

<p>lec20021003 下平英寿 shimo@is.titech.ac.jp</p>	<p style="text-align: center;">R</p> <ul style="list-style-type: none"> データ操作，統計計算，グラフィックのための統合ソフトウェア環境 行列操作に優れている，データ解析の一貫したツール群，簡単に効率的なプログラミング言語 Rはフリーソフトでソースも公開 (http://cran.r-project.org) Rの開発は1990年代後半からネット上で行われている Rの前身であるSはC言語やUNIXと同じAT&T(現 Lucent Technologies) のベル研究所で1984年ごろ開発(ちなみにC言語およびUNIXの開発は1971年ごろ) 	<p style="text-align: center;">講義内容</p> <p>「データ解析」では統計処理ソフトウェアであるRを用いた多変量解析の実践的な講義を行う。Rの使用方法について簡単に紹介した後、実際にRを使ったデータ解析を行う。単にソフトの使用法を学ぶのではなく、その背後にある数学を自分のものにするのが目標である。回帰分析や主成分分析のためのR関数を自分自身で書き、それを使ってデータ解析を行うのである。この経験は将来未知の問題で新しい手法を開発する場面で役に立つだろう。</p>															
<p style="text-align: center;">講義予定</p> <p>R入門 (1) 端末室での演習1 (2) Rによる線形代数 (3) 回帰分析 (4, 5) 主成分分析 (6, 7) 端末室での演習2 (8) クラスター分析 (9) 正準相関分析 (10) 判別分析 (11) モデル選択 (12) プートストラップ法 (13) 端末室での演習3 (14)</p>	<p style="text-align: center;">講義について</p> <ul style="list-style-type: none"> ホームページ http://www.is.titech.ac.jp/~shimo/class/ 担当：下平 テイナーチングアシスタント：坂口，多賀 評価方法：出席とレポート 質問受け付け：まずテイナーチングアシスタントに質問内容のメールを出すこと。面談が必要な場合はあらかじめメールにてアポイントを取る。こと。もしくは講義か演習時に直接質問する。 	<p style="text-align: center;">eduでのRの利用の準備</p> <ul style="list-style-type: none"> 最新版のRが~/shimo/program/Rにインストールされている。利用するにはまず自分の.cshrcファイルに以下のように書き加える。 <pre>setenv SHIMO ~/shimo/program setenv PATH \${SHIMO}/R/bin:\${PATH}</pre> 以上の設定を有効にするため、source ~/.cshrcを実行する。(次回にログインするときはこの必要ない) 作業用ディレクトリを作る。例えば <pre>shimo@edu[~] mkdir -p class/gakubu200209 shimo@edu[~] chmod og-rx class/gakubu200209</pre> 															
<p style="text-align: center;">eduでのRの利用</p> <ul style="list-style-type: none"> 作業用ディレクトリに移動しRを起動 <pre>shimo@edu[~] cd class/gakubu200209 shimo@edu[gakubu200209] R</pre> たとえば $\sin(x)$ の曲線を描く <pre>> x <- seq(0,10,0.1) > plot(x, sin(x), type="l")</pre> Rの終了は q() コマンド <pre>> q() Save workspace image? [y/n/c]: y</pre> 作業用ディレクトリにファイル.Rdataが自動的に作られる。 	<p style="text-align: center;">emacsでの統合環境 ESS</p> <ul style="list-style-type: none"> .emacsファイルに以下を書き加えてから emacs を再起動。 <pre>;; ess (setq load-path (cons (expand-file-name "~/shimo/program/ess/lisp") load-path)) (autoload 'R "ess-site" "" t) (add-hook 'ess-post-run-hook (function (lambda () (set-buffer-process-coding-system 'euc-japan 'euc-japan))))</pre> emacs内でM-x RによってRを起動する。 セッションはC-x C-wによってxxxx.Rtのような名前でセッションする。 プログラムのファイルはyyyy.Rのような名前で作成しRコマンド source ("yyyy.R") でロード。 RのヘルプはC-c C-vで別ウインドウが開く 	<p style="text-align: center;">講義で用いているデータセット</p> <p>総務庁統計局統計センターが公開している社会・人口統計体系 http://www.stat.go.jp/data/ssds/index.htm</p> <pre>edu:~/shimo/class/gakubu200209/data/X2000.R というファイルでX2000というリスト型の変数を定義している。</pre> <table border="1"> <thead> <tr> <th>データ本体</th> <th>サイズ</th> <th>47×1173の行列</th> </tr> </thead> <tbody> <tr> <td>X2000\$item</td> <td>【項目名】</td> <td>サイズ1173の文字型ベクトル (日本語)</td> </tr> <tr> <td>X2000\$jitem</td> <td>【県名】</td> <td>サイズ47の文字型ベクトル (日本語)</td> </tr> <tr> <td>X2000\$pref</td> <td></td> <td></td> </tr> <tr> <td>X2000\$jpref</td> <td></td> <td></td> </tr> </tbody> </table>	データ本体	サイズ	47×1173の行列	X2000\$item	【項目名】	サイズ1173の文字型ベクトル (日本語)	X2000\$jitem	【県名】	サイズ47の文字型ベクトル (日本語)	X2000\$pref			X2000\$jpref		
データ本体	サイズ	47×1173の行列															
X2000\$item	【項目名】	サイズ1173の文字型ベクトル (日本語)															
X2000\$jitem	【県名】	サイズ47の文字型ベクトル (日本語)															
X2000\$pref																	
X2000\$jpref																	

- 1 A05201 自然増加率
- 2 A06102 一般世帯の平均人員
- 3 A06202 核家族世帯割合
- 4 F01503 共働き世帯割合
- 5 A06205 単独世帯割合
- 6 A06301 65歳以上の親族のいる世帯割合
- 7 A06302 高齢夫婦のみの世帯の割合
- 8 A06304 高齢单身世帯の割合
- 9 A06601 離婚率（人口千人当たり）
- 10 A06602 離婚率（人口千人当たり）

10

11

12

サンプルセッション 1

```

Last login: Fri Sep 20 2002 15:54:22 +0900 from grandma
Sun Microsystems Inc. SunOS 5.8 Generic Patch October 2001
No mail.
Sun Microsystems Inc. SunOS 5.8 Generic Patch October 2001
shimo@edul[~] cd class/gakuh200209
shimo@edul[gakuh200209] R

R : Copyright 2002, The R Development Core Team
Version 1.5.1 Patched (2002-09-08)

```

```

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information.

```

13

> X2000\$item[a]

```

      A05201      A06102      A06202      F01503      ...
"Rate of natural increase "
A06102      "Members per private household "
A06202      "Ratio of family nuclei households "
F01503      "Ratio of dual-income households "
A06205      "Ratio of one-person households "
A06301      "Ratio of households with members65 years old and over "
A06302      "Ratio of aged-couple households "
A06304      "Ratio of aged-single person households "
A06601      "Rate of marriages (per 1,000 persons) "

```

	A05201	A06102	A06202	F01503	...
Hokkaido	0.04	2.42	60.54	26.54	...
Aomori	-0.02	2.86	54.20	34.38	...
Iwate	-0.07	2.92	50.87	38.82	...
Miyagi	0.18	2.80	51.86	31.88	...
Akita	-0.25	3.00	50.48	39.55	...
Yamagata	-0.12	3.25	45.79	47.09	...
Fukushima	0.06	3.05	52.12	39.53	...
Ibaraki	0.16	2.99	58.28	35.84	...
Tochigi	0.13	2.97	56.47	37.59	...
Gumma	0.12	2.88	60.07	36.35	...
Saitama	0.36	2.78	65.46	30.89	...
Chiba	0.27	2.70	62.55	29.57	...
Tokyo	0.11	2.21	52.15	22.09	...
Kanagawa	0.36	2.53	62.04	25.04	...
Okinawa	0.67	2.91	64.54	25.70	...

```

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for a HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

```

> source("shimo/class/gakuh200209/data/X2000.R")

```

> names(X2000)
[1] "x"      "pref"   "code"   "item"   "club"   "jpref"  "jitem"  "junit"
[10] "jcate" "xm"
> dim(X2000$x)
[1] 47 1173
> X2000$pref
[1] "Hokkaido" "Aomori" "Iwate" "Miyagi" "Akita" "Yamagata"
[7] "Fukushima" "Ibaraki" "Tochigi" "Gumma" "Saitama" "Chiba"
[13] "Tokyo" "Kanagawa" "Miyagi" "Tochigi" "Ishikawa" "Fukui"
[19] "Yamanashi" "Nagano" "Gifu" "Shizuoka" "Aichi" "Mie"
[25] "Shiga" "Kyoto" "Osaka" "Hyogo" "Nara" "Wakayama"
[31] "Tottori" "Shimane" "Okayama" "Hiroshima" "Yamaguchi" "Tokushima"

```

> X2000\$item[a]

```

      A05201      A06102      A06202      F01503      ...
"Rate of divorces (per 1,000 persons) "
A06102      "自然増加率 "
A06202      "一般世帯の平均人員 "
F01503      "核家族世帯割合 "
A06205      "共働き世帯割合 "
A06301      "単独世帯割合 "
A06302      "65歳以上の親族のいる世帯割合 "
A06304      "高齢夫婦のみの世帯の割合 "
A06601      "高齢单身世帯の割合 "
A06602      "離婚率（人口千人当たり） "
> na <- paste(seq(along=a),a,X2000$item[a])
> na
[1] "1 A05201 Rate of natural increase "
[2] "2 A06102 Members per private household "
[3] "3 A06202 Ratio of family nuclei households "
[4] "4 F01503 Ratio of dual-income households "

```

西7号館端末室(edu)で演習する。セッションファイルと同じことを自分でやってみる。詳しい意味は分からなくても良い。データ解析とRについて慣れることが目的。

●サンプルセッション1

Rによるデータ解析1（主成分分析，クラスターング，重回帰分析）
 ファイル： "shimo/class/gakuh200209/rote20020919a.Rt"

●サンプルセッション2

Rによるデータ解析2（ランダムにデータ項目を選ぶ）
 ファイル： "shimo/class/gakuh200209/rote20020919b.Rt"

●サンプルセッション3

R言語入門
 ファイル： "shimo/class/gakuh200209/rote20020919c.Rt"

```

[37] "Kagawa" "Ehime" "Kochi" "Fukuoka" "Saga" "Nagasaki"
[43] "Kumamoto" "Oita" "Miyazaki" "Kagoshima" "Okinawa"
> X2000$jpref
Hokkaido Aomori Iwate Miyagi Akita Yamagata Fukushima
"北海道" "青森県" "岩手県" "宮城県" "秋田県" "山形県" "福島県"
Ibaraki Tochigi Saitama Chiba Tokyo Kanagawa
"茨城県" "栃木県" "群馬県" "埼玉県" "千葉県" "東京都" "神奈川県"
Miyagi Toyama Ishikawa Fukui Yamanashi Nagano Gifu
"新潟県" "富山県" "石川県" "福井県" "山梨県" "長野県" "岐阜県"
Shizuoka Aichi Mie Shiga Kyoto Osaka Hyogo
"静岡県" "愛知県" "三重県" "滋賀県" "京都府" "大阪府" "兵庫県"
Nara Wakayama Tottori Shimane Okayama Hiroshima Yamaguchi
"奈良県" "和歌山県" "鳥取県" "島根県" "岡山県" "広島県" "山口県"
Tokushima Kagawa Ehime Kochi Fukuoka Saga Nagasaki
"徳島県" "香川県" "愛媛県" "高知県" "福岡県" "佐賀県" "長崎県"
Kumamoto Oita Miyazaki Kagoshima Okinawa
"熊本県" "大分県" "宮崎県" "鹿児島県" "沖縄県"
> ## get some items
> a <- c("A05201","A06102","A06202","F01503","A06205","A06301","A06302",

```

```

[5] "5 A06205 Ratio of one-person households "
[6] "6 A06301 Ratio of households with members65 years old and over "
[7] "7 A06302 Ratio of aged-couple households "
[8] "8 A06304 Ratio of aged-single person households "
[9] "9 A06601 Rate of marriages (per 1,000 persons) "
[10] "10 A06602 Rate of divorces (per 1,000 persons) "
> jna <- paste(seq(along=a),a,X2000$item[a])
> jna
[1] "1 A05201 自然増加率 "
[2] "2 A06102 一般世帯の平均人員 "
[3] "3 A06202 核家族世帯割合 "
[4] "4 F01503 共働き世帯割合 "
[5] "5 A06205 単独世帯割合 "
[6] "6 A06301 65歳以上の親族のいる世帯割合 "
[7] "7 A06302 高齢夫婦のみの世帯の割合 "
[8] "8 A06304 高齢单身世帯の割合 "
[9] "9 A06601 離婚率（人口千人当たり） "
[10] "10 A06602 離婚率（人口千人当たり） "
> x <- X2000$x[,a]

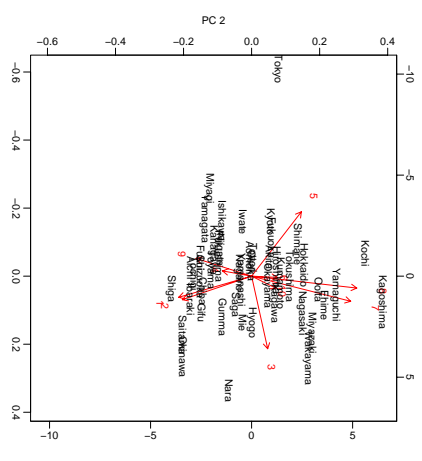
```

	> x
A05201	0.04
A06102	2.42
A06202	60.54
F01503	26.54
A06205	29.95
A06301	30.50
A06302	9.90
A06304	7.39
A06601	5.77
A066	
Hokkaido	0.04
Aomori	-0.02
Iwate	-0.07
Miyagi	0.18
Aita	-0.25
Yamagata	-0.12
Fukushima	0.06
Ibaraki	0.16
Tochigi	0.13
Gunma	0.32
Saitama	0.36
Chiba	0.27
Tokyo	0.11
Kanagawa	0.36
Niigata	-0.03
Toyama	-0.01
Tshikawa	0.12

主成分分析

主成分分析 = PCA (principal component analysis)

- 多変量解析の定番
- データの変動をできるだけ少数個の変換された変量で説明
- 主成分 = 変換された変量
- バイオロットによって結果を視覚的に示す



Fukui	0.13	3.14	50.72	46.69	20.94	43.41	7.67	5.73	5.48	1
Yamanashi	0.08	2.84	57.42	36.04	24.17	38.14	8.61	6.19	5.70	1
Nagano	0.06	2.89	55.17	41.54	23.13	41.74	9.41	6.03	5.70	1
Gifu	0.15	3.07	56.74	39.64	19.74	38.46	7.79	5.09	5.59	1
Shizuoka	0.18	2.91	56.87	38.15	22.91	35.33	6.89	4.83	5.94	1
Aichi	0.40	2.75	59.79	32.91	26.23	28.08	6.66	4.89	6.59	1
Mie	0.35	2.88	59.40	36.87	21.73	37.49	9.17	6.65	5.72	1
Shiga	0.31	3.02	57.50	36.76	22.22	33.67	6.65	4.48	6.24	1
Kyoto	0.12	2.55	57.65	27.97	30.86	31.42	8.10	7.49	5.96	1
Osaka	0.30	2.51	61.90	23.78	29.78	27.21	7.30	7.38	6.68	2
Hyogo	0.22	2.69	63.21	26.94	24.95	32.23	8.54	7.43	6.23	2
Nara	0.18	2.93	64.94	27.21	19.13	33.77	8.31	5.94	5.35	1
Makayama	-0.06	2.77	61.98	30.71	21.97	41.08	10.63	9.54	5.35	1
Tottori	-0.08	3.00	51.83	43.23	22.69	44.73	8.44	7.33	5.24	1
Shimane	-0.24	2.90	50.97	42.89	24.02	48.06	10.46	8.24	4.67	1
Okyama	0.07	2.77	57.42	35.14	24.98	37.73	9.70	7.27	5.90	1
Hiroshima	0.12	2.57	59.94	32.03	28.02	32.87	9.36	7.73	5.60	1
Yamaguchi	-0.15	2.56	59.83	32.46	26.75	39.33	11.35	9.65	5.15	1
Tokushima	-0.11	2.78	55.68	35.88	24.40	40.95	9.61	8.08	5.07	1

PCAの関数

```

mypca <- function(dat) {
  s <- mysvd(dat)
  x <- s$u %**% diag(s$d)
  y <- s$v %**% diag(s$d)
  list(x=x, y=y, d=s$d, u=s$u, v=s$v)
}
> p <- mypca(scale(x))
> mybiplot(p$u, p$v)
> mybiplot(p$u, p$v, choi=3:2)

```

クラスター分析

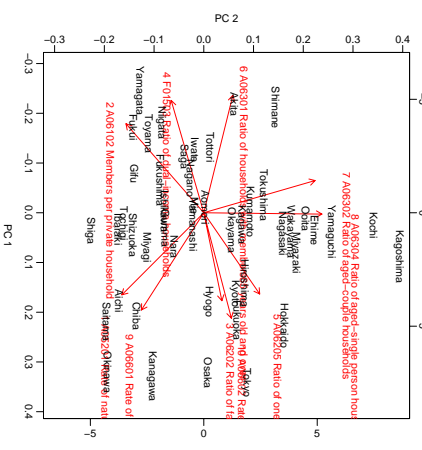
- 対象の自動分類「教師なし」の分類
- 特徴ベクトルなどから対象間の「距離」を定義
- 距離に従って、似たもの同士の群（クラスター）に分類
- 階層的クラスタリング、樹状図

Kagawa	0.03	2.75	58.53	36.35	23.81	38.88	9.95	7.59	5.63	1
Ehime	-0.08	2.59	60.33	31.24	26.30	38.15	11.03	9.06	5.12	1
Kochi	-0.25	2.47	57.67	33.12	29.85	40.21	10.98	11.16	5.00	2
Fukuoka	0.14	2.57	57.86	26.51	30.24	31.10	7.88	7.48	5.94	2
Saga	0.07	3.08	55.06	39.49	20.96	42.83	8.27	6.99	5.19	1
Nagasaki	0.02	2.71	59.92	31.00	26.30	39.10	9.84	9.18	5.12	1
Kumamoto	0.02	2.81	56.19	34.98	26.04	40.22	9.52	8.93	5.15	1
Oita	-0.06	2.64	58.01	32.47	26.42	39.43	10.79	8.93	5.08	1
Miyazaki	0.07	2.61	62.18	35.51	25.74	36.93	11.13	9.11	5.32	2
Kagoshima	-0.13	2.43	62.44	29.85	30.12	38.01	12.66	12.39	5.09	1
Okinawa	0.67	2.91	64.54	25.70	24.26	27.92	5.39	6.22	6.47	2

```

[1] 47 10
> # load "mybiplot", "mypca", "myplot", "mysvd", "psint"
> source("shimo/class/gakubu200209/myfunc20020919.R")

```

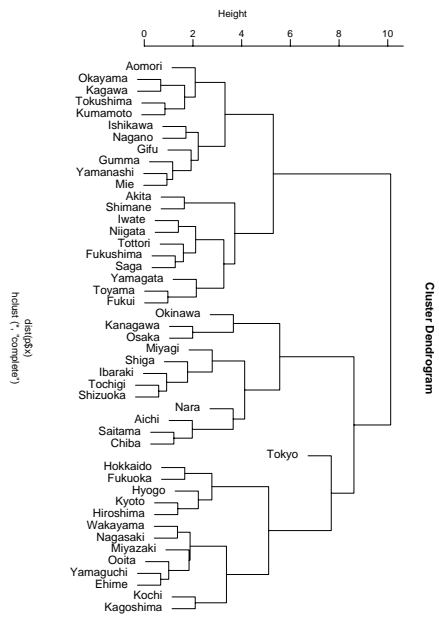


Rでのクラスター分析

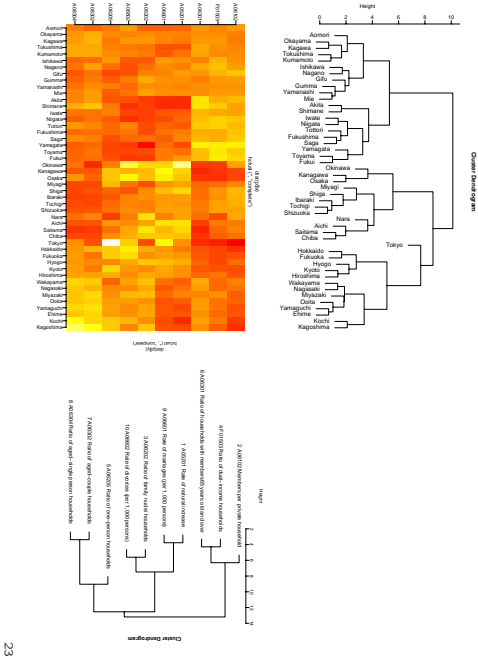
```

library(mva) # load multivariate analysis library
> hx <- hclust(dist(p$x))
> hy <- hclust(dist(p$y))
> # show clustering
> plot(hx, cex=1.2)
> plot(hy, labels=na, cex=0.9)
> # show matrix
> image(seq(along=hx$order), seq(along=hy$order),
+ scale(x) [hx$order, rev(hy$order)]),
+ col=heat.colors(60), axes=F, ylab="")
> axis(1, seq(along=hx$order), rownames(p$x) [hx$order], las=2)
> axis(2, seq(along=hy$order), rownames(p$y) [rev(hy$order)]], las=2)

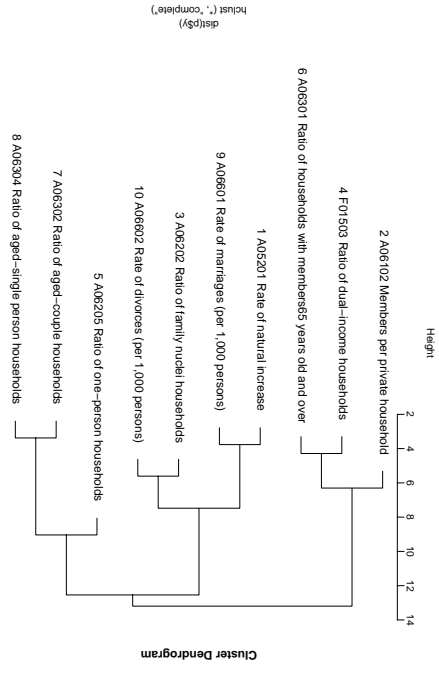
```



20



23



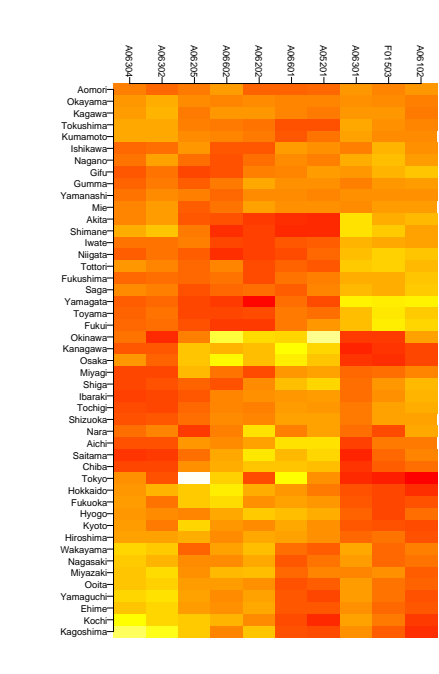
21

回帰分析

- データ解析で最も利用頻度の高い分析法
- 重回帰 = multiple regression
- 従属変数 y と独立変数 x_1, x_2, \dots, x_m の関係を推定

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_m x_m + \epsilon$$
- 係数 β_i が y と x_i の関係を表す

24



22

重回帰分析の関数

```

mysfit <- function(x,y) {
  ...
  sx <- mysvd(x)
  coef <- sx$y %*% (1/sx$d * (t(sx$u) %*% y))
  ...
  ypred <- x %*% coef
  resid <- y - ypred
  ...
  list(coefficients=coef, residuals=resid, pred=ypred, ...)
}
> f <- mysfit(x[, -1], x[, 1])
> myplot(f$pred, x[, 1], xlab="prediction", ylab="na[1]", cex=0.8)
> abline(0, 1, col=2, lwd=0.5)

```

25

サンプリング

```

> round(f$tsummary, 3)
, , = Y
Estimate Std. Err t-value Pr(>|t|)
Intercept -7.887 1.334 -5.913 0.000
A06102 1.691 0.192 8.808 0.000
A06202 0.024 0.010 2.553 0.015
F01503 0.003 0.003 0.978 0.334
A06205 0.056 0.013 4.359 0.000
A06301 -0.022 0.009 -2.335 0.025
A06302 0.051 0.017 2.992 0.005
A06304 -0.010 0.016 -0.599 0.553
A06601 0.107 0.027 3.984 0.000
A06602 0.108 0.043 2.521 0.016

```

26

27

```

> ## randomly selecting items
> a0 <- X2000$code[grepl(".", X2000$clab)] # indices
> a0
[1] "A01101" "A01601" "A0160101" "A0160102" "A0160103" "A01201"
[7] "A01202" "A01302" "A01401" "A01402" "A02101" "A02102"
[13] "A02103" "A02104" "A03101" "A03102" "A03103" "A03401"
[19] "A03402" "A03403" "A03404" "A04101" "A04102" "A04103"
[25] "A04104" "A04105" "A04106" "A04107" "A04108" "A04109"
[31] "A04110" "A04111" "A04112" "A04113" "A04114" "A04115"
[37] "A04116" "A04117" "A04118" "A04119" "A04120" "A04121"
[43] "A05201" "A05202" "A05203" "A05204" "A05205" "A05206"
[49] "A05207" "A05208" "A05209" "A05210" "A05211" "A05212"
[55] "A05213" "A05214" "A05215" "A05216" "A05217" "A05218"
[61] "A05219" "A05220" "A05221" "A05222" "A05223" "A05224"
[67] "A05225" "A05226" "A05227" "A05228" "A05229" "A05230"
[73] "A05231" "A05232" "A05233" "A05234" "A05235" "A05236"
[79] "A05237" "A05238" "A05239" "A05240" "A05241" "A05242"

```

28

```

[85] "C01101" "C01105" "C01106" "C01107" "C02102" "C02103"
[91] "C02201" "C02205" "C03201" "C03205" "C03303" "C03303"
[97] "C03304" "C03408" "C04101" "C04105" "C04106" "C04107"
[103] "C04401" "C04404" "C04505" "C04507" "C04601" "C04601"
[109] "C04602" "C04201" "C04203" "C04101" "C04102" "C04104"
[115] "L04105" "L04106" "L04107" "L04108" "L04109" "L04110"
[121] "L04111" "L04112" "L04113" "L04302" "L04304" "L04110"
[127] "D01102" "D01201" "D01302" "D01401" "D01402" "D01403"
[133] "D02101" "D02102" "D02103" "D02201" "D02202" "D02204"
[139] "D02206" "D02207" "D03103" "D03104" "D03105" "D03106"
[145] "D03107" "D03108" "D03109" "D03110" "D03111" "D03112"
[151] "D03113" "D03114" "D03115" "D0313" "D03201" "D03202"
[157] "D03203" "D03301" "D03302" "D03303" "D03304" "D03305"
[163] "D03306" "D03307" "D03308" "D03309" "D0331" "D03313"
[169] "D03320" "D03321" "D03322" "D03323" "D03324" "D03325"
[175] "D03319" "E01101" "E01102" "E01103" "E01104" "E01105"
[181] "E01106" "E01107" "E01108" "E01109" "E01110" "E01111"
[187] "E02101" "E02102" "E02103" "E02104" "E02105" "E02106"
[193] "E02701" "E02702" "E02703" "E04102" "E04103" "E05103"

```

```

[199] "E05103" "E05104" "E05105" "E05106" "E05107" "E05108"
[205] "E06205" "E06101" "E06102" "E06103" "E06104" "E06105"
[211] "E06204" "E06205" "E06206" "E06207" "E06208" "E06209"
[217] "E09101" "E09102" "E09103" "E09104" "E09105" "E09106"
[223] "E09211" "E09212" "E09213" "E09214" "E09215" "E09216"
[229] "E09501" "E09502" "E09503" "E09504" "E09505" "E09506"
[235] "E09601" "E09602" "E09603" "E09604" "E09605" "E09606"
[241] "E10101" "E10102" "E10103" "E10104" "E10105" "E10106"
[247] "E10201" "E10202" "E10203" "E10204" "E10205" "E10206"
[253] "E10301" "E10302" "E10303" "E10304" "E10305" "E10306"
[259] "E10401" "E10402" "E10403" "E10404" "E10405" "E10406"
[265] "E10501" "E10502" "E10503" "E10504" "E10505" "E10506"
[271] "E10601" "E10602" "E10603" "E10604" "E10605" "E10606"
[277] "E10701" "E10702" "E10703" "E10704" "E10705" "E10706"
[283] "E10801" "E10802" "E10803" "E10804" "E10805" "E10806"
[289] "E10901" "E10902" "E10903" "E10904" "E10905" "E10906"
[295] "E11001" "E11002" "E11003" "E11004" "E11005" "E11006"
[301] "E11101" "E11102" "E11103" "E11104" "E11105" "E11106"
[307] "E11201" "E11202" "E11203" "E11204" "E11205" "E11206"

```

```

[313] "H02101" "H02102" "H02103" "H02104" "H02105" "H02106"
[319] "H02302" "H02303" "H02601" "H04101" "H04102" "H04301"
[325] "H05102" "H05201" "H05304" "H05306" "H05401" "H05503"
[331] "H05504" "H05601" "H06103" "H06105" "H06106" "H06107"
[337] "H06109" "H06113" "H06114" "H06115" "H06116" "H06117"
[343] "H06119" "H06121" "H06302" "H06309" "H06305" "H06306"
[349] "H06307" "H06401" "H06402" "H06403" "H06404" "H06405"
[355] "H06501" "H06502" "H06503" "H06504" "H06505" "H06506"
[361] "H07201" "H07202" "H07203" "H07204" "H07205" "H07206"
[367] "H08301" "H08302" "H08303" "H08304" "H08305" "H08306"
[373] "I04102" "I04103" "I04201" "I04202" "I04203" "I04204"
[379] "I05101" "I05201" "I05202" "I05203" "I05204" "I05205"
[385] "I05206" "I06101" "I06102" "I06103" "I06104" "I06105"
[391] "I06106" "I06201" "I07101" "I07102" "I07103" "I07104"
[397] "I07105" "I07201" "I02101" "I02102" "I02103" "I02104"
[403] "I02105" "I02106" "I02107" "I02108" "I02109" "I02110"
[409] "I02111" "I02112" "I02113" "I02114" "I02115" "I02116"
[415] "I09101" "I09102" "I09103" "I09104" "I09105" "I09106"
[421] "I09301" "I09401" "I09402" "I09501" "I09502" "I09503"

```

```

[427] "I10101" "I10102" "I10103" "I10104" "I10105" "I10201"
[433] "I10202" "I10203" "I10204" "I10205" "I11101" "I1102"
[439] "I11201" "I11202" "I11203" "I11204" "I11205" "I11206"
[445] "I14101" "I14102" "I14201" "I14202" "I01101" "I01107"
[451] "J01108" "J01109" "J01110" "J01111" "J01112" "J01113"
[457] "J02101" "J02102" "J02103" "J02104" "J02105" "J02106"
[463] "J03101" "J03102" "J03103" "J03104" "J03105" "J03106"
[469] "J03501" "J04101" "J04102" "J04103" "J04104" "J04105"
[475] "J04201" "J04301" "J04302" "J04401" "J04402" "J05101"
[481] "J05102" "J05103" "J05104" "J05105" "J05106" "J05107"
[487] "J05201" "J05202" "J05203" "J05204" "J05205" "J05206"
[493] "J05102" "J05103" "J05104" "J05105" "J05106" "J05107"
[499] "J05204" "J05205" "J05206" "J05207" "J05208" "J05209"
[505] "J05210" "J05211" "J05212" "J05213" "J05214" "J05215"
[511] "K01302" "K01401" "K01402" "K01403" "K01404" "K01405"
[517] "K02205" "K02301" "K02302" "K02303" "K02304" "K02305"
[523] "K03112" "K04102" "K04103" "K04104" "K04105" "K04106"
[529] "K04201" "K04202" "K04301" "K05102" "K05103" "K06101"
[535] "K06104" "K06201" "K06204" "K06401" "K06402" "K06403"

```

```

[541] "K06405" "K06301" "K06304" "K06501" "K06503" "K07105"
[547] "K08101" "K10101" "K10105" "K10107" "K10201" "K10210"
[553] "K10203" "K10301" "K10304" "K10305" "K10401" "K10403"
[559] "K10501" "K10502" "K10503" "K10504" "L01201" "L02602"
[565] "L02201" "L02401" "L02402" "L02403" "L02404" "L02405"
[571] "L02406" "L02407" "L02408" "L02409" "L02410" "L02411"
[577] "L02735" "L01100" "L01101" "L01102" "L01103" "L02601"
[583] "L02101" "L02301" "L03212" "L03213" "L03214" "L03215"
[589] "L03401" "L03402" "L03602" "L03603" "L03604" "L03606"
[595] "L03607" "M01101" "M01201" "M01202" "M01203" "M01204"
[601] "M01205" "M01301" "M01302" "M01303" "M01304" "M01305"
[607] "M02101" "M03101" "M03102" "M03103" "M03104" "M03105"
[613] "M03201" "M03301" "M03302" "M03303" "M03304" "M03305"

```

```

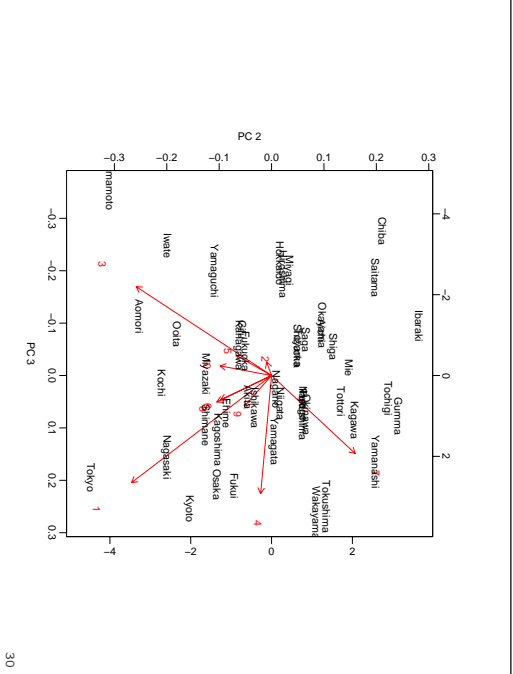
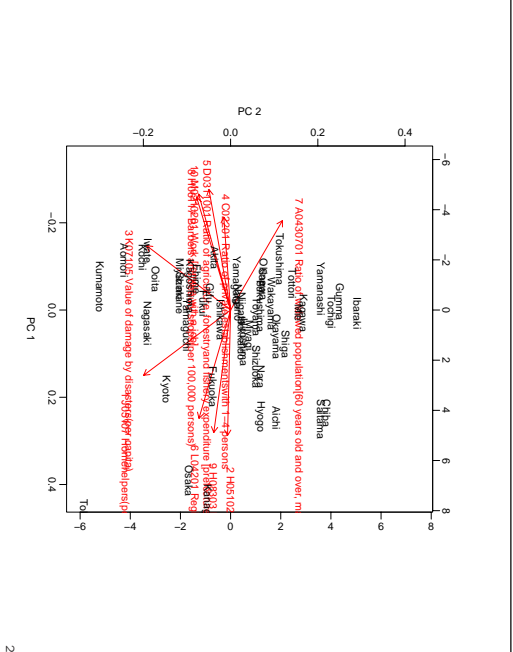
[1] "1 J05107 Homelopers(per 100,000 persons) "
[2] "2 H05102 Ratio of households covered by city gas supply system "
[3] "3 K07105 Value of damage by disasters(per capita) "
[4] "4 C02201 Ratio of private establishments with 1-4 persons "
[5] "5 D031101 Ratio of agriculture, forestry and fishery expenditure [prefecture level] "
[6] "6 L04201 Regional difference index of consumer prices [general : ku-area] "
[7] "7 A0430701 Ratio of widowed population[60 years old and over, male] "
[8] "8 H06117 Barbers and beauty shops(per 100,000 persons) "
[9] "9 H08303 Neighborhood parks per inhabitant area 100k "
[10] "10 M0210201 Work [female with a job] "

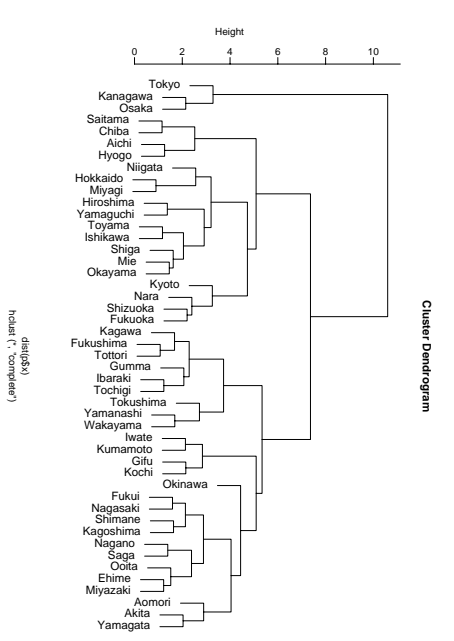
```

```

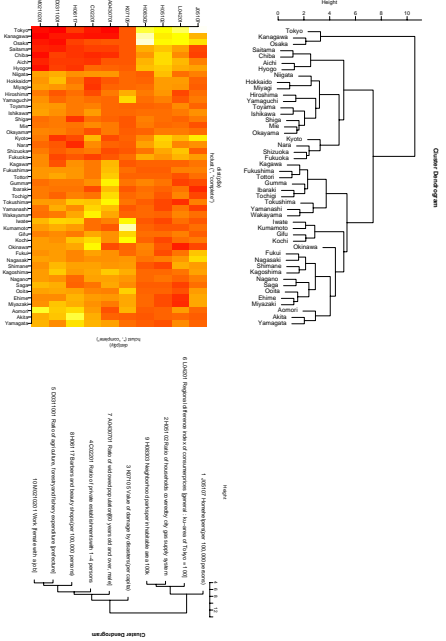
[4] "4 C02201 従業者1-4人の事業所割合[民営] "
[5] "5 D031101 農林水産業割合[農林] "
[6] "6 L04201 消費者物価地域差指数[総合：東京都都区部=100] "
[7] "7 A0430701 死別者割合[60歳以上・男] "
[8] "8 H06117 理容・美容所数(人口10万人当たり) "
[9] "9 H08303 近隣公園数(可住地面積100k当たり) "
[10] "10 M0210201 仕事の平均時間[有業者・女] "

```

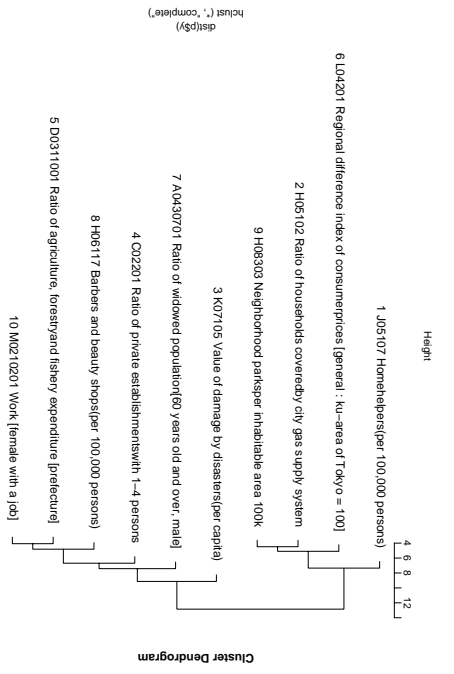




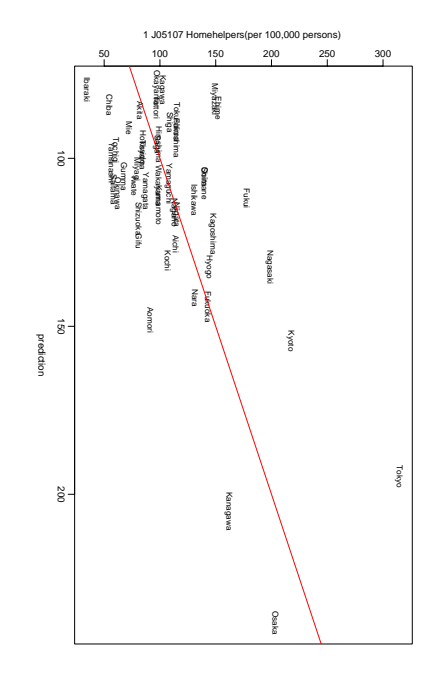
Cluster Dendrogram (detail)



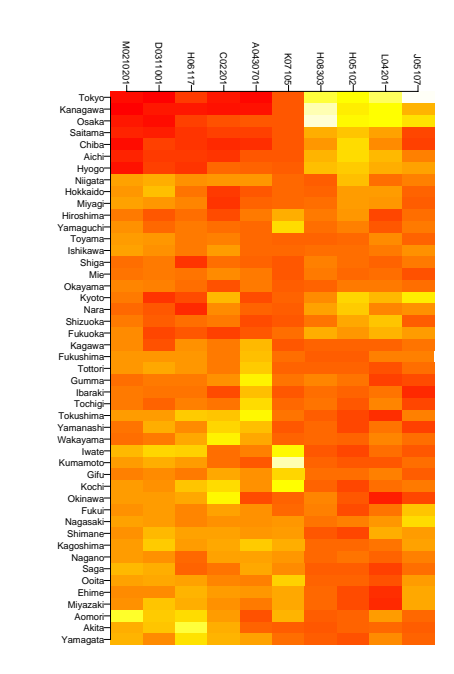
Cluster Dendrogram



Cluster Dendrogram



Cluster Dendrogram



Cluster Dendrogram

サンプルセッション3

```
> ## values
> # character, numeric, and NA types
> "hello"
[1] "hello"
> 123/456
[1] 0.2697368
> sqrt(3/4)
[1] 0.8660254
> sin(pi/3)
[1] 0.8660254
> 1=2
[1] FALSE
> 1=1
[1] TRUE
> 1/0 # infinity
[1] Inf
> sqrt(-1) # not a number
```

```
[1] NaN
Warning message:
NaNs produced in: sqrt(-1)
> sin(NA) # not available
[1] NA
> # vector
> c(1,3,2,4)
[1] 1 3 2 4
> 10:20
[1] 10 11 12 13 14 15 16 17 18 19 20
> 30:20
[1] 30 29 28 27 26 25 24 23 22 21 20
> seq(0,10,0.1) # sequence from 0 to 10 increasing by 0.1
[1] 0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0 1.1 1.2 1.3
[16] 1.5 1.6 1.7 1.8 1.9 2.0 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8
[31] 3.0 3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.8 3.9 4.0 4.1 4.2 4.3
[46] 4.5 4.6 4.7 4.8 4.9 5.0 5.1 5.2 5.3 5.4 5.5 5.6 5.7 5.8
[61] 6.0 6.1 6.2 6.3 6.4 6.5 6.6 6.7 6.8 6.9 7.0 7.1 7.2 7.3
[76] 7.5 7.6 7.7 7.8 7.9 8.0 8.1 8.2 8.3 8.4 8.5 8.6 8.7 8.8
```

```
[91] 9.0 9.1 9.2 9.3 9.4 9.5 9.6 9.7 9.8 9.9 10.0
> a <- 10:20
> length(a) # length of vector
[1] 11
> seq(along=a)
[1] 1 2 3 4 5 6 7 8 9 10 11
> seq(from=1,to=length(a))
[1] 1 2 3 4 5 6 7 8 9 10 11
> 1:length(a)
[1] 1 2 3 4 5 6 7 8 9 10 11
> help(seq) # show the help message for the function "seq"
> rep(3,10) # replicate elements
[1] 3 3 3 3 3 3 3 3 3 3
> rep(c(3,4),c(10,20))
[1] 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
> rep(abc,1:4)
[1] "abc" "def" "ghi" "ghi" "hello" "hello" "hello" "hello"
[10] "hello"
> help(rep)
```

```
> round(f$summary,3)
, , = Y
Estimate Std. Err t-value Pr(>|t|)
Intercept -1092.573 522.218 -2.092 0.043
H05102 0.458 0.600 0.764 0.450
K07105 0.001 0.000 1.259 0.216
C02201 4.757 3.484 1.365 0.180
D0311001 0.411 2.963 0.139 0.891
L04201 5.493 4.361 1.259 0.216
A0430701 6.486 13.400 0.484 0.631
H06117 -0.048 0.170 -0.283 0.778
H08303 7.267 2.901 2.505 0.017
M0210201 53.139 40.318 1.318 0.196
```

Cluster Dendrogram

```

> ## variables
> ## simple assignment
> a <- 1
> a
[1] 1
> a <- 1:10
> a
[1] 1 2 3 4 5 6 7 8 9 10
> a^2
[1] 1 4 9 16 25 36 49 64 81 100
> b <- a*a
> b
[1] 1 4 9 16 25 36 49 64 81 100

```

```

> a <- seq(0,10,0.1)
> plot(a,sqrt(a))
> plot(a,sqrt(a),type="l")

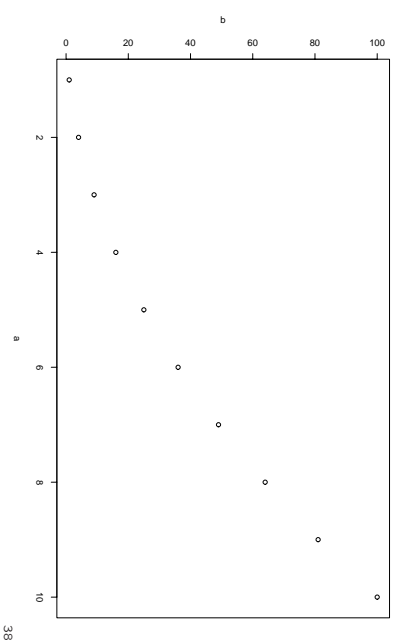
```

```

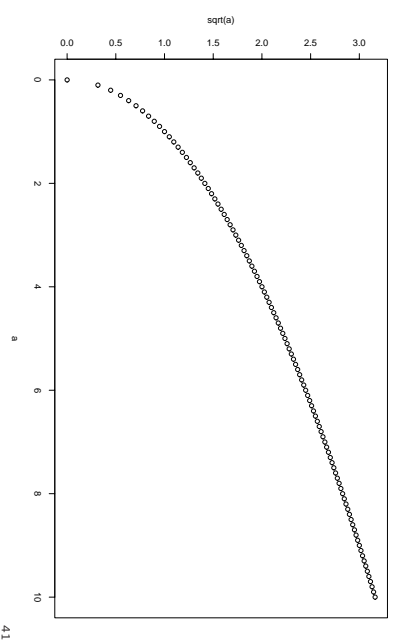
> abc <- c("abc", "def", "ghi", "hello")
> abc
[1] "abc" "def" "ghi" "hello"
> ## output postscript file
> options(paperize="a4")
> a <- 1:10
> b <- a*a
> postscript("ex01prt1.eps")
> plot(a,b)
> dev.off()
X11
2
> postscript("ex01prt2.eps")
> plot(a,sqrt(a))
> dev.off()
X11
2
> a <- seq(0,10,0.1)
> plot(a,sqrt(a))

```

```
> plot(a,b)
```



```
> plot(a,sqrt(a))
```

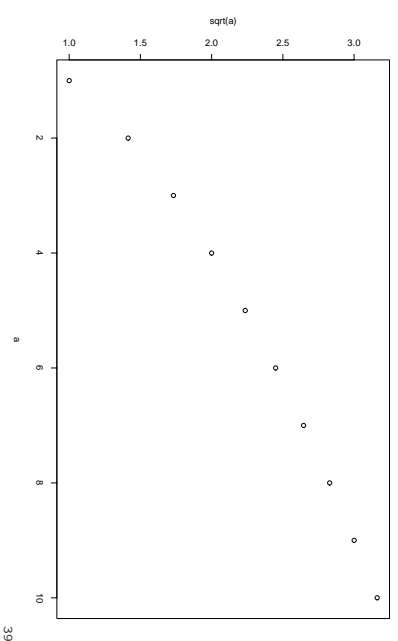


```

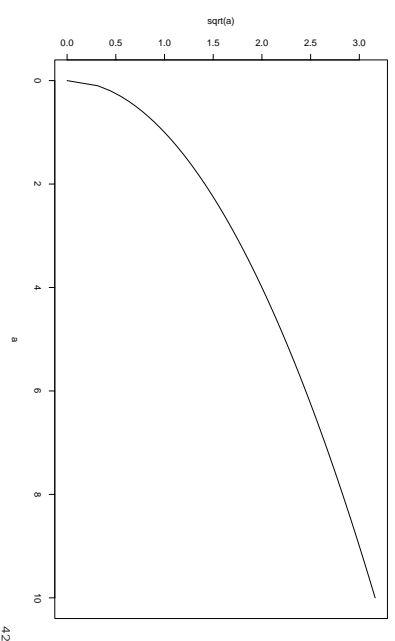
> postscript("ex01prt3.eps")
> plot(a,sqrt(a))
> dev.off()
X11
2
> postscript("ex01prt4.eps")
> plot(a,sqrt(a),type="l")
> dev.off()
X11
2
> ## extract elements
> abc
[1] "abc" "def" "ghi" "hello"
> abc[4]
[1] "hello"
> abc[1:3]
[1] "abc" "def" "ghi"
> abc[-1]
[1] "def" "ghi" "hello"

```

```
> plot(a,sqrt(a))
```



```
> plot(a,sqrt(a),type="l")
```



```

> abc[5]
[1] NA
> a <- 101:110
> a
[1] 101 102 103 104 105 106 107 108 109 110
> a[1]
[1] 101
> a[10]
[1] 110
> a[11]
[1] NA
> a[-1]
[1] 102 103 104 105 106 107 108 109 110
> a[-(1:5)]
[1] 106 107 108 109 110
> a %% 2 # modulo by 2
[1] 1 0 1 0 1 0 1 0 1 0
> a %/ 2 == 0 # even numbers
[1] FALSE TRUE FALSE TRUE FALSE TRUE FALSE TRUE

```

```

> a[a%%2==0]
[1] 102 104 106 108 110
> a[a%%2==1]
[1] 101 103 105 107 109
> a>= 105
[1] FALSE FALSE FALSE TRUE TRUE TRUE TRUE
> a[a>=105]
[1] 105 106 107 108 109 110
> ## matrix
> a <- matrix(1:6,3,2) # matrix of size 3 x 2 with elements 1:6
[1,] 1 4
[2,] 2 5
[3,] 3 6
> dim(a) # dimensions
[1] 3 2
> t(a) # matrix transpose
[1,] [2,] [3,]

```

```

[1,] 1 2 3
[2,] 4 5 6
> t(a) %*% a # multiplication
[1,] 14 32
[2,] 32 77
> a %*% t(a)
[1,] [2,] [3,]
[1,] 17 22 27
[2,] 22 29 36
[3,] 27 36 45
> a <- matrix(1:4,2,2) # matrix of size 2 x 2 with elements 1:4
> a
[1,] [2,]
[1,] 1 3
[2,] 2 4
> b <- solve(a) # inverse matrix
[1,] [2,]

```

```

[1,] -2 1.5
[2,] 1 -0.5
> a %*% b # should be identity matrix
[1,]
[1,] 1.000000e+00 -5.551115e-17
[2,] 4.440892e-16 1.000000e-00
> round(a %*% b, 6) # show results with 6 digits
[1,] [2,]
[1,] 1 0
[2,] 0 1
> b %*% a
[1,] [2,]
[1,] 1 4.440892e-16
[2,] 0 1.000000e-00
> a * a # element-wise multiplication
[1,] [2,]
[1,] 1 9
[2,] 4 16
> a %*% a # matrix multiplication

```

```

[1,] [2,]
[1,] 7 15
[2,] 10 22
> sin(a) # element-wise sin()
[1,]
[1,] 0.8414710 0.1411200
[2,] 0.9092974 -0.7568025
> a %/ 2 == 0
[1,] [2,]
[1,] FALSE FALSE
[2,] TRUE TRUE
> a[a %/ 2 == 0] # result becomes a vector
[1] 2 4
> ## more on matrix
> a <- matrix(1:20,4) # matrix of size 4 x 5
[1,] 1 5 9 13 17
[2,] 2 6 10 14 18

```

```

[3,] 3 7 11 15 19
[4,] 4 8 12 16 20
> a[3,] # the third row
[1] 3 7 11 15 19
> a[,3] # the third column
[1] 9 10 11 12
> a[2:3,] # the second and third rows
[1,] [2,] [3,] [4,] [5]
[1,] 2 6 10 14 18
[2,] 3 7 11 15 19
> a[,2:3] # the second and third columns
[1,] [2,]
[1,] 5 9
[2,] 6 10
[3,] 7 11
[4,] 8 12
> a[3,drop=F] # the third row as matrix
[1,] [2,] [3,] [4,] [5]
[1,] 3 7 11 15 19

```

```

> a[,3,drop=F] # the third column as matrix
[1,]
[1,] 9
[2,] 10
[3,] 11
[4,] 12
> ## names and dimnames
> ## vector
> a <- 10:15
[1] 10 11 12 13 14 15
> names(a) <- c("ten", "eleven", "twelve", "thirteen", "fourteen", "fifteen")
> a
ten eleven twelve thirteen fourteen fifteen
10 11 12 13 14 15
> names(a)
[1] "ten" "eleven" "twelve" "thirteen" "fourteen" "fifteen"
> a[2]
eleven

```

```

11
> a["eleven"]
eleven
11
> a[2:3]
eleven twelve
> a[c("eleven", "twelve")]
eleven twelve
11 12
> b <- names(a)
> b[>=13]
[1] "thirteen" "fourteen" "fifteen"
> b[a%%2==0]
[1] "ten" "twelve" "fourteen"
> ## matrix
> a <- matrix(1:6,3) # matrix of size 3 x 2
> a
[1,] [2,]

```

```

[1,] 1 4
[2,] 2 5
[3,] 3 6
> rownames(a) <- c("one", "two", "three")
> a
[1,] [2,]
one 1 4
two 2 5
three 3 6
> colnames(a) <- c("ichi", "ni")
> a
ichi ni
one 1 4
two 2 5
three 3 6
> rownames(a)
[1] "one" "two" "three"
> colnames(a)
[1] "ichi" "ni"

```

```

> dimnames(a)
[1,]
[1] "one" "two" "three"
[2,]
[1] "ichi" "ni"
> a[, "ni"]
one two three
4 5 6
> a["two", ]
ichi ni
2 5
> ## list
> a <- list(1:10, abc)
[1,]
[1] 1 2 3 4 5 6 7 8 9 10

```



```

[[2]]
[1] "abc" "def" "ghi" "hello"
> a[[1]]
[1] 1 2 3 4 5 6 7 8 9 10
> a[[2]]
[1] "abc" "def" "ghi" "hello"
> a <- list(suji=1:10,moji=abc)
> a
$suji
[1] 1 2 3 4 5 6 7 8 9 10

$moji
[1] "abc" "def" "ghi" "hello"

> asuji
[1] 1 2 3 4 5 6 7 8 9 10
> asmoji
[1] "abc" "def" "ghi" "hello"

```

```

Hiroshima 2878915 110162
Yamaguchi 1527964 55796
Tokushima 824108 26357
Kagawa 1022890 38295
Ehime 1493092 48146
Kochi 813949 23417
Fukuoka 5015699 169834
Saga 876654 28484
Nagasaki 1516523 46426
Kumamoto 1859344 57580
Oita 1221140 42965
Miyazaki 117007 34026
Kagoshima 1786194 51166
Okinawa 1318220 34249
> x["Tokyo",]
jinkou souseisan
12064101 846809
> x["jinkou"]
Hokkaido Amori Iwate Miyagi Akita Yamagata Fukushima Toarr

```

```

> plot(x,type="n") # draw only frame
> text(x,rownames(x)) # draw labels
> f <- lfit(x[,1],x[,2]) # least square fitting (kaiki-bunseki)
> abline(f) # draw a fitted line

```

```

> ## simple analysis
> ## load a sample dataset
> source("shimo/class/gakubu200209/atac20020919c.R")
> x
jinkou souseisan
Hokkaido 5683062 197473
Amori 1475728 46520
Iwate 1416180 46949
Miyagi 2365320 86155
Akita 1189279 38414
Yamagata 1244147 41119
Fukushima 2126935 78345
Ibaraki 2985676 110819
Tochigi 2004917 79962
Gumma 2024852 77960
Saitama 6938006 199636
Chiba 5926285 183721
Tokyo 12064101 846809
Kanagawa 8489974 298661

```

```

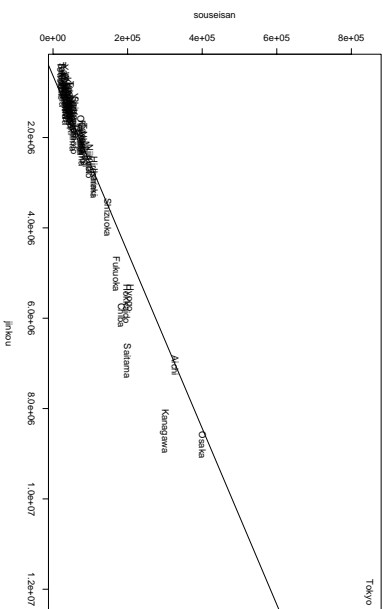
5683062 1475728 1416180 2365320 1189279 1244147 2126935 298661
Tochigi Gumma Saitama Chiba Tokyo Kanagawa Niigata Toyama
2004817 2024852 6938006 5926285 12064101 8489974 2475733 11208
Ishikawa Fukui Yamanaishi Nagano Gifu Shizuoka Aichi Niigata
1180977 828944 888172 2215168 2107700 3767393 7043300 1857
Shiga Kyoto Osaka Hyogo Nara Wakayama Tottori Shimane Fuku
1342832 2644391 8805081 550574 1442795 1069912 613289 761
Okayama Hiroshima Yamaguchi Tokushima Kagawa Kochi Fuku
1950828 2878915 1527964 824108 1022890 1493092 813949 5015
Saga Nagasaki Kumamoto Oita Miyazaki Kagoshima Okinawa
876654 1516523 1859344 1221140 117007 1786194 1318220

```

```

> ## plot and fitting
> plot(x) # simple plot

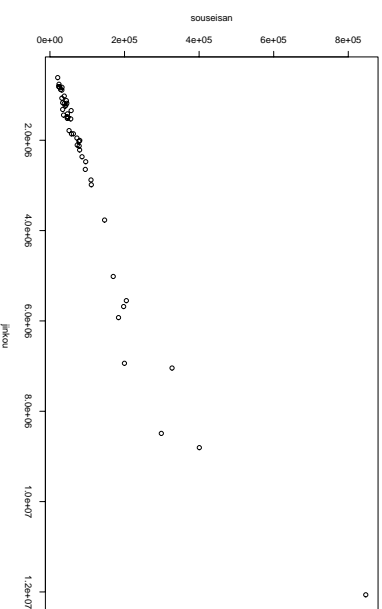
```



```

Niigata 2475733 95874
Toyama 1120851 44087
Ishikawa 1180977 45230
Fukui 828944 32426
Yamanaishi 888172 31981
Nagano 2215168 79508
Gifu 2107700 73078
Shizuoka 3767393 146616
Aichi 7043300 327476
Me 1857339 62717
Shiga 1342832 56815
Kyoto 2644391 94863
Osaka 8805081 400519
Hyogo 550574 204939
Nara 1442795 36523
Wakayama 1069912 31854
Tottori 613289 20817
Shimane 761503 24100
Okayama 1950828 72200

```



```

> ## calculate statistics
> mean(x[,1])
[1] 2700550
> mean(x[,2])
[1] 105961.0
> apply(x,2,mean)
jinkou souseisan
2700549.9 105961.0
> ## output postscript file
> postscript("ex01rg1.eps")
> plot(x)
> dev.off()
X11
2
> postscript("ex01rg2.eps")
> plot(x,type="n") # draw only frame
> text(x,rownames(x)) # draw labels
> abline(f) # draw a fitted line
> dev.off()

```

```

>> ## random number generation
> a <- runif(100) # uniform distribution on (0,1)
> a
[1] 0.33772072 0.96523453 0.11471802 0.45899514 0.25807340 0.91897002
[7] 0.19894305 0.71841781 0.17071827 0.58036916 0.72244387 0.30885956
[13] 0.39961291 0.44603197 0.49604300 0.08504009 0.02123887 0.71410722
[19] 0.42955228 0.44309812 0.64121792 0.95145627 0.38856641 0.96265811
[25] 0.29707709 0.04824197 0.29456821 0.45555714 0.43653072 0.09194056
[31] 0.96184772 0.23232472 0.99816295 0.87468042 0.59191003 0.58080336
[37] 0.75491096 0.15443375 0.13984499 0.51942845 0.68094018 0.91656951
[43] 0.87344172 0.28951874 0.20232800 0.44015944 0.20591796 0.80711408
[49] 0.30058366 0.70400184 0.15151447 0.36760816 0.87091817 0.902998059
[55] 0.36557006 0.70513375 0.93172830 0.04053871 0.78096573 0.33618138
[61] 0.17551581 0.49283572 0.47098055 0.85567305 0.42323136 0.42056184
[67] 0.45873132 0.84085186 0.67309793 0.14937437 0.37588200 0.61308521
[73] 0.27323279 0.15935226 0.07312308 0.26750706 0.31644778 0.50176397
[79] 0.55026224 0.37149845 0.94189510 0.10848281 0.73368034 0.22186991
[85] 0.45946215 0.80568116 0.41801127 0.46940970 0.71094321 0.91322983
[91] 0.35198930 0.65997991 0.74573909 0.17367779 0.03160446 0.35358868

```

```

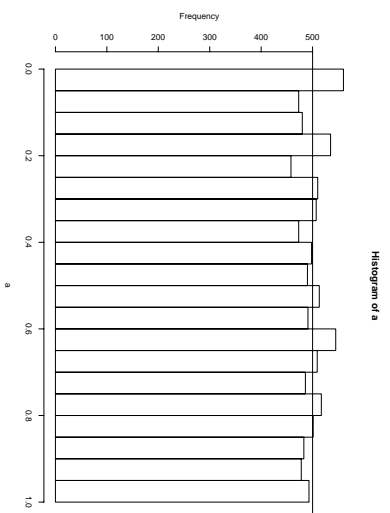
[97] 0.39337573 0.75346727 0.02309114 0.91162517
> hist(a)
> a <- runif(1000)
> hist(a)
> a <- runif(10000)
> mean(a)
[1] 0.5010653
> var(a)
[1] 0.08278935
> a <- runif(10000)
> mean(a)
[1] 0.4979175
> var(a)
[1] 0.08332883
> hist(a)
> abline(h=500)
> postscript("ex01unif.eps")
> hist(a)
> abline(h=500)

```

```

> dev.off()
X11
2
> hist(a)
> a <- rnorm(1000)
> hist(a)
> a <- rnorm(10000)
> hist(a)
[1] -0.005828347
> var(a)
[1] 0.992512
> b <- seq(-4, 4, length=100)
> lines(b, dnorm(b)*0.5*10000)
> postscript("ex01norm.eps")
> hist(a)
> lines(b, dnorm(b)*0.5*10000)
> dev.off()
X11
2

```



48

```

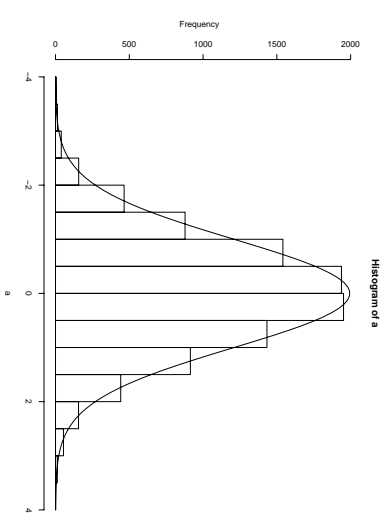
> a <- rnorm(100) # normal distribution with mean 0 and variance 1
> a
[1] -0.82691765 -2.02067099 0.19400892 -0.30382863 -0.31811388 -0.77346336
[7] -0.40153550 2.41079419 0.60562417 -0.97393404 -1.23733634 0.85196384
[13] -0.60534823 -0.77889770 -0.07444445 2.66534621 -0.83451795 0.45235685
[19] -0.10990285 1.08580316 -0.74960909 0.40406038 0.60829232 1.45375506
[25] 0.75997487 -1.20344286 -0.85792115 -0.15464927 0.15831217 0.08548434
[31] 0.50799391 -1.80017291 0.88274744 -1.15015568 0.01921664 2.05726484
[37] 0.37508515 -0.49540912 -0.92401419 0.38382862 0.27878916 -1.00021783
[43] -0.63664893 0.69261984 -0.32744108 0.01033082 -0.50170459 1.24632214
[49] 1.52796633 -0.49338017 0.68571754 -0.70039005 -0.28592276 -1.85568353
[55] 0.99690176 0.98597080 0.51932234 2.18836775 0.12552985 0.46521984
[61] -1.33403257 -0.89277524 1.32653428 -0.35091993 -0.37484179 0.40620333
[67] 1.42748708 0.65026413 0.42211847 1.34728761 -0.30578058 0.53137457
[73] 0.15892649 -0.35528093 1.53916414 1.65100974 0.70355433 0.56355801
[79] -0.24340207 -0.44164793 0.73611704 1.18361392 -0.87211264 1.08444714
[85] -0.57204720 1.17112253 -0.50147376 -0.32345874 2.00594036 0.68166394
[91] -0.65396266 0.24280340 0.89660829 -0.19244041 0.91239919 0.13789873
[97] -1.45605476 -1.66931721 -0.66200807 -2.32467900

```

```

> hist(a)
> a <- rnorm(1000)
> hist(a)
> a <- rnorm(10000)
> hist(a)
[1] -0.005828347
> var(a)
[1] 0.992512
> b <- seq(-4, 4, length=100)
> lines(b, dnorm(b)*0.5*10000)
> postscript("ex01norm.eps")
> hist(a)
> lines(b, dnorm(b)*0.5*10000)
> dev.off()
X11
2

```



50

```

>> ## function
> foo <- function(x) x**x
> foo
function(x) x**x
[1] 25
> foo(1:10)
[1] 1 4 9 16 25 36 49 64 81 100
> goo <- function(x,y) x**y
> goo(3,5)
[1] 15
> goo(1:10,2)
[1] 2 4 6 8 10 12 14 16 18 20
> goo <- function(x,y=2) x**y
> goo(1:10,2)
[1] 2 4 6 8 10 12 14 16 18 20
> goo(1:10)
[1] 2 4 6 8 10 12 14 16 18 20
> foo <- function(x) { # dont type the following "+" prompt marks

```

```

+ if(x > 0) y <- log(x)
+ else y <- log(-x)
+ y
+ }
> foo
function(x) {
if(x > 0) y <- log(x)
else y <- log(-x)
}
> foo(10)
[1] 2.302585
> foo(-5)
[1] 1.609438
> foo <- function(x) if(x>0) log(x) else log(-x)
> foo(-5)
[1] 1.609438
> foo <- function(x) log(abs(x))
> foo(-5)

```

```
[1] 1.609438
> ## and
> q()
```

```
Save workspace image? [y/n/c]: y
```

```
Process R finished at Fri Sep 20 00:47:40 2002
```