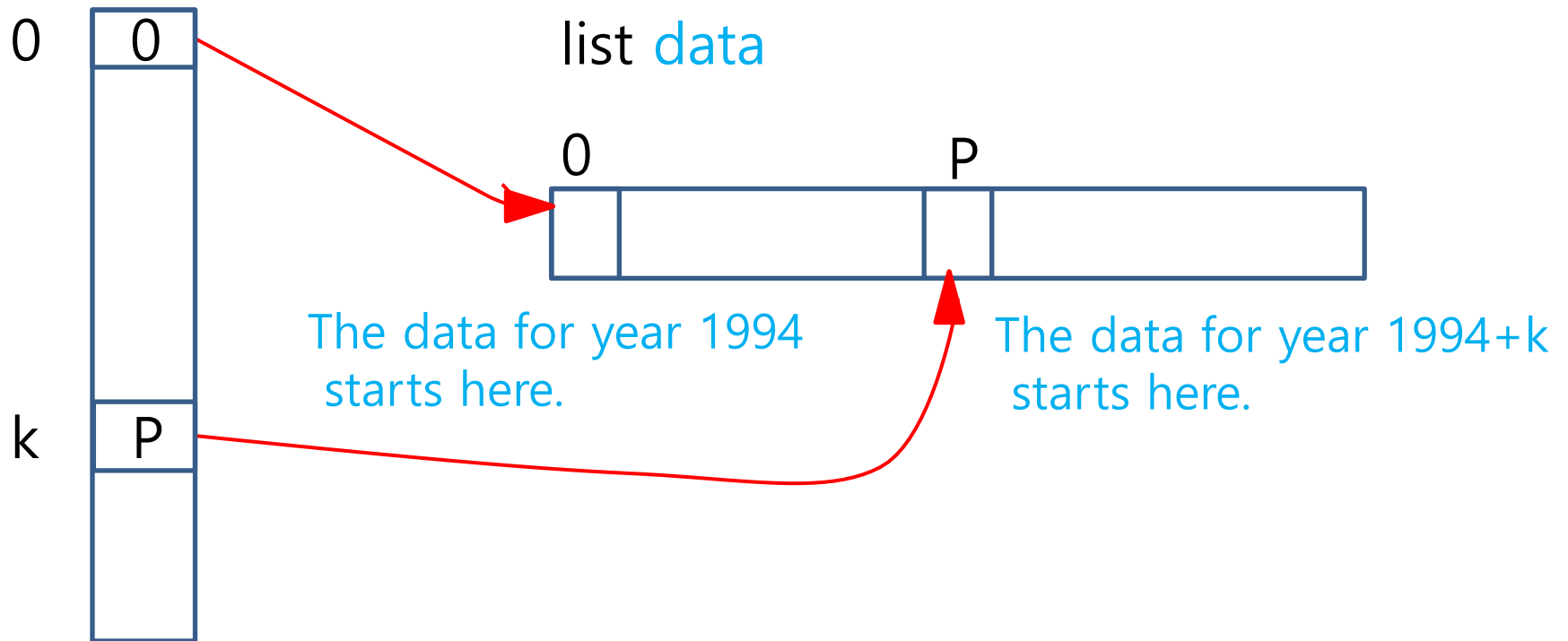
## PROBLEM 5: USING AN INDEX TABLE

For fast access to the data of a given year in the list, we want to add a function to create an **index table** such that **each element** of **the table** points to the **first tuple** of the year corresponding to the index, as shown in the **next slide**. The mapping from indices to years is:

$$\text{index} = \text{year} - 1994.$$

The index table, **inx\_tb** can be used to efficiently **select** the **tuples** of **exchange rates** from the list **data** for a **given year**.

Provided with a series of query years, print each year and its average, minimum, and maximum exchange rates. The query years are given one at a time through the keyboard.



index table  
inx\_tb

---

## How to create the index table

```
def create_ind_tb(data):  
    yr = 1994  
    inx_tb = [0] * 16  
    for i in range(len(data)):  
        date, rate = data[i]  
        year = date / 10000  
        if year != yr:  
            inx_tb[year - 1994] = i  
            yr = year  
    return inx_tb
```