

資料轉換

吳漢銘

國立政治大學 統計學系





本章大綱&學習目標

2/63

- Types of Data Scales
- 常用基礎敘述統計套件: MASS , Hmisc , fBasics
- Why Data Transformations?
- Data Discretization, Reasons for Non-normality
- Common Transformations
 - Range , Reciprocal, Square Root, Log, Power Transformation , Box-Cox Transformations.
 - 範例: BUPA Liver Data Set
- Transformations for Proportions and Percents: Logit Transformation
- Variance Stabilizing Transformations
- Standardization
 - 範例: Microarray Data of Yeast Cell Cycle, Crab Data
- Species Data Transformation: The Doubs Fish Data
- Sphering for Multivariate Variables
- Which Transformation?
- Normalization for Microarray Gene Expression Data



Types of Data Scales

- **Categorical (類別資料), discrete, or nominal (名目變數)** — Values contain no ordering information: 性別、種族、教育程度、宗教信仰、交通工具、音樂類型... (qualitative 屬質)
- **Ordinal (順序)** — Values indicate order, but no arithmetic operations are meaningful (e.g., "novice", "experienced", and "expert" as designations of programmers participating in an experiment); 非常同意，同意，普通，不同意，非常不同意; 優，佳，劣。
- **Interval** — **Distances** between values are meaningful, but zero point is not meaningful. (e.g., degrees Fahrenheit)
- **Ratio (Continuous Data 連續型資料)** — Distances are meaningful and a zero point is meaningful (e.g., degrees K, 年收入、年資、身高、... (quantitative 計量))
- **Ordinal** methods cannot be used with nominal variable
- **Nominal** methods can be used with nominal, ordinal variables.



MASS套件常用基礎指令 (1/2)

4/63

■ MASS: nrow, ncol, dim, head, attributes, str, summary.

```
> library(MASS)
> data(Insurance)
> nrow(Insurance); ncol(Insurance)
[1] 64
[1] 5
> dim(Insurance)
[1] 64 5
> head(Insurance)
  District  Group   Age Holders Claims
1         1    <11   <25     197     38
2         1    <11 25-29     264     35
3         1    <11 30-35     246     20
4         1    <11   >35    1680    156
5         1 1-1.5l   <25     284     63
6         1 1-1.5l 25-29     536     84
> attributes(Insurance)
$names
[1] "District" "Group"    "Age"      "Holders"  "Claims"

$class
[1] "data.frame"

$row.names
[1] 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
[28] 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54
[55] 55 56 57 58 59 60 61 62 63 64
```

- **Insurance**資料集: 某保險公司在1973年第三季車險保險人的相關資訊。
- District: 投保人家庭住址所在區域: 1-4。
- Group: 所投保汽車的引擎排氣量(四個等級): <1公升, 1-1.5公升, 1.5-2公升, >2公升。
- Age: 投保人年紀(四個組別): <25歲, 25-29歲, 30-35歲, >35歲。
- Holders: 投保人數量。
- Claims: 要求索賠的投保人數量。



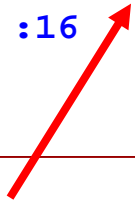
MASS套件常用基礎指令 (2/2)

5/63

- 一般來說，可以透過**平均值**和**中位數**的差異程度來判別資料的偏倚程度。當兩者相差過大，常常說明資料具有明顯的右偏或左偏情況。

```
> str(Insurance)
'data.frame': 64 obs. of 5 variables:
 $ District: Factor w/ 4 levels "1","2","3","4": 1 1 1 1 1 1 1 1 1 1 ...
 $ Group   : Ord.factor w/ 4 levels "<11"<"1-1.51"<...: 1 1 1 1 2 2 2 2 3 3
 ...
 $ Age      : Ord.factor w/ 4 levels "<25"<"25-29"<...: 1 2 3 4 1 2 3 4 1 2
 ...
 $ Holders  : int 197 264 246 1680 284 536 696 3582 133 286 ...
 $ Claims   : int 38 35 20 156 63 84 89 400 19 52 ...

> summary(Insurance)
District      Group      Age      Holders      Claims
1:16          <11      :16    <25      :16    Min.      : 3.00    Min.      : 0.00
2:16          1-1.51:16    25-29:16    1st Qu.: 46.75    1st Qu.: 9.50
3:16          1.5-21:16    30-35:16    Median : 136.00   Median : 22.00
4:16          >21      :16    >35      :16    Mean      : 364.98   Mean      : 49.23
                                     3rd Qu.: 327.50   3rd Qu.: 55.50
                                     Max.      :3582.00   Max.      :400.00
```



平均值高於中位數的兩倍左右，資料有右偏趨勢。



Hmisc套件: describe (1/2)

6/63

- Hmisc: describe 對於不同類型變數列出不同內容的結果，具有一套輸出規則。
 - 對於一個取樣水準不超過10的數值型變數，會被預設為離散型變數。函數會列出連續變數的各分位點值; 對於一個非二分變數，且其取樣水準不超過20，則會列出該變數的頻率表; 當任一變數的取樣水準超過20，就會分別列出頻率最低和最高的5個水準值。

```
> install.packages("Hmisc")
> library(Hmisc)
> describe(Insurance[, 1:3])
Insurance[, 1:3]
```

```
3 Variables      64 Observations
```

```
District
```

```
  n missing unique
64         0      4
```

```
1 (16, 25%), 2 (16, 25%), 3 (16, 25%), 4 (16, 25%)
```

```
Group
```

```
  n missing unique
64         0      4
```

```
<11 (16, 25%), 1-1.51 (16, 25%), 1.5-21 (16, 25%), >21 (16, 25%)
```

```
Age
```

```
  n missing unique
64         0      4
```

```
<25 (16, 25%), 25-29 (16, 25%), 30-35 (16, 25%), >35 (16, 25%)
```




Hmisc套件: describe (2/2)

7/63

```
> describe(Insurance[, 4:5])
```

```
Insurance[, 4:5]
```

```
2 Variables      64 Observations
```

Holders

n	missing	unique	Info	Mean	.05	.10	.25	.50	.75
64	0	63	1	365	16.30	24.00	46.75	136.00	327.50
.90	.95								
868.90	1639.25								

```
lowest :      3      7      9     16     18, highest: 1635 1640 1680 2443 3582
```

Claims

n	missing	unique	Info	Mean	.05	.10	.25	.50	.75
64	0	46	1	49.23	3.15	4.30	9.50	22.00	55.50
.90	.95								
101.70	182.35								

```
lowest :      0      2      3      4      5, highest: 156 187 233 290 400
```



- fBasics: basicStats
- 金融工程相關的套件，basicStats 用於計算時間序列資料基礎統計指標的函數。

```
> install.packages("fBasics")
> library(fBasics)
> basicStats(Insurance$Holders)
X..Insurance.Holders
nobs          6.400000e+01
NAs           0.000000e+00
Minimum       3.000000e+00
Maximum       3.582000e+03
1. Quartile   4.675000e+01
3. Quartile   3.275000e+02
Mean          3.649844e+02
Median        1.360000e+02
Sum           2.335900e+04
SE Mean       7.784632e+01
LCL Mean      2.094209e+02
UCL Mean      5.205478e+02
Variance      3.878432e+05
Stdev         6.227706e+02
Skewness       3.127833e+00
Kurtosis       1.099961e+01
```

sum 23359, 約2.3萬投保人資訊
。且在District, Group, Age ,
平均有365位投保人。

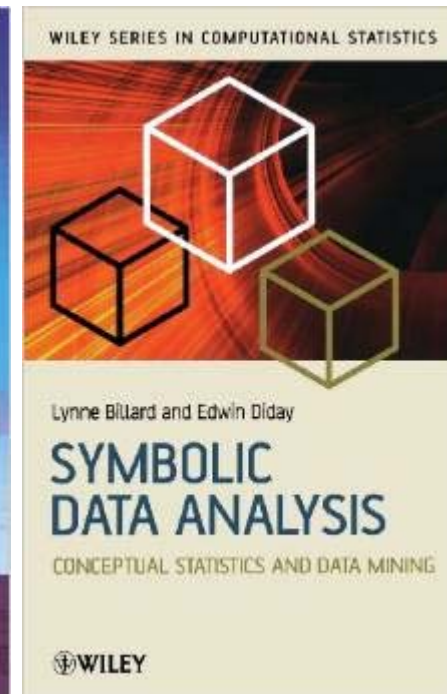
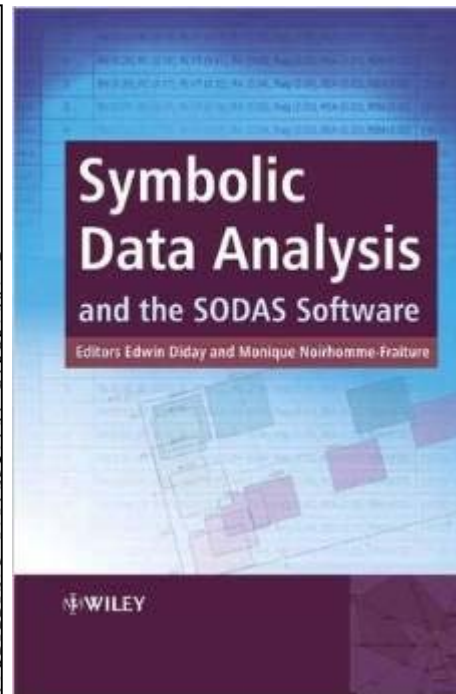
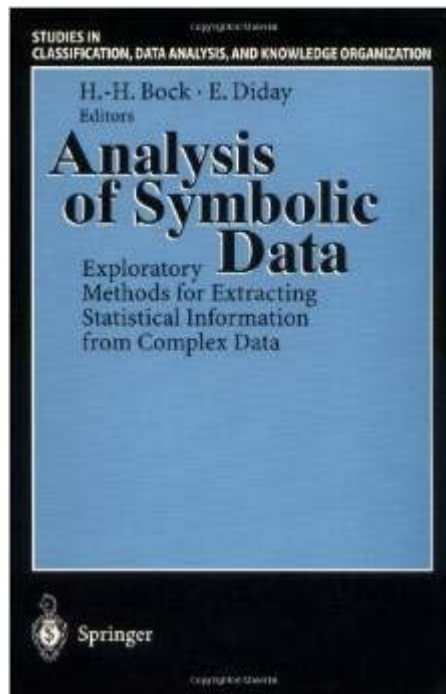
偏態係數 (偏度): $s=0$: 分佈對稱, $|s|<1$: 對稱性較強, $|s|>1$ & $s>0$: 右偏, $|s|>1$ & $s<0$: 左偏
峰態係數 (峰度)

```
> library(timeDate)
> skewness(Insurance[, 4:5])
Holders   Claims
3.127833  2.877292
> kurtosis(Insurance[, 4:5])
Holders   Claims
10.999610  9.377258
```

偏態係數= $E(X-u)^3/\sigma^3$
大於0: 右偏分配
等於0: 對稱分配
小於0: 左偏分配
峰態係數= $E(X-u)^4/\sigma^4$
大於3: 高峽峰
等於3: 常態峰
小於3: 低闊峰

Symbolic Data (象徵性資料)

9/63



- Hans-Hermann Bock, Edwin Diday, 2000, Analysis of Symbolic Data: Exploratory Methods for Extracting Statistical Information from Complex Data, Springer.
 - Lynne Billard, Edwin Diday, 2003, From the Statistics of Data to the Statistics of Knowledge- Symbolic Data Analysis. Journal of the American Statistical Association, 98, pp.470-487
 - Lynne Billard, Edwin Diday, 2007, Symbolic Data Analysis: Conceptual Statistics and Data Mining, Wiley; 1 edition.
 - Edwin Diday, Monique Noirhomme-Fraiture, 2008, Symbolic Data Analysis and the SODAS Software, Wiley-Interscience.
- **How to do the EDA for symbolic data?**
 - **How to do the transformation (e.g., standardization) for symbolic data?**



Classical (Numerical) Data Table

10/63

j th variables

subjects
 i th sample

UID	alpha0	alpha7	alpha14	alpha21	alpha28	alpha35	alpha42
YAR007C	-0.48	-0.42	0.87	0.92	0.67	-0.18	-0.35
YBL035C	-0.39	-0.58	1.08	1.21	0.52	-0.33	-0.13
YBR023C	0.87	0.25	-0.17	0.18	-0.13	-0.44	-0.13
YBR067C	1.57	1.03	1.22	0.31	0.16	-0.49	-1.02
YBR088C	-1.15	-0.86	1.21	1.62	1.12	0.16	-0.44
YBR278W	0.04	-0.12	0.31	0.16	0.17	-0.06	0.08
YCL055W	2.95	0.45	-0.4	-0.66	-0.59	-0.38	-0.76
YDL003W	-1.22	-0.74	1.34	1.5	0.63	0.29	-0.55
YDL055C	-0.73	-1.06	-0.79	-0.02	0.16	0.44	0.03
YDL102W	-0.58	-0.4	0.13	0.58	-0.09	0.02	-0.45
YDL164C	-0.5	-0.42	0.66	1.05	0.68	0.06	0.01
YDL197C	-0.86	-0.29	0.42	0.46	0.3	0.1	-0.63
YDL227C	-0.16	0.2877	0.17	-0.28	-0.02	-0.55	-0.04
YDR052C	-0.36	-0.03	-0.03	-0.08	-0.23	-0.25	-0.21
YDR097C	-0.72	-0.85	0.54	1.04	0.84	0.24	-0.64
YDR113C	-0.78	-0.52	0.26	0.2	0.48	0.48	0.27
YDR309C	0.6	-0.55	0.41	0.45	0.18	-0.66	-1.02
YDR356W	-0.2	-0.67	0.13	0.1	0.38	0.44	0.05
YER001W	-2.29	-0.635739	0.77	1.6	0.53	0.55	-0.38
YER070W	-1.46	-0.76	1.08	1.5	0.74	0.47	-0.7
YER095W	-0.57	0.42	1.03	1.35	0.64	0.42	-0.4
YGL163C	-0.11	0.13	0.41	0.6	0.23	0.31	0.19
YGL225W	-1.08	-0.99	-0.16	0.2	0.61	0.37	0.1
YGR109C	-1.79	0.9449	2.13	1.75	0.23	0.15	-0.66

transformation
for each row

transformation
for each column

transformation
for both rows
and columns



Data Transformations

- A transformation of a set of data points x_1, x_2, \dots, x_n is a function T that substitutes each observation x_i with a new value $T(x_i)$.
- Transformations should have the following properties:
 - The **order** of the data is preserved by the transformation.
 - They are **continuous functions** guaranteeing that points that are close together in raw form are also close together using their transformed values, relative to the scale used.
 - They are **smooth functions** that have derivatives of all orders, and they are specified by elementary functions.
- In EDA, we might want to change the shape of data (**reexpressed**) to facilitate visualization, smoothing, and other analyses.



Why Data Transformations? (1/2)

- Many statistical procedures make two assumptions that are relevant to data transformation:
 - (a) the variables (or their error terms) are **normally distributed**.
 - (b) homoscedasticity or **homogeneity of variance**, meaning that the variance of the variable remains constant over the observed range of some other variable.
- In regression analyses the assumption (b) is that the variance around the regression line is constant across the entire observed range of data.
- In ANOVA analyses, the assumption (b) is that the variance in one cell is not significantly different from that of other cells.
- In some cases, transforming the data will make it **fit the assumptions** better.



Why Data Transformations? (2/2)

13/63

- Transforms are usually applied:
 - so that the data appear to more closely **meet the assumptions** of a statistical inference procedure that is to be applied,
 - to make it **easier to visualize** (appearance of graphs),
 - to improve **interpretability**, even if no formal statistical analysis or visualization is to be performed.
 - to make descriptors that have been measured in **different units comparable**,
 - to make the relationships among **variables linear**,
 - to modify the **weights** of the variables or objects (e.g. give the same length (or norm) to all object vectors)
 - to **code** categorical variables into dummy binary variables.
- Guidance for how data should be transformed, or whether a transformation should be applied at all, should come from the particular statistical analysis to be performed.



Reasons for Using Transformations (1/2)

1. Convenience

- A transformed scale may be as natural as the original scale and more convenient for a specific purpose (e.g. percentages rather than original data, sines rather than degrees)

2. Reducing skewness

- A distribution that is symmetric or nearly so (e.g, Gaussian distribution) is often easier to handle and interpret than a skewed distribution.
 - To reduce right skewness, take roots or logarithms or reciprocals (roots are weakest).
 - To reduce left skewness, take squares or cubes or higher powers.

3. Equal spreads

- Equal spreads makes data easier to handle and interpret.
- Each data set or subset having about the same spread or variability is a condition called **homoscedasticity**: its opposite is called **heteroscedasticity**.



Reasons for Using Transformations (2/2)

15/63

4. Linear relationships

- It is easier to think about patterns (relationships between variables) that are approximately linear than about patterns that are highly curved.
- This is vitally important when using linear regression, which amounts to fitting such patterns to data.
 - e.g, a plot of logarithms of a series of values against time has the property that periods with constant rates of change (growth or decline) plot as straight lines.
 - The model $y = a e^{bx}$ is made linear by $\ln y = \ln a + bx$.

5. Additive relationships

- Relationships are often easier to analyze when additive rather than (say) multiplicative.
- So $y = a + bx$ in which two terms a and bx are added is easier to deal with than $y = ax^b$ in which two terms a and x^b are multiplied.
- Additivity is a vital issue in analysis of variance.



Data Transformation

16/63

- **Smoothing:** This uses binning, regression, and clustering to remove noise from the data.
- **Attribute construction:** In this routine, new attributes are constructed and added from the given set of attributes.
- **Aggregation:** In this summary or aggregation, operations are performed on the data.
- **Normalization:** Here, the attribute data is scaled so as to fall within a smaller range.
- **Discretization:** In this routine, the raw values of a numeric attribute are replaced by interval label or conceptual label.
- **Concept hierarchy generation for nominal data:** Here, attributes can be generalized to higher level concepts.



Data Discretization

- Data discretization transforms numeric data by mapping values to interval or concept labels.
- **by binning**: This is a top-down unsupervised splitting technique based on a specified number of bins.
- **by histogram analysis**: In this technique, a histogram partitions the values of an attribute into disjoint ranges called buckets or bins. It is also an unsupervised method.
- **by cluster analysis**: In this technique, a clustering algorithm can be applied to discretize a numerical attribute by partitioning the values of that attribute into clusters or groups.
- **by decision tree analysis**: Here, a decision tree employs a top-down splitting approach; it is a supervised method. To discretize a numeric attribute, the method selects the value of the attribute that has minimum entropy as a split-point, and recursively partitions the resulting intervals to arrive at a hierarchical discretization.
- **by correlation analysis**: This employs a bottom-up approach by finding the best neighboring intervals and then merging them to form larger intervals, recursively. It is supervised method.



Reasons for Non-normality

18/63

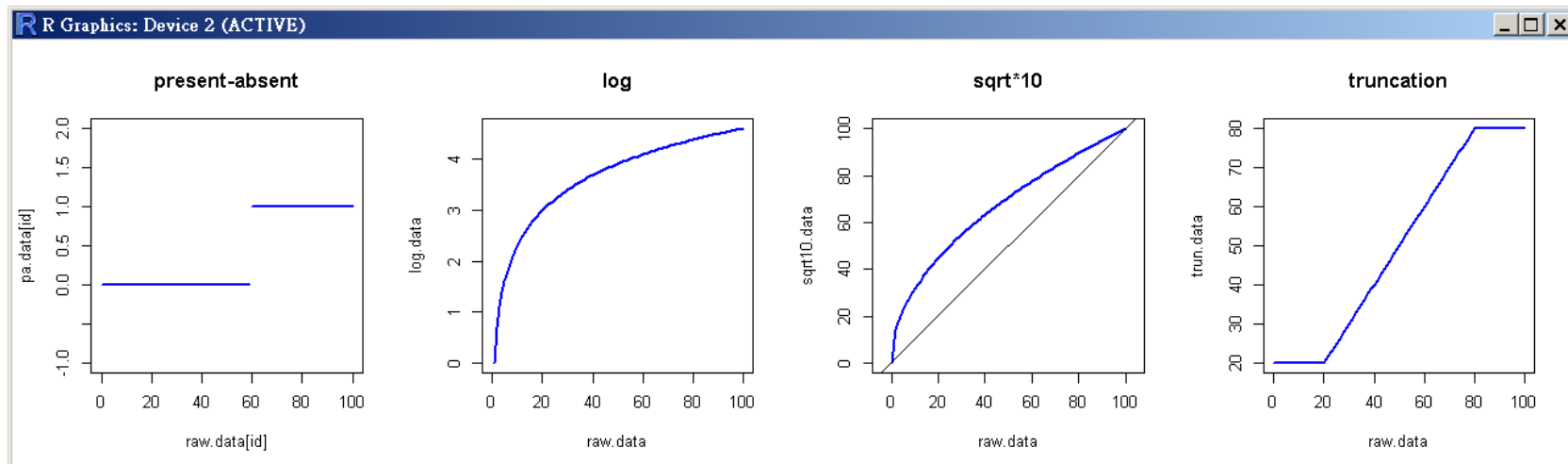
- The **mistakes** appeared in data entry.
- The **missing** data values were not declared as missing.
- The presence of **outliers** (scores that are extreme relative to the rest of the sample).
- The **nature** of the variable itself.
- How does one tell when a variable is violating the assumption of normality?
 - "eyeballing the data," or visual inspection (Orr, Sackett, and DuBois, 1991),
 - simple examination of skew and kurtosis,
 - examination of QQ plots,
 - inferential tests of normality: Kolmogorov-Smirnov test.



Common Transformations (1/3)

```
> par(mfrow=c(1,4))
> raw.data <- 0:100
> pa.data <- ifelse(raw.data >= 60, 1, 0)
> id <- which(pa.data==1)
> plot(raw.data[id], pa.data[id], main="present-absent",
+ type="l", lwd=2, col="blue", ylim=c(-1, 2), xlim=c(0, 100))
> points(raw.data[-id], pa.data[-id], type="l", lwd=2, col="blue")
>
> log.data <- log(raw.data)
> plot(raw.data, log.data, main="log", type="l", lwd=2, col="blue")
>
> sqrt10.data <- sqrt(raw.data)*10
> plot(raw.data, sqrt10.data, main="sqrt*10", type="l", lwd=2, col="blue", asp=1)
> abline(a=0, b=1)
>
> trun.data <- ifelse(raw.data >= 80, 80, ifelse(raw.data < 20, 20, raw.data))
> plot(raw.data, trun.data, main="truncation", type="l", lwd=2, col="blue")
```

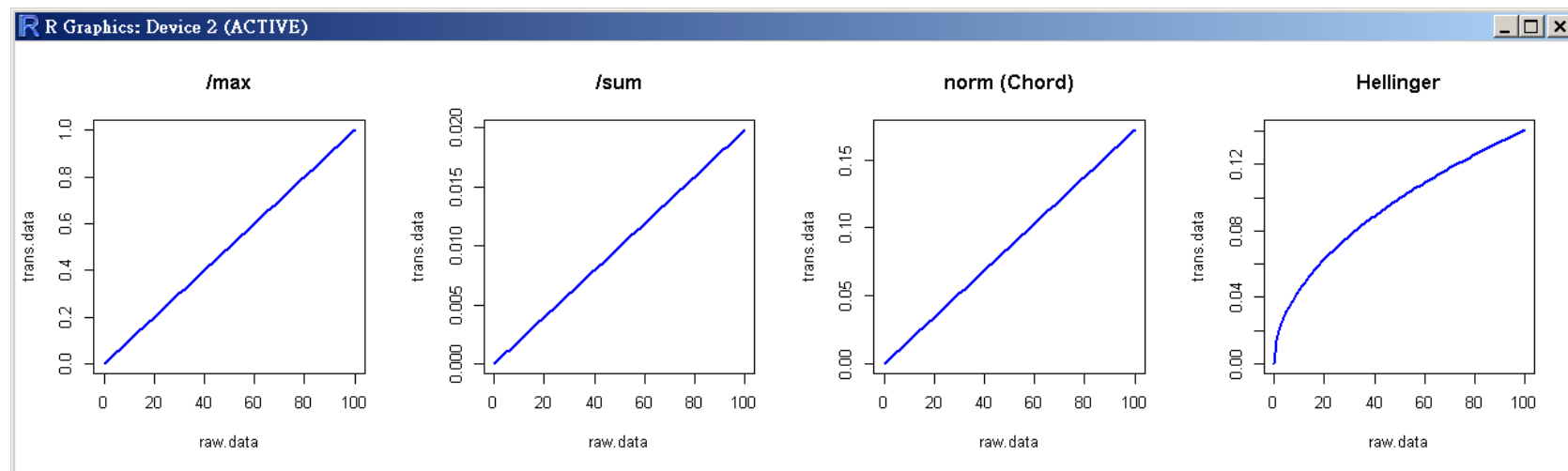
NOTE: `apply(raw.data.matrix, 2, log)`
`apply(raw.data.matrix, 2, function(x) sqrt(x)*10)`
`apply(raw.data.matrix, 2, myfun)`





Common Transformations (2/3)

```
> par(mfrow=c(1,4))
> raw.data <- 0:100
> trans.data <- raw.data/max(raw.data)
> plot(raw.data, trans.data, main="/max", type="l", lwd=2, col="blue")
>
> trans.data <- raw.data/sum(raw.data) #Species profile transformation
> plot(raw.data, trans.data, main="/sum", type="l", lwd=2, col="blue")
>
> trans.data <- raw.data/sqrt(sum(raw.data^2)) #length of 1, Chord transformation
> plot(raw.data, trans.data, main="norm (Chord)", type="l", lwd=2, col="blue")
>
> trans.data <- sqrt(raw.data/sum(raw.data)) #Hellinger transformation
> plot(raw.data, trans.data, main="Hellinger", type="l", lwd=2, col="blue")
```

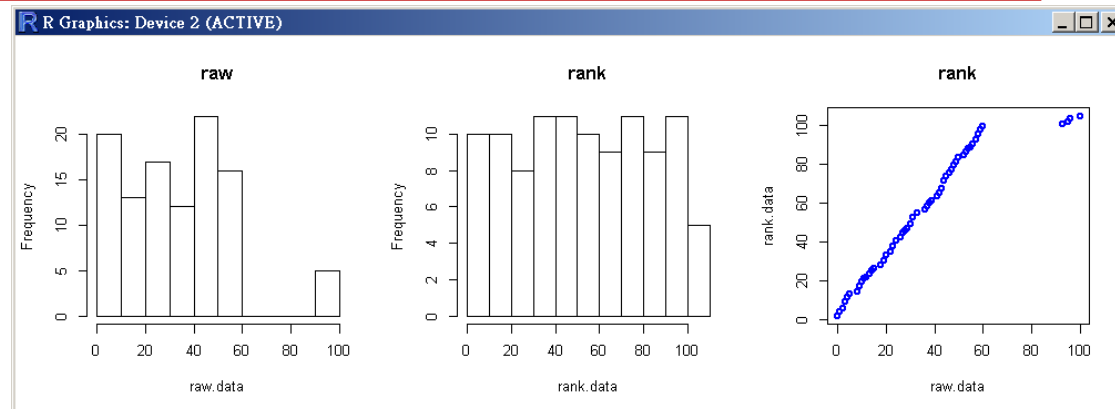


Other Transformations for community composition data: Chi-square distance transformation, Chi-square metric transformation



Common Transformations (3/3)

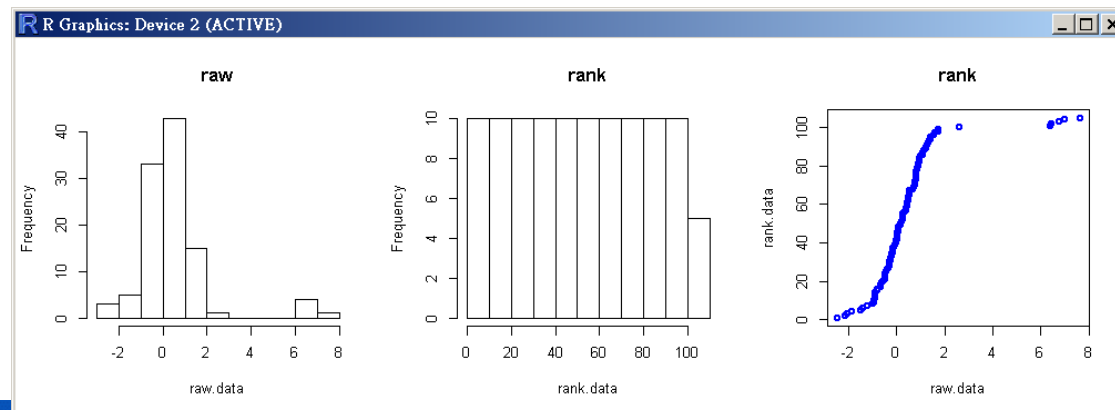
```
> par(mfrow=c(1,3)); set.seed(12345)
> raw.data <- c(sample(0:60, 100, replace=T), sample(90:100, 5, replace=T))
> rank.data <- rank(raw.data)
> hist(raw.data, main="raw")
> hist(rank.data, main="rank")
> plot(raw.data, rank.data, main="rank", lwd=2, col="blue")
>
> raw.data <- c(rnorm(100), rnorm(5)+ 2*sqrt(qchisq(0.975, 5)))
...
```



Outlier values ~

$$2 \times \sqrt{\chi_{0.975,p}^2} + N(0, 1)$$

```
> qchisq(0.975,5)
[1] 12.83250
```



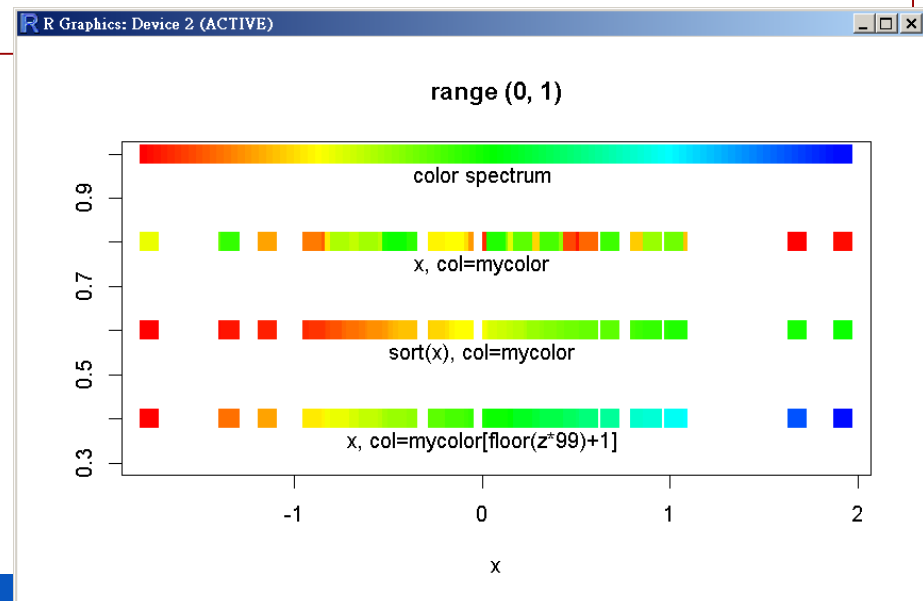


Transformation Using the Range

- use the range of the variable as the divisor:
 - $z = x / (\max(x) - \min(x))$; $z = (x - \min(x)) / (\max(x) - \min(x))$, is bounded by zero and one, with at least one observed value at each of the end points.

```
x <- rnorm(50)
mycolor <- rainbow(150)[1:100]
z <- (x - min(x)) / (max(x) - min(x))
plot(x, rep(1, length(x)), main="range (0, 1)", type="n", ylab="", ylim=c(0.3,1))
points(c(seq(min(x), max(x), length.out=100)), rep(1, 100), col=mycolor, cex=2, pch=15)
text(0, 0.95, "color spectrum")
points(x, rep(0.8, length(x)), col=mycolor, cex=2, pch=15)
text(0, 0.75, "x, col=mycolor")
points(sort(x), rep(0.6, length(x)), col=mycolor, cex=2, pch=15)
text(0, 0.55, "sort(x), col=mycolor")
points(x, rep(0.4, length(x)), col=mycolor[floor(z*99)+1], cex=2, pch=15)
text(0, 0.35, "x, col=mycolor[floor(z*99)+1]")
```

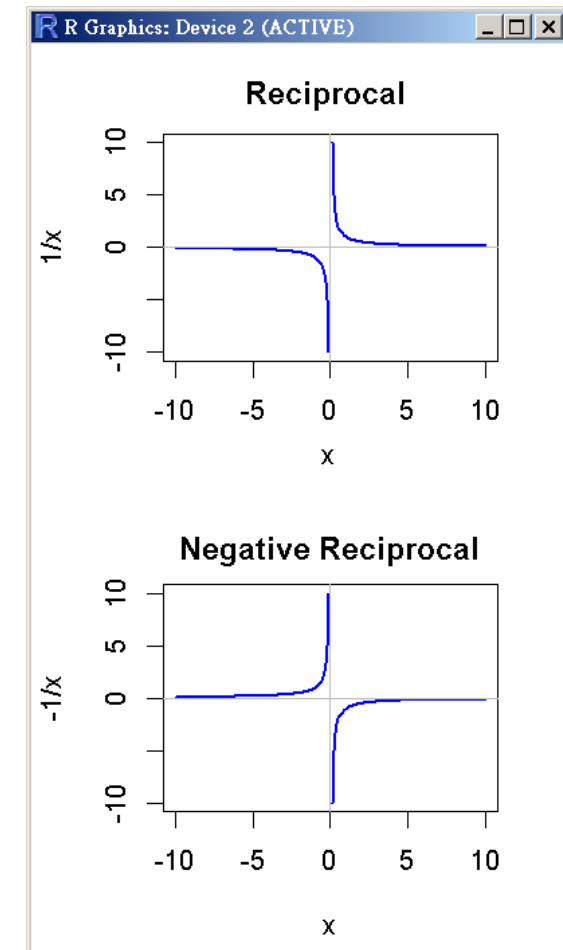
- The transformed variate is a linear function of the other one, so data standardized using these transformations will result in identical Euclidean distances.





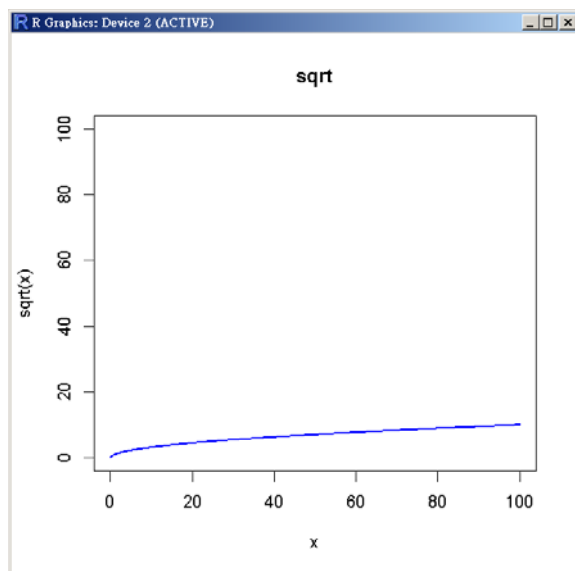
The Reciprocal Transformation

- The **reciprocal**, x to $1/x$, with its sibling the **negative reciprocal**, x to $-1/x$, is a very strong transformation with a drastic effect on distribution shape.
- The reciprocal reverses order among values of the same sign: largest becomes smallest, etc.
- The negative reciprocal preserves order among values of the same sign.
- (In practice, we might want to multiply or divide the results of taking the reciprocal by some constant, such as 1000 or 10000, to get numbers that are easy to manage, but that itself has no effect on skewness or linearity.)
- The reciprocal of a **ratio** may be interpreted as easily as the ratio itself:
 - population density (people per unit area) becomes area per person;
 - persons per doctor becomes doctors per person;
 - rates of erosion become time to erode a unit depth.

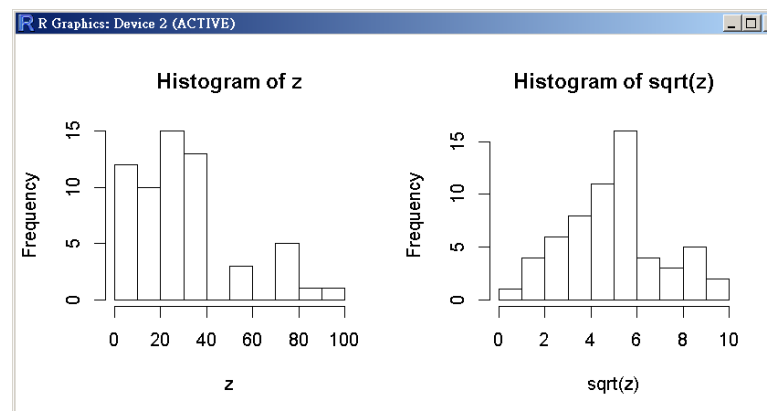


The Square Root Transformation

- The square root, x to $x^{1/2} = \text{sqrt}(x)$, is a transformation with a moderate effect on distribution shape: it is weaker than the logarithm and the cube root.
- It is used for **reducing right skewness**, and also has the advantage that it can be applied to zero values.
- Note that the square root of an area has the units of a length. It is commonly applied to **counted data**, especially if the values are mostly rather small.



```
> x <- sample(0:40, 50, replace=T)
> y <- sample(40:100, 10)
> z <- c(x,y)
> par(mfrow=c(1,2))
> hist(z)
> hist(sqrt(z))
```





範例: Software Inspection Data

25/63

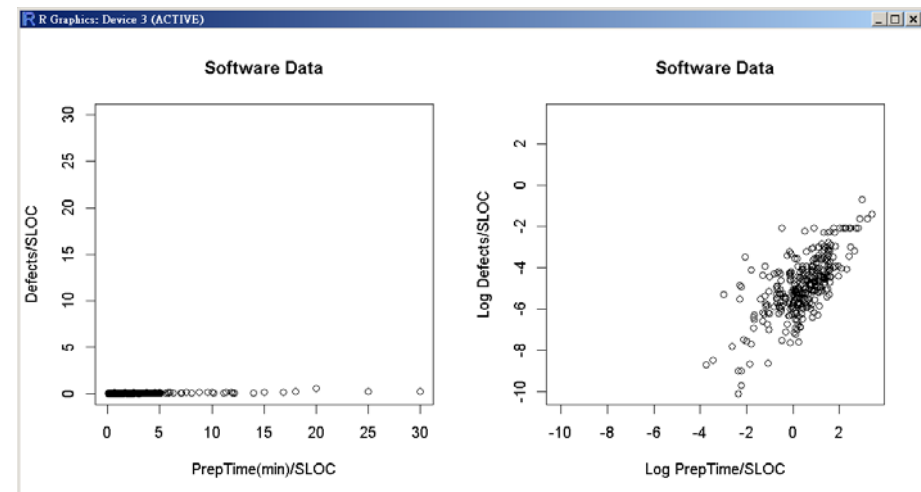
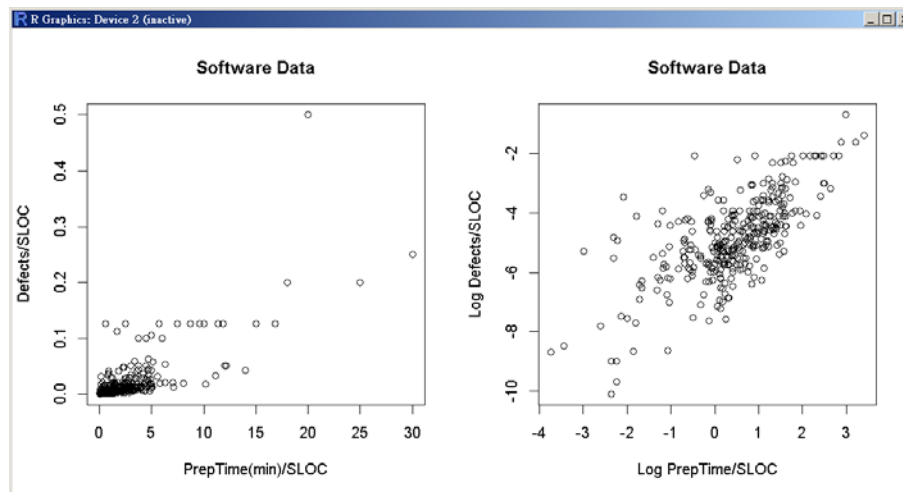
- The data were collected in response to efforts for process improvement in software testing by code inspection.
- First they look for **inconsistencies**, **logical errors**, etc., and decide what they perceive as **defects**. The defect types include compatibility, design, human-factors, standards, and others.
- The variables are normalized by the size of the inspection (the number of pages or **SLOC (single lines of code)**: the preparation time in minutes (**prepage**, **prepsloc**), the total work hours in minutes for the meeting (**mtgsloc**), and the number of defects found (**defpage**, **defsloc**).
- Interested in: understanding the relationship between the inspection time and the number of defects found.

```
> library('R.matlab')
> data <- readMat("software.mat")
> print(data)
...
> str(data)
List of 5
 $ prepsloc: num [1:426, 1] 0.485 0.54 0.54 0.311 0.438 ...
 $ defsloc : num [1:426, 1] 0.005 0.002 0.002 0.00328 0.00278 ...
 $ mtgsloc : num [1:426, 1] 0.075 0.06 0.06 0.2787 0.0417 ...
 $ prepage : num [1:491, 1] 6.15 1.47 1.47 5.06 5.06 ...
 $ defpage : num [1:491, 1] 0.0385 0.0267 0.0133 0.0128 0.0385 ...
```

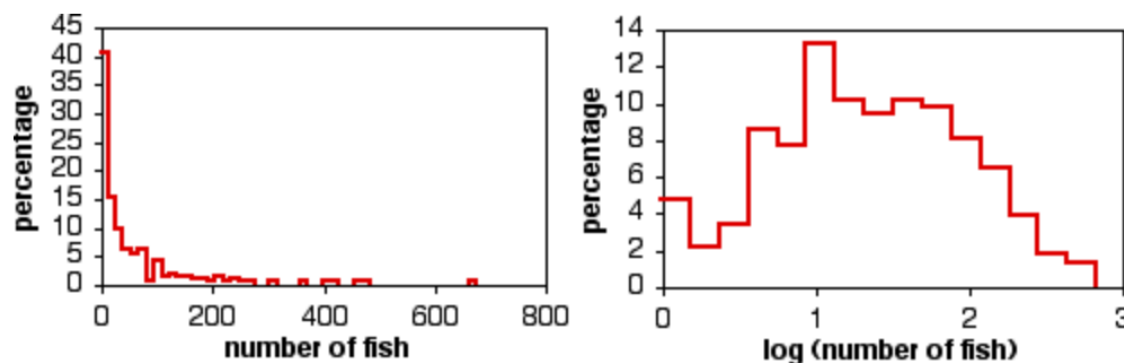
Log Transformations (1/3)

- The data are skewed, and the relationship between the variables is difficult to understand.
- We apply a log transform to both variables. Be careful about creating something that is not really there, but is just an artifact of the processing.
- In any application of EDA, the analyst should go back to the **subject area** and consult **domain experts** to verify and help interpret the results.

```
par(mfrow=c(1,2))
xlim <- range(data$prepsloc, data$defsloc)
plot(data$prepsloc, data$defsloc, xlab="PrepTime(min)/SLOC", ylab="Defects/SLOC",
     main="Software Data", xlim=xlim, ylim=ylim)
logxlim <- range(log(data$prepsloc), log(data$defsloc))
plot(log(data$prepsloc), log(data$defsloc), xlab="Log PrepTime/SLOC",
     ylab="Log Defects/SLOC", main="Software Data", xlim=logxlim, ylim=logxlim)
```



Log Transformations (2/3)

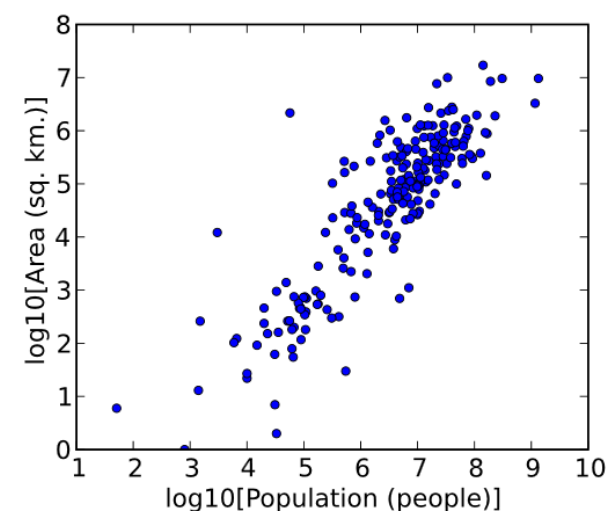
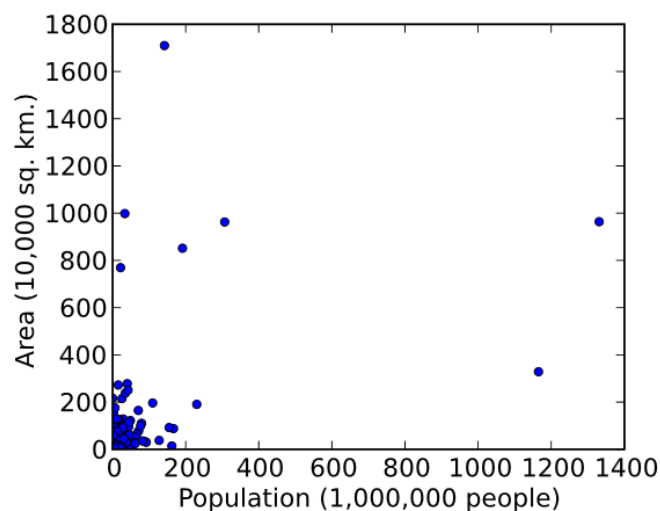


Source:

<http://www.biostathandbook.com/transformation.html>

Histograms of number of Eastern mudminnows per 75 m section of stream (samples with 0 mudminnows excluded). Untransformed data on left, log-transformed data on right.

The areas of the sovereign states and dependent territories in the world are plotted on the vertical axis against their populations on the horizontal axis.



https://en.wikipedia.org/wiki/Data_transformation_%28statistics%29



Log Transformations (3/3)

- Consider ratios $y = p/q$ where p and q are both **positive** in practice. Examples are
 - males / females;
 - dependents / workers;
 - downstream length / downvalley length.
- Then y is somewhere between 0 and infinity, or between 1 and infinity. (If $p = q$, then $y = 1$.) Such definitions often lead to **skewed data**, because there is a clear lower limit and no clear upper limit.
- The logarithm, $\log y = \log (p/q) = \log p - \log q$, is somewhere between -infinity and infinity and $p = q$ means that $\log y = 0$. Hence the logarithm of such a ratio is likely to be more **symmetrically distributed**.



How to Handle Negative Data Values?

- In many cases, the variable of interest is positive and the log transformation is immediately applicable. However, some quantities (for example, profit) might contain a few negative values. How do you handle negative values if you want to log-transform the data?

Solution 1: Translate, then Transform

- $\log(Y+a)$ where a is the constant.
- Some people like to choose a so that $\min(Y+a)$ is a very small positive number (like 0.001). Others choose a so that $\min(Y+a) = 1$. For the latter choice, you can show that $a = b - \min(Y)$, where b is either a small number or is 1.
- $\text{cond}(x \leq 0, -\ln(-x + 1), \ln(x + 1))$

Solution 2: Use Missing Values

- A criticism of the previous method is that some practicing statisticians don't like to add an arbitrary constant to the data.
- They argue that a better way to handle negative values is to use missing values for the logarithm of a nonpositive number.



Power Transformation (1/2)

- For data vectors (y_1, \dots, y_n) in which each $y_i > 0$, the power transform is

$$y_i^{(\lambda)} = \begin{cases} \frac{y_i^\lambda - 1}{\lambda(\text{GM}(y))^{\lambda-1}}, & \text{if } \lambda \neq 0 \\ \text{GM}(y) \ln y_i, & \text{if } \lambda = 0 \end{cases}$$

where

$$\text{GM}(y) = (y_1 \cdots y_n)^{1/n}$$

is the **geometric mean** of the observations y_1, \dots, y_n .

- The power transform corresponds to **a family of functions** that are applied to create a **monotonic transformation** of data using power functions.
- This transformation is used to **stabilize variance**, make the data more **normal distribution-like**, improve the validity of measures of association such as the Pearson correlation between variables and for other data stabilization procedures.
- When both negative and positive values are observed, it is more common to begin by **adding a constant** to all values, producing a set of non-negative data to which any power transformation can be applied.



Box-Cox Transformations (1/3)

$$y(\lambda) = \begin{cases} \frac{y^\lambda - 1}{\lambda}, & \text{if } \lambda \neq 0; \\ \log y, & \text{if } \lambda = 0. \end{cases} \quad \text{Box and Cox(1964)}$$

$$y(\boldsymbol{\lambda}) = \begin{cases} \frac{(y + \lambda_2)^{\lambda_1} - 1}{\lambda_1}, & \text{if } \lambda_1 \neq 0; \\ \log(y + \lambda_2), & \text{if } \lambda_1 = 0. \end{cases} \quad \begin{aligned} &\boldsymbol{\lambda} = (\lambda_1, \lambda_2)' \\ &\text{choose } \lambda_2 \text{ such that} \\ &y + \lambda_2 > 0 \text{ for any } y. \end{aligned}$$

- The aim of the Box-Cox transformations is to ensure the **usual assumptions for Linear Model hold**. $\mathbf{y} \sim N(\mathbf{X}\boldsymbol{\beta}, \sigma^2 \mathbf{I}_n)$
- Clearly not all data could be power-transformed to Normal. Draper and Cox (1969) studied this problem and conclude that even in cases that no power-transformation could bring the distribution to exactly normal, the usual estimates of lambda will lead to a distribution that satisfies certain restrictions on the **first 4 moments**, thus will be usually **symmetric**.

Source: Box-Cox Transformations: An Overview, Pengfei Li, Department of Statistics, University of Connecticut, Apr 11, 2005

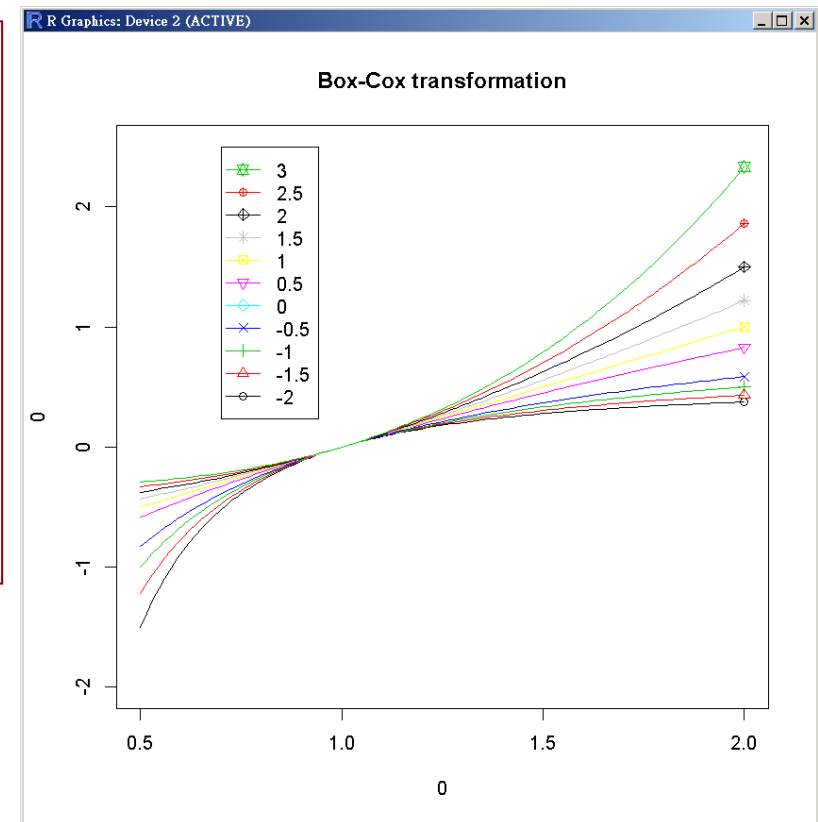


Box-Cox Transformations (2/3)

$$x'_\lambda = \frac{e^{\lambda \log(x)} - 1}{\lambda} \approx \frac{(1 + \lambda \log(x) + \frac{1}{2}\lambda^2 \log(x)^2 + \dots) - 1}{\lambda} \rightarrow \log(x) \text{ as } \lambda \rightarrow 0.$$

```
x <- seq(0.5, 2, length.out=100)
bc <- function(y, lambda){
  (y^lambda - 1)/lambda
}
lambda <- seq(-2, 3, 0.5)
plot(0, 0, type="n", xlim=c(0.5, 2),
     ylim=c(-2, 2.5), main="Box-Cox transformation")
for(i in 1:length(lambda)){
  points(x, bc(x, lambda[i]), type="l", col=i)
  points(2, bc(2, lambda[i]), col=i, pch=i)
}
legend(0.7, 2.5, legend=as.character(rev(lambda)),
      lty=1, pch=length(lambda):1,
      col=length(lambda):1)
```

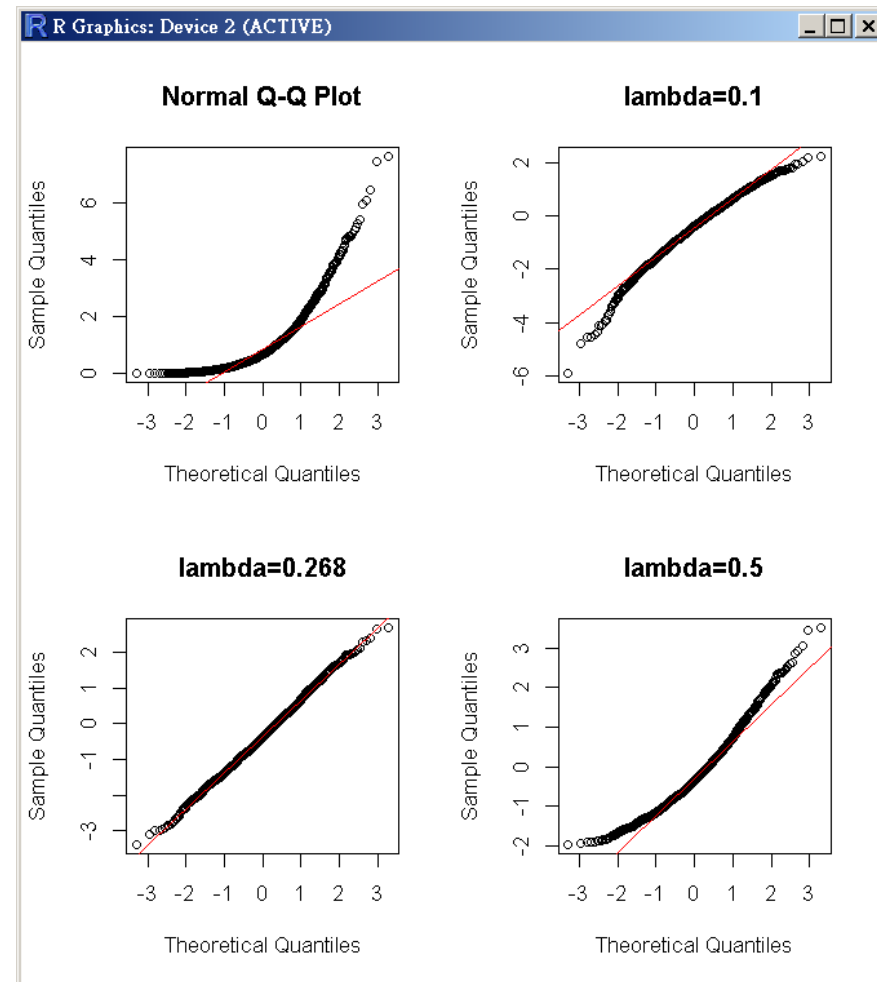
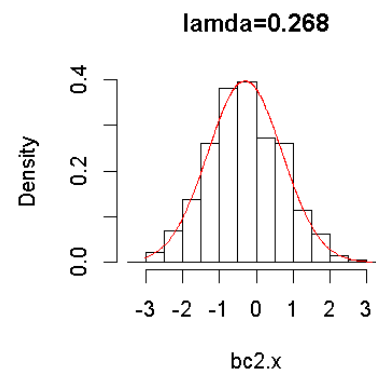
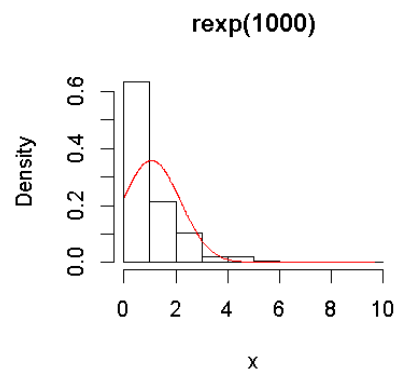
$\lambda = 0$?



<http://onlinestatbook.com/2/transformations/box-cox.html>

Box-Cox Transformations (3/3)

```
> x <- rexp(1000)
> bc <- function(y, lambda){
+   (y^lambda - 1)/lambda
+ }
> bc1.x <- bc(x, 0.1)
> bc2.x <- bc(x, 0.268)
> bc3.x <- bc(x, 0.5)
> par(mfrow=c(2, 2))
> qqnorm(x); qqline(x, col="red")
> qqnorm(bc1.x, main="lambda=0.1")
> qqline(bc1.x, col="red")
> qqnorm(bc2.x, main="lambda=0.268")
> qqline(bc2.x, col="red")
> qqnorm(bc3.x, main="lambda=0.5")
> qqline(bc3.x, col="red")
```



$$\left(\Phi^{-1} \left(\frac{i - 0.5}{n} \right), x_{(i)} \right), \quad \text{for } i = 1, 2, \dots, n,$$

Source: Box-Cox Transformations: An Overview, Pengfei Li, Department of Statistics, University of Connecticut, Apr 11, 2005

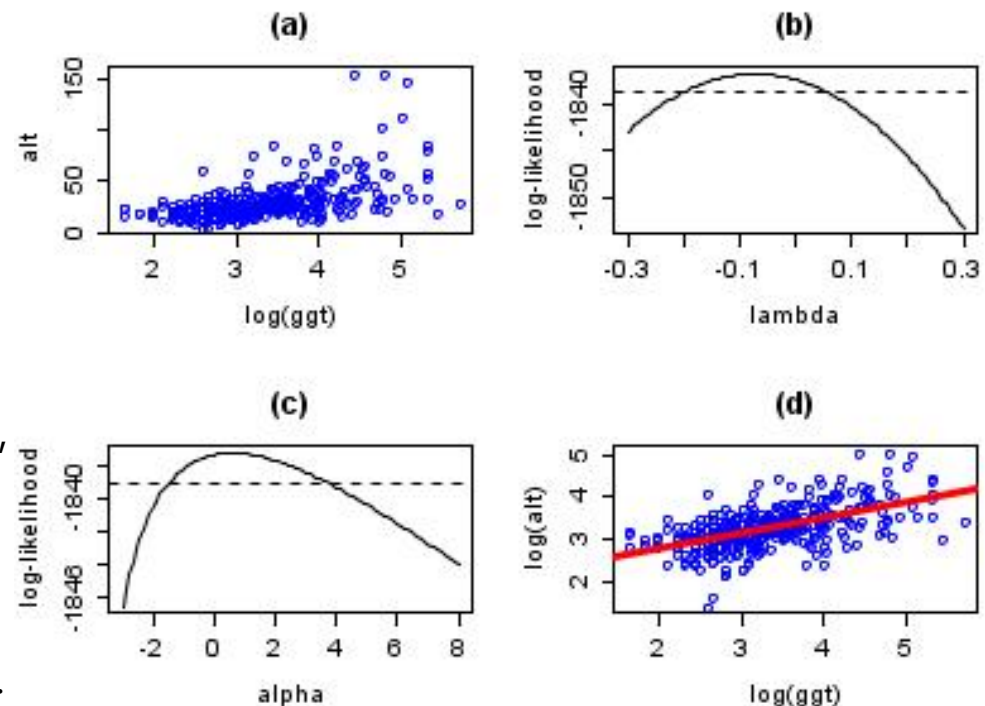


範例: BUPA Liver Data Set

34/63

- The BUPA liver data set contains data on liver enzymes ALT and γ GT. Suppose we are interested in using $\log(\gamma\text{GT})$ to predict ALT. (a): there appears to be non-constant variance, and a Box–Cox transformation might help.
- Possibly, the transformation could be improved by adding a shift parameter to the log transformation. (c): in this case, the maximum of the likelihood is close to zero suggesting that a shift parameter is not needed.
- (d) The final panel shows the transformed data with a superimposed regression line.
- Note that although Box–Cox transformations can make big improvements in model fit, there are some issues that the transformation cannot help with. In the current example, the data are rather heavy-tailed so that the assumption of normality is not realistic and a robust regression approach leads to a more precise model.

https://en.wikipedia.org/wiki/Power_transform





Box-Cox Transformations: Summary

- It is not always necessary or desirable to transform a data set to resemble a normal distribution. However, if symmetry or normality are desired, they can often be induced through one of the power transformations.

$$y(\lambda) = \begin{cases} \frac{y^\lambda - 1}{\lambda}, & \text{if } \lambda \neq 0; \\ \log y, & \text{if } \lambda = 0. \end{cases}$$

$\lambda = 1.00$: no transformation needed; produces results identical to original data

$\lambda = 0.50$: square root transformation

$\lambda = 0.33$: cube root transformation

$\lambda = 0.25$: fourth root transformation

$\lambda = 0.00$: natural log transformation

$\lambda = -0.50$: reciprocal square root transformation

$\lambda = -1.00$: reciprocal (inverse) transformation

and so forth.



Modified Box-Cox Transformations

Manly(1971)

$$y(\lambda) = \begin{cases} \frac{e^{\lambda y} - 1}{\lambda}, & \text{if } \lambda \neq 0; \\ y, & \text{if } \lambda = 0. \end{cases}$$

Negative y's could be allowed. The transformation was reported to be successful in transform unimodal skewed distribution into normal distribution, but is not quite useful for **bimodal** or **U-shaped distribution**.

John and Draper(1980) “Modulus Transformation”

$$y(\lambda) = \begin{cases} \text{Sign}(y) \frac{(|y|+1)^\lambda - 1}{\lambda}, & \text{if } \lambda \neq 0; \\ \text{Sign}(y) \log(|y| + 1), & \text{if } \lambda = 0, \end{cases} \quad \text{Sign}(y) = \begin{cases} 1, & \text{if } y \geq 0; \\ -1, & \text{if } y < 0. \end{cases}$$

Bickel and Doksum(1981)

$$y(\lambda) = \frac{|y|^\lambda \text{Sign}(y) - 1}{\lambda}, \quad \text{for } \lambda > 0,$$

Yeo and Johnson(2000)

$$y(\lambda) = \begin{cases} \frac{(y+1)^\lambda - 1}{\lambda}, & \text{if } \lambda \neq 0, y \geq 0; \\ \log(y + 1), & \text{if } \lambda = 0, y \geq 0; \\ \frac{(1-y)^{2-\lambda} - 1}{\lambda - 2}, & \text{if } \lambda \neq 2, y < 0; \\ -\log(1 - y), & \text{if } \lambda = 2, y < 0. \end{cases}$$

Source: Box-Cox Transformations: An Overview, Pengfei Li, Department of Statistics, University of Connecticut, Apr 11, 2005



Transformations for Proportions and Percents: Logit Transformation (2/2)

37/63

- If values are naturally restricted to be in the **range 0 to 1**, not including the end-points, then a logit transformation may be appropriate: this yields values in the range $(-\infty, \infty)$.
- The logit of a number p between 0 and 1 is given by the formula:

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right) = \log(p) - \log(1-p) = -\log\left(\frac{1}{p} - 1\right).$$

- When the function's parameter represents a probability p , the logit function gives the log-odds, or the logarithm of the odds $p/(1-p)$.

$$P(y_i = 0) = 1 - \pi_i, \quad P(y_i = 1) = \pi_i$$

$$\text{Logistic link function: } g(\pi) = \log\left(\frac{\pi}{1-\pi}\right)$$

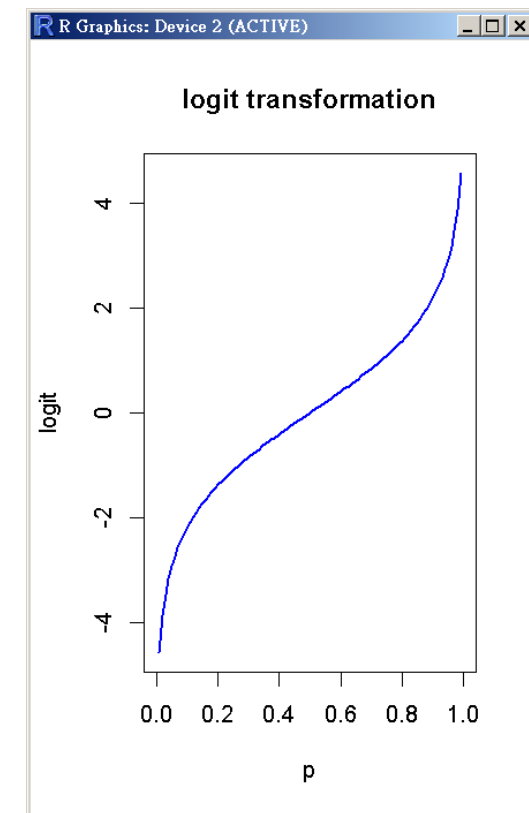
$$\text{Logistic regression: } \log\left(\frac{\pi}{1-\pi}\right) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon$$



Transformations for Proportions and Percents: Logit Transformation (2/2)

38/63

- This transformation treats very small and very large values symmetrically, pulling out the tails and pulling in the middle around 0.5 or 50%. The plot of p against $\text{logit}(p)$ is thus a flattened S-shape.
- Strictly, $\text{logit } p$ cannot be determined for the extreme values of 0 and 1 (100%): if they occur in data, there needs to be some adjustment.





Variance Stabilizing Transformations

- Many types of statistical data exhibit a "**variance-on-mean relationship**", meaning that the variability is different for data values with different expected values.
- A variance-stabilizing transformation aims to remove a variance-on-mean relationship, so that the variance becomes **constant** relative to the mean.
- Examples of variance-stabilizing transformations
 - Fisher transformation for the sample correlation coefficient,
 - Square root transformation or Anscombe transform for Poisson data (count data),
 - Box-Cox transformation for regression analysis and
 - Arcsine square root transformation or angular transformation for proportions (binomial data).

https://en.wikipedia.org/wiki/Data_transformation_%28statistics%29

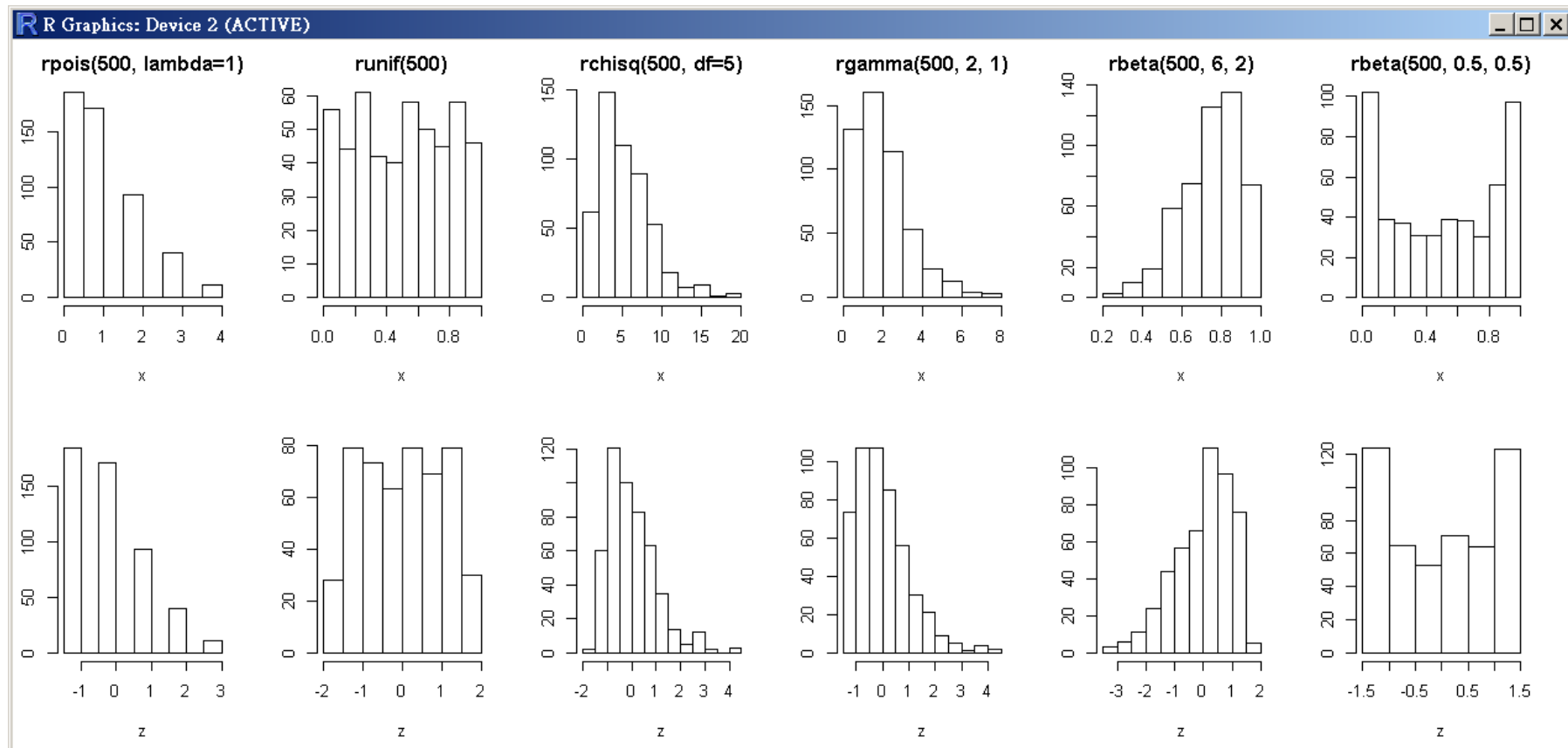


Standardization (1/5)

- $z = x/s$, will have a variance of one and a transformed mean equal to \bar{x}/s .
- Standardization, $z = (x - \bar{x})/s$, (called z-score): the new variate z will have a mean of zero and a variance of one. (also called centering and scaling.)
- If the variables are measurements along a **different scale** or if the standard deviations for the variables are different from one another, then one variable might **dominate** the distance (or some other similar calculation) used in the analysis:
- Standardization is useful for comparing variables expressed in different units.
- In some multivariate contexts, the transformations may be applied to each variable separately.
 - **Standardization makes no difference to the shape of a distribution.**

Standardization (2/5)

- Standardization makes no difference to the shape of a distribution.



```
x <- rpois(500, lambda=1)
hist(x, main="rpois(500, lambda=1)"); z <- scale(x); hist(z, main="")
```



In Clustering Analysis

- When the z-score transformation is used in a clustering context, it is important that it be applied in a global manner across all observations.
- If standardization is done within clusters, then false and misleading clustering solutions can result [Milligan and Cooper, 1988].

In Calculating Distance/Similarity (e.g., multidimensional scaling)

- Euclidean distances calculated on data that have been transformed using the two formulas result in identical dissimilarity values.
- For robust versions of Equations, we can substitute the **median** and the **interquartile range** for the sample mean and sample variance.

Standardization (4/5)

USArrests {datasets}: Violent Crime Rates by US State

A data frame with 50 observations on 4 variables.

- [1] Murder: Murder arrests (per 100,000)
- [2] Assault: Assault arrests (per 100,000)
- [3] UrbanPop: Percent urban population
- [4] Rape: Rape arrests (per 100,000)

```
> head(USArrests)
```

	Murder	Assault	UrbanPop	Rape
Alabama	13.2	236	58	21.2
Alaska	10.0	263	48	44.5
Arizona	8.1	294	80	31.0
Arkansas	8.8	190	50	19.5
California	9.0	276	91	40.6
Colorado	7.9	204	78	38.7

```
> par(mfrow=c(4,1))
```

```
> r <- range(USArrests)
```

```
> hist(USArrests$Murder, xlim = r)
```

```
> hist(USArrests$Assault, xlim = r)
```

```
> hist(USArrests$UrbanPop, xlim = r)
```

```
> hist(USArrests$Rape, xlim = r)
```

```
> USArrests.std <- as.data.frame(
  apply(USArrests, 2, scale))
```

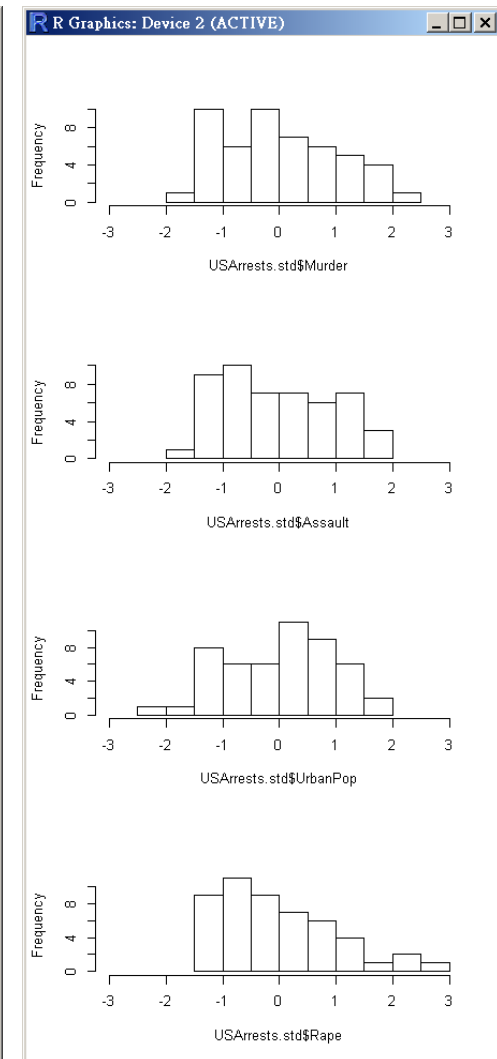
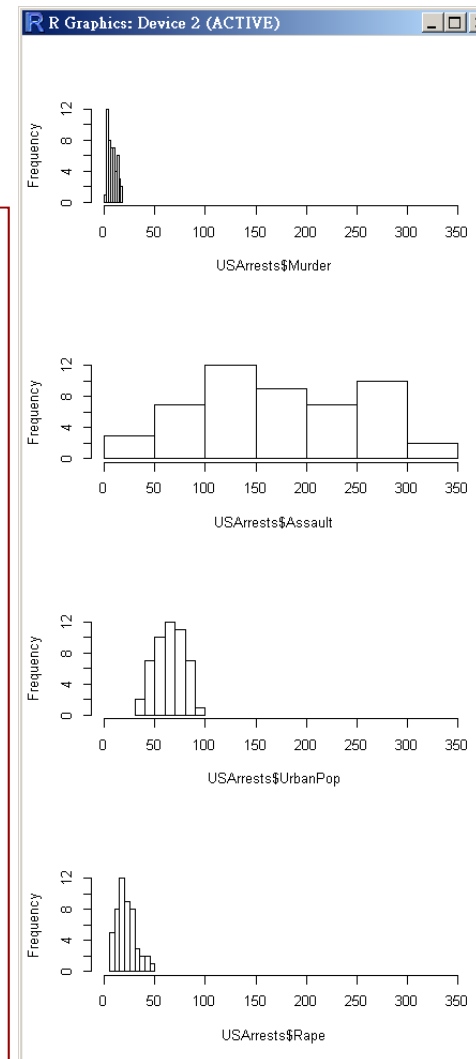
```
> r.std <- c(-3, 3)
```

```
> hist(USArrests.std$Murder, xlim = r.std)
```

```
> hist(USArrests.std$Assault, xlim = r.std)
```

```
> hist(USArrests.std$UrbanPop, xlim = r.std)
```

```
> hist(USArrests.std$Rape, xlim = r.std)
```



Standardization (5/5)

```
airquality {datasets}
```

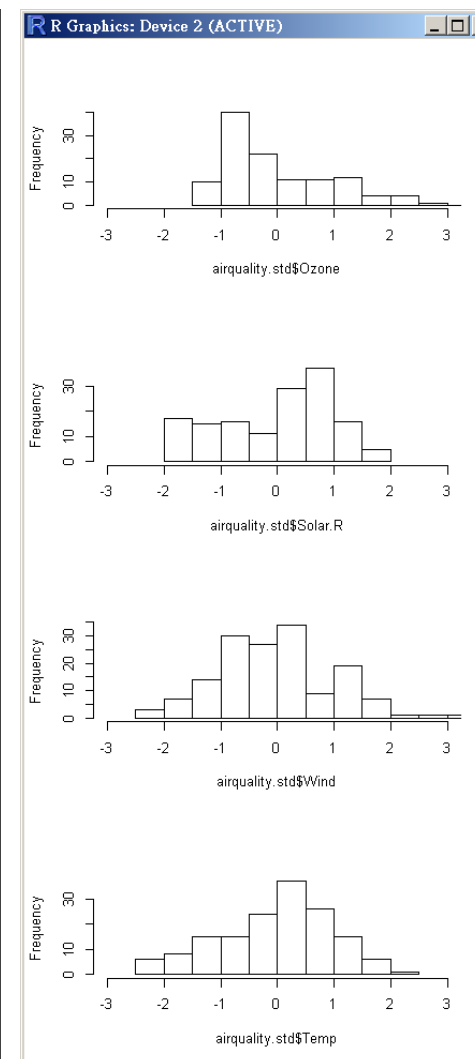
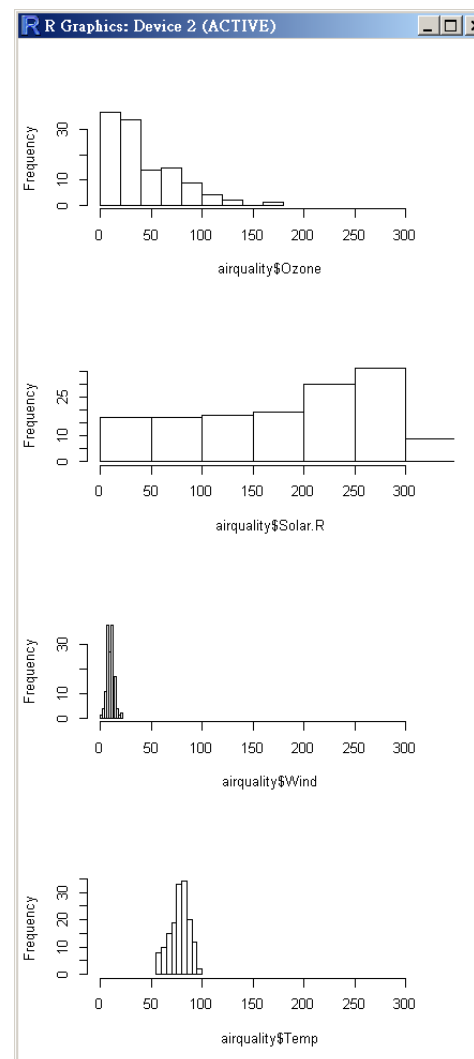
New York Air Quality Measurements: Daily air quality measurements in New York, May to September 1973.

A data frame with 154 observations on 6 variables.

- [1] Ozone: Ozone (ppb)
- [2] Solar.R: Solar R (lang)
- [3] Wind: Wind (mph)
- [4] Temp: Temperature (degrees F)
- [5] Month: Month (1--12)
- [6] Day: Day of month (1--31)

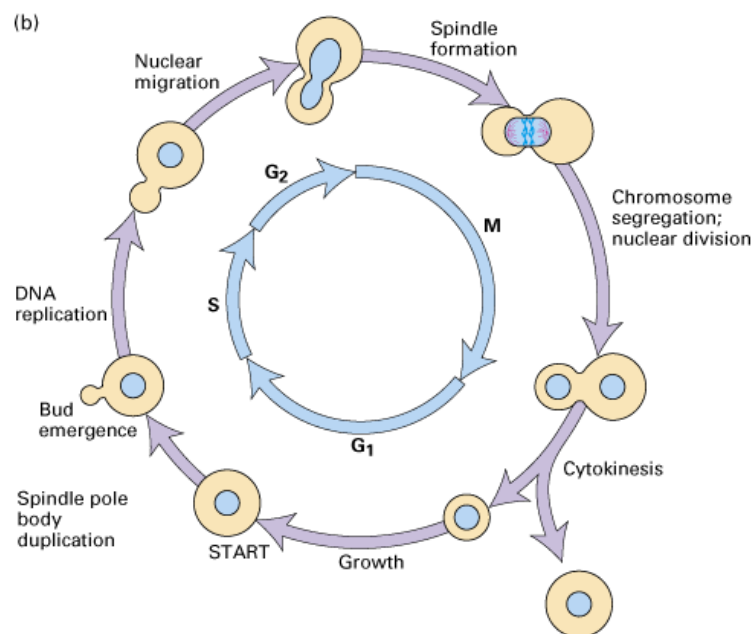
```
> head(airquality )
  Ozone Solar.R Wind Temp Month Day
1    41     190  7.4   67     5    1
2    36     118  8.0   72     5    2
3    12     149 12.6   74     5    3
4    18     313 11.5   62     5    4
5    NA      NA 14.3   56     5    5
6    28      NA 14.9   66     5    6

> r <- range(airquality[,1:4], na.rm = T)
> hist(airquality$Ozone , xlim = r)
> hist(airquality$Solar.R, xlim = r)
> hist(airquality$Wind, xlim = r)
> hist(airquality$Temp, xlim = r)
>
> airquality.std <- as.data.frame(
  apply(airquality, 2, scale))
> r.std <- c(-3, 3)
> hist(airquality.std$Ozone, xlim = r.std)
> hist(airquality.std$Solar.R, xlim = r.std)
> hist(airquality.std$Wind, xlim = r.std)
> hist(airquality.std$Temp, xlim = r.std)
```



範例: Microarray Data of Yeast Cell Cycle

- The data consists of several sub-sets collected under different conditions: alpha factor arrest, elutriation, arrest of cdc15 and cdc28 temperature-sensitive mutant.
- Each of these sub-sets is a single experiment. These experiment methods are used to synchronize the yeast cell cycle. Synchronized by alpha factor arrest method: Spellman et al. (1999).
- Time course data: every 7 minutes and totally 18 time points.
- Known genes: there are 103 cell cycle-regulated genes by traditional method in G₁, S, S/G₂, G₂/M, or M/G₁.



Microarray Data of Yeast Cell Cycle.xls								
	A	B	C	D	E	F	G	H
1	gene	phase.name	alpha0	alpha7	alpha14	alpha21	alpha28	alpha35
2	YAR007C	G1	-0.48	-0.42	0.87	0.92	0.67	-0.18
3	YBL035C	G1	-0.39	-0.58	1.08	1.21	0.52	-0.33
4	YBR023C	G1	0.87	0.25	-0.17	0.18	-0.13	-0.44
5	YBR067C	G1	1.57	1.03	1.22	0.31	0.16	-0.49
6	YBR088C	G1	-1.15	-0.86	1.21	1.62	1.12	0.16
7	YBR278W	G1	0.04	-0.12	0.31	0.16	0.17	-0.06
8	YCL055W	G1	2.95	0.45	-0.40	-0.66	-0.59	-0.38
9	YDL003W	G1	-1.22	-0.74	1.34	1.50	0.63	0.29
10	YDL055C	G1	-0.73	-1.06	-0.79	-0.02	0.16	0.44
11	YDL102W	G1	-0.58	-0.40	0.13	0.58	-0.09	0.02
12	YDL164C	G1	-0.50	-0.42	0.66	1.05	0.68	0.06
13	YDL197C	G1	-0.86	-0.29	0.42	0.46	0.30	0.10
14	YDL227C	G1	-0.16	0.29	0.17	-0.28	-0.02	-0.55
15	YDR052C	G1	-0.36	-0.03	-0.03	-0.08	-0.23	-0.25
16	YDR097C	G1	-0.72	-0.85	0.54	1.04	0.84	0.24
17	YDR113C	G1	-0.78	-0.52	0.26	0.20	0.48	0.48
18	YDR309C	G1	0.60	-0.55	0.41	0.45	0.18	-0.66
19	YDR356W	G1	-0.20	-0.67	0.13	0.10	0.38	0.44
20	YER001W	G1	-2.29	-0.64	0.77	1.60	0.53	0.55
21	YER070W	G1	-1.46	-0.76	1.08	1.50	0.74	0.47
22	YER095W	G1	-0.57	0.42	1.03	1.35	0.64	0.42
23	YGL163C	G1	-0.11	0.13	0.41	0.60	0.23	0.31
24	YGL225W	G1	-1.08	-0.99	-0.16	0.20	0.61	0.37
25	YGR109C	G1	-1.79	0.94	2.13	1.75	0.23	0.15

Spellman et al., (1998). Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast *Saccharomyces cerevisiae* by Microarray Hybridization. *Molecular Biology of the Cell* 9, 3273-3297.

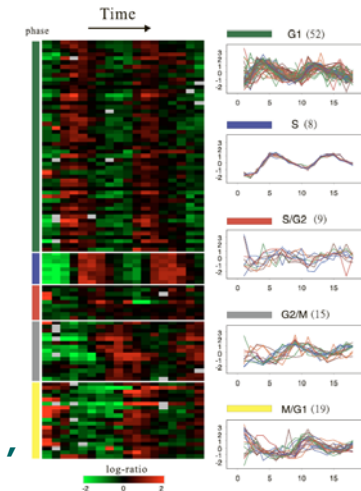
Image: http://www.pha.jhu.edu/~ghzheng/old/webct/note7_3.htm



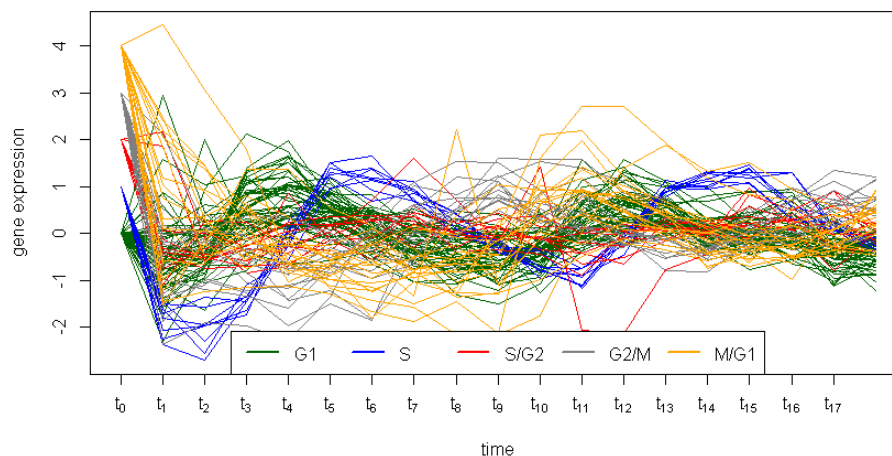
Standardization in Time Series Microarray Gene Expression Experiments

46/63

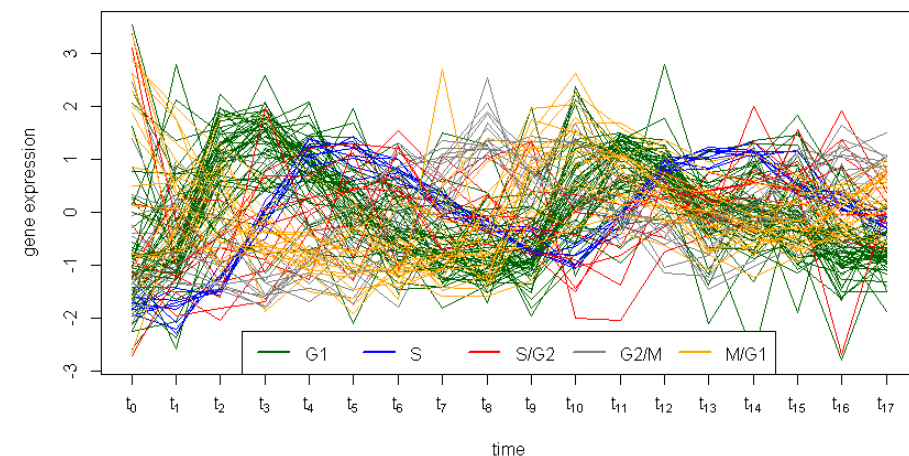
```
cell.raw <- read.table("trad_alpha103.txt", row.names=1, header=T)
head(cell.raw)
cell.xdata <- t(scale(t(cell.raw[,2:19]), center=T, scale=T))
y.C <- as.integer(cell.raw[,1])
table(y.C)
no.cluster <- length(unique(y.C))
cellcycle.color <- c("darkgreen", "blue", "red", "gray50", "orange")
p <- ncol(cell.raw) - 1
ycolors <- cellcycle.color[y.C+1]
my.pch <- c(1:no.cluster)[y.C+1]
phase <- c("G1", "S", "S/G2", "G2/M", "M/G1")
matplot(t(cell.xdata), pch = 1:p, lty=1, type = "l", ylab="gene expression",
        col=ycolors, xlab="time", main="Time series", xaxt="n")
time.label <- parse(text=paste("t[",0:p,"]",sep=""))
axis(1, 1:(p+1), time.label)
legend("bottom", legend=phase, col=cellcycle.color, lty=1, horiz = T, lwd=2)
```



Time series



Time series



The data map for 103 cell cycle-regulated genes and the plots of time courses for each phase. Each expression profile is normalized as mean equals zero and variance 1.



範例: Crab Data (1/4)

47/63

crabs {MASS}

Morphological Measurements on Leptograpsus Crabs

Description: The crabs data frame has **200 rows** and **8 columns**, describing 5 morphological measurements on 50 crabs each of two colour forms and both sexes, of the species *Leptograpsus variegatus* (紫岩蟹) collected at Fremantle, W. Australia.

This data frame contains the following columns:

sp: species - "B" or "O" for blue or orange.

sex: "M" or "F" for male or female

index: 1:50 within each of the four groups.

FL: carapace frontal lobe (lip) size (mm).

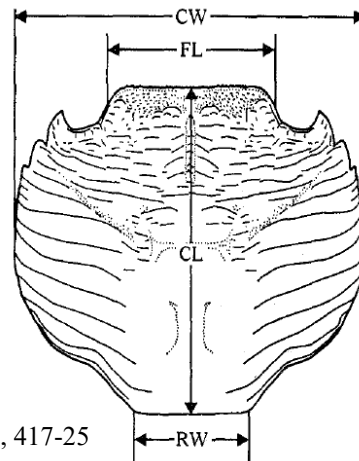
RW: carapace rear width (mm).

CL: carapace length (mm).

CW: carapace width (mm).

BD: body depth (mm).

```
> library(MASS)
> data(crabs)
```



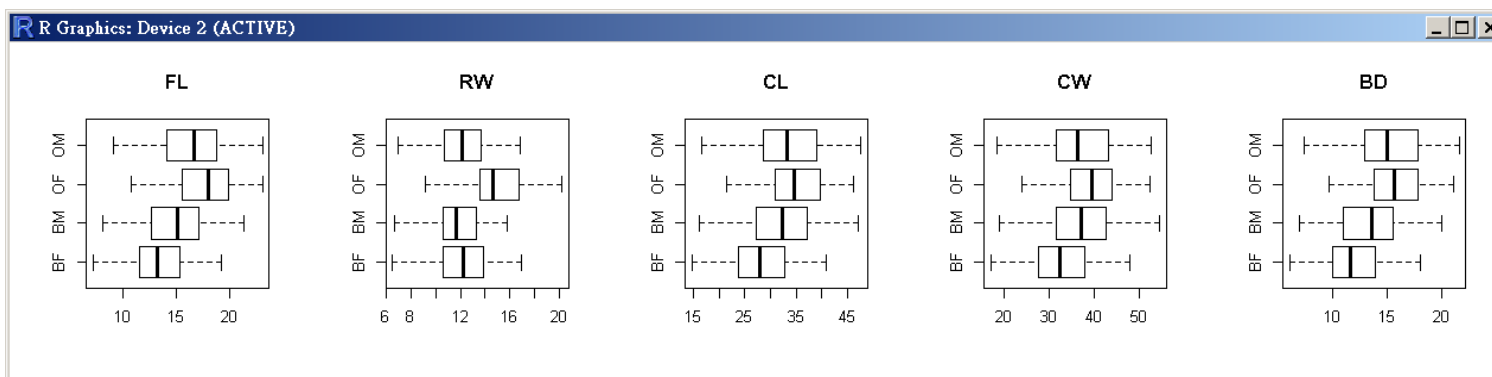
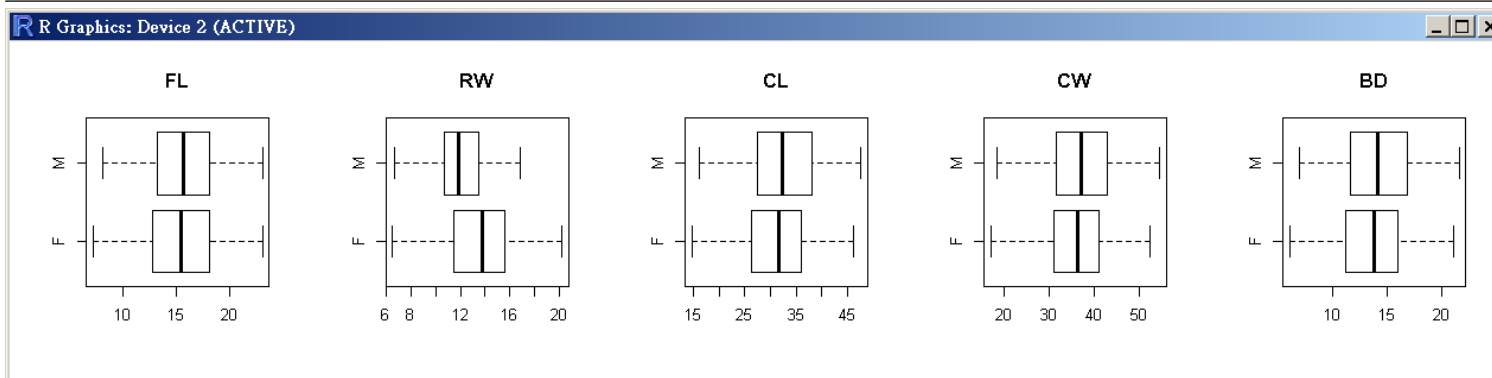
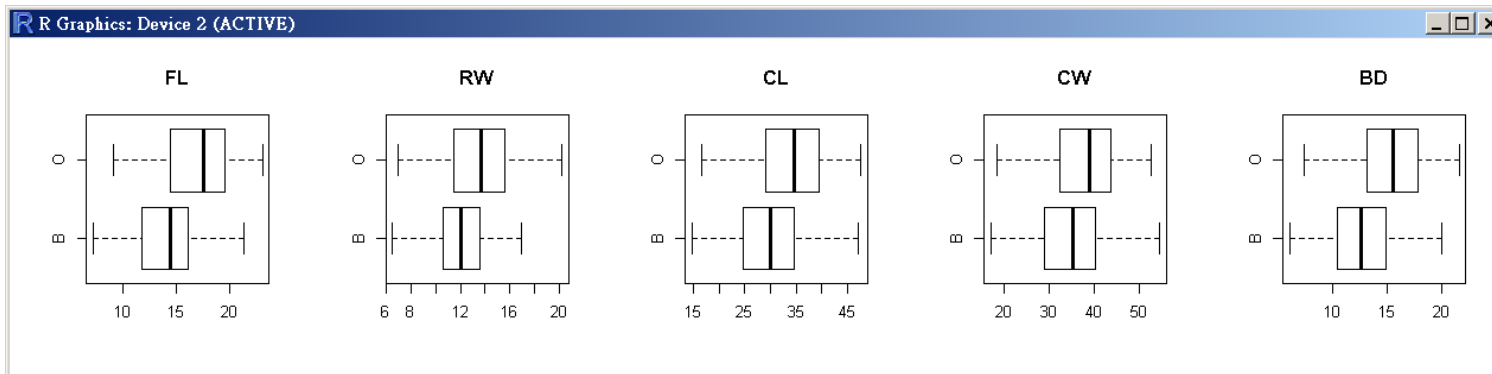
Aust. J. Zool. 1974, 22, 417-25



<http://www.qm.qld.gov.au/Find+out+about/Animals+of+Queensland/Crustaceans/Common+marine+crustaceans/Crabs/Purple+Swift-footed+Shore+Crab#.VhPWYiurFhs>

範例: Crab Data (2/4)

```
boxplot(crabs$FL~crabs$sp, main="FL", horizontal=T)
```

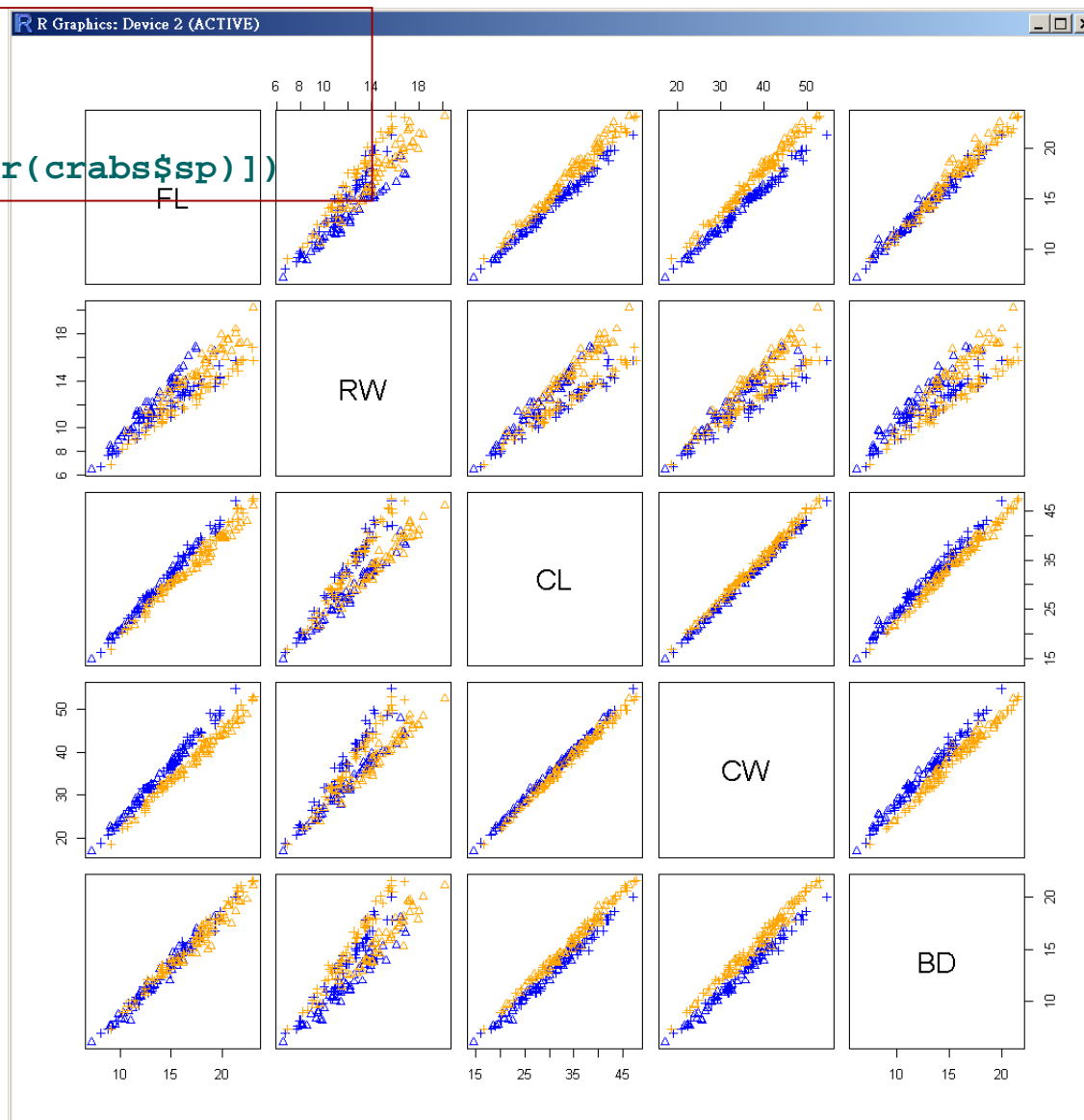


範例: Crab Data (3/4)

```
# tri: F, cross: M
pairs(crabs[,4:8],
pch=as.integer(crabs$sex)+1,
col=c("blue","orange")[as.integer(crabs$sp)])
```

- The analysis of ratios of body measurements is deeply ingrained in the taxonomic literature.
- Whether for plants or animals, certain ratios are commonly indicated in identification keys, diagnoses, and descriptions.

(Hannes Baur and Christoph Leuenberger, Analysis of Ratios in Multivariate Morphometry, Systematic Biology 60(6), 813-825.)



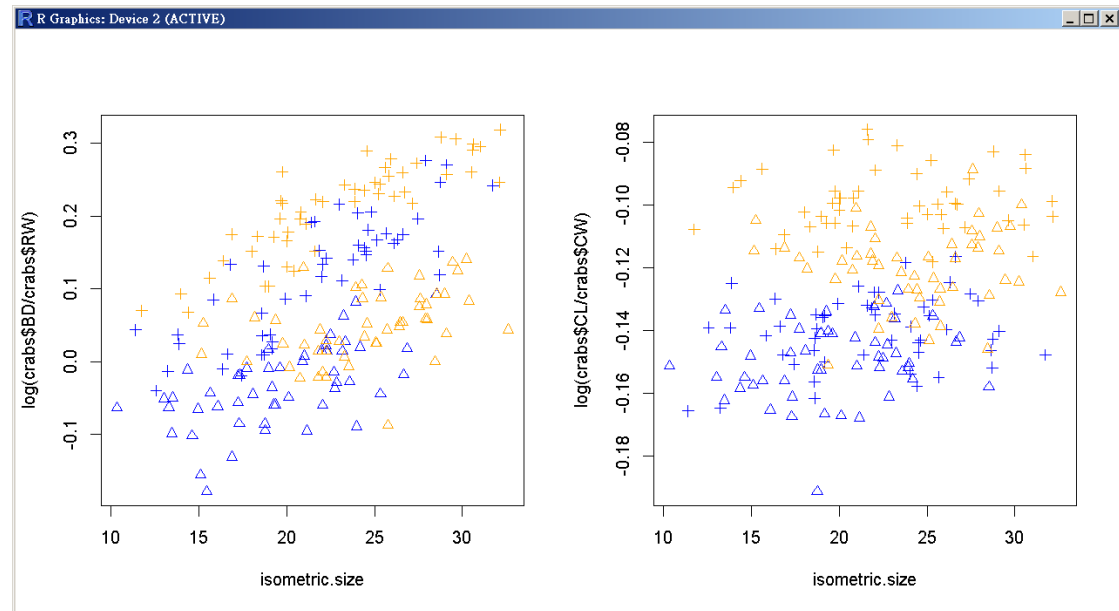
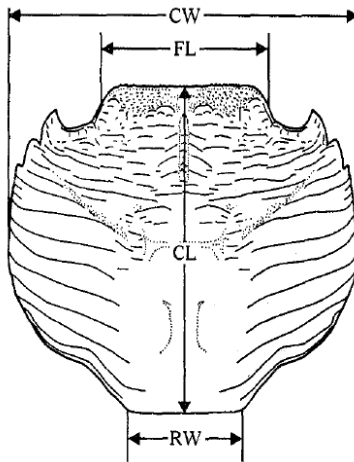
範例: Crab Data (4/4)

50/63

- The use of ratios of measurements (i.e., of body proportions), has a long tradition and is deeply ingrained in morphometric taxonomy.

Three size vectors have been commonly proposed in the literature:

- (a) isometric size (the arithmetic mean of x),
- (b) allometric size,
- (c) shape-uncorrelated size.



```
par(mfrow=c(1,2))
mp <- as.integer(crabs$sex)+1
mc <- c("blue","orange")[as.integer(crabs$sp)]
isometric.size <- apply(crabs[,4:8], 1, mean)
plot(isometric.size, log(crabs$BD/crabs$RW), pch=mp, col=mc)
plot(isometric.size, log(crabs$CL/crabs$CW), pch=mp, col=mc)
```




範例：房屋實價登錄資料

51/63

2014年臺灣資料分析競賽資料 (使用R軟體):
大約 682724筆紀錄，28個變數

1	檔案(xls)	縣市別	資料筆數	欄位數	1	鄉鎮市區	大安區	松山區
2	List_A	臺北市	54111	28	2	交易標的	房地(土地+建物)	房地(土地+建物)
3	List_B	臺中市	94683	28	3	土地區段位置/建物區段門牌	臺北市大安區和平東路xxx	臺北市松山區三民路xxx
4	List_C	基隆市	10833	28	4	土地移轉總面積(平方公尺)	19.39	35.53
5	List_D	臺南市	54643	28	5	使用分區或編定	住	住
6	List_E	高雄市	74565	28	6	非都市土地使用分區		
7	List_F	新北市	119719	28	7	非都市土地使用地		
8	List_G	宜蘭縣	16104	28	8	交易年月	10106	10107
9	List_H	桃園縣	95612	28	9	交易筆棟數	土地1建物2車位0	土地1建物1車位0
10	List_I	嘉義市	6840	28	10	移轉層次	五層	七層
11	List_J	新竹縣	23399	28	11	總樓層數	017	007
12	List_K	苗栗縣	17515	28	12	建物型態	住宅大樓(11層含以上有電梯)	華廈(10層含以下有電梯)
13	List_M	南投縣	13651	28	13	主要用途	國民住宅	國民住宅
14	List_N	彰化縣	22613	28	14	主要建材	鋼筋混凝土造	鋼筋混凝土造
15	List_O	新竹市	15664	28	15	建築完成年月	0740522	0810303
16	List_P	雲林縣	13171	28	16	建物移轉總面積(平方公尺)	100.98	146.66
17	List_Q	嘉義縣	9847	28	17	建物現況格局-房	3	3
18	List_T	屏東縣	17583	28	18	建物現況格局-廳	2	2
19	List_U	花蓮縣	11327	28	19	建物現況格局-衛	1	2
20	List_U	臺東縣	5825	28	20	建物現況格局-隔間	有	有
21	List_W	金門縣	2509	28	21	有無管理組織	有	有
22	List_X	澎湖縣	2423	28	22	總價(元)	18680000	25800000
23	List_Z	連江縣	87	28	23	單價(元/平方公尺)	184999	175917
					24	車位類別		
					25	車位移轉總面積(平方公尺)	0	0
					26	車位總價(元)	0	0
					27	交易標的橫坐標	305057	306980
					28	交易標的縱坐標	2768793	2771975



Species Data Transformation:

52/63

The Doubs Fish Data (1/2)

- Species abundances are dimensionally homogenous (expressed in the same physical units), quantitative (count, density, cover, biovolume, biomass, frequency, etc.) or semi-quantitative (classes) variables and restricted to positive or null values (zero meaning absence).
- For these, simple transformations may be used to reduce the importance of observations with very high values:
 - `sqrt()` (square root), `sqrt(sqrt())` (fourth root), or `log1p()` (natural logarithm of abundance + 1 to keep absence as zero).
 - In extreme cases, to give the same weight to all positive abundances irrespective of their values, the data can be transformed to binary 1-0 form (presence-absence).

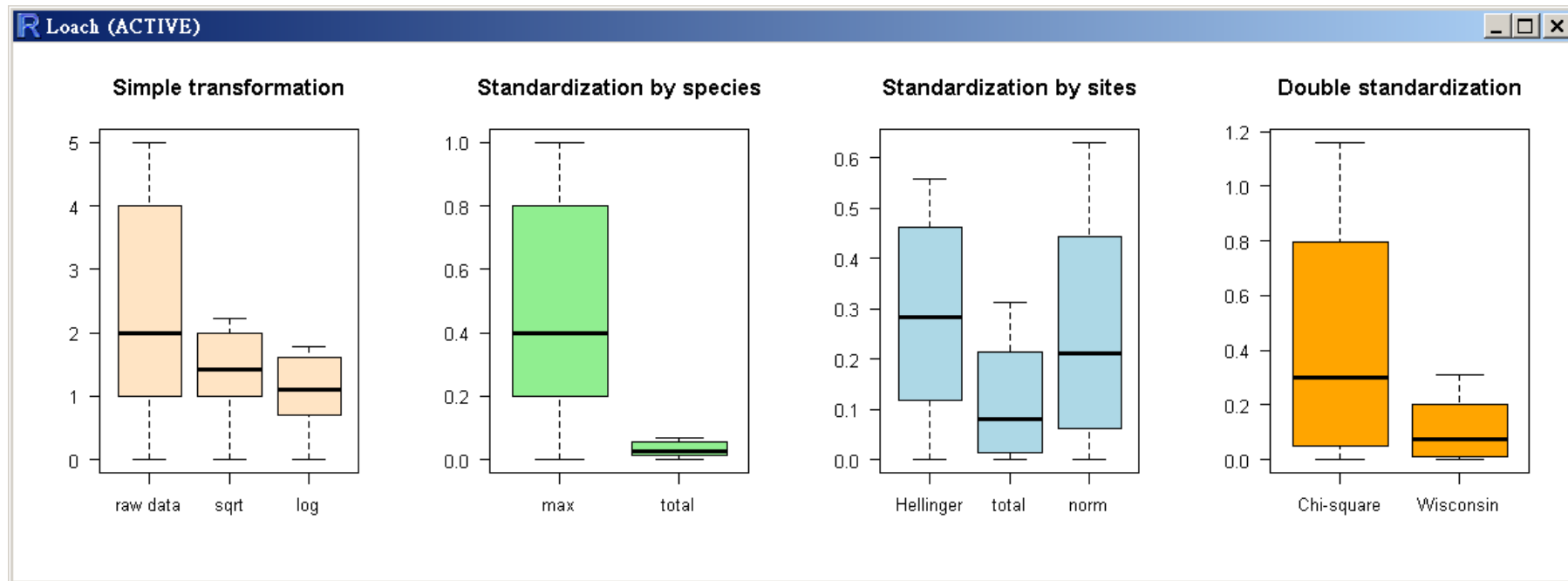


Species Data Transformation:

The Doubs Fish Data (2/2)

53/63

- The `decostand()` function of the `vegan` package provides many options for common standardization of ecological data.
- Standardization can be done relative to sites (site profiles), species (species profiles), or both (double profiles), depending on the focus of the analysis.



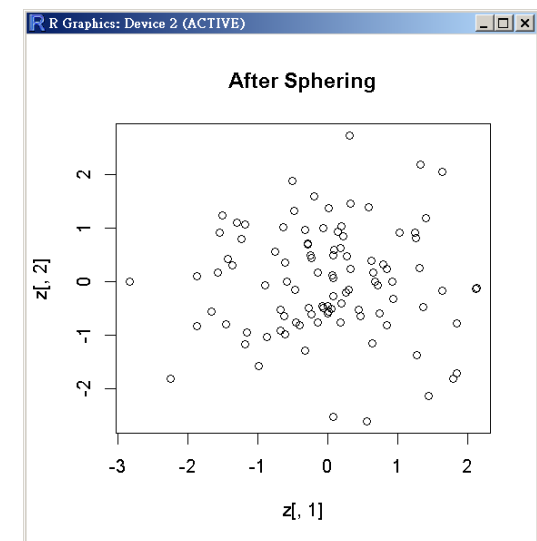
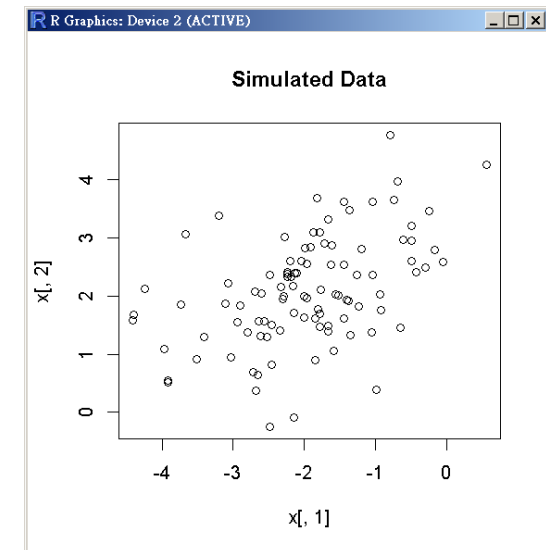


Sphering for Bivariate Variables

- A scatterplot of the 2-D multivariate normal random variables. Note that these are not centered at the origin, and the cloud is not spherical.
- The sphered data are now centered at the origin with a spherical spread. This is similar to the z-score standardization in 1-D.

```
n <- 100
mu <- c(-2, 2)
sigma <- matrix(c(1, 0.5, 0.5, 1), ncol=2)
library(MASS)
x <- mvrnorm(n, mu, sigma)
plot(x[,1], x[,2], main="Simulated Data")

x.bar <- colMeans(x)
ei <- eigen(cov(x))
D <- diag(ei$values)
V <- ei$vectors
xc <- x - matrix(rep(1, n), ncol=1)%*%x.bar
z <- xc%*%V%*%diag((ei$values)^{-1/2})
plot(z[,1], z[,2], main="After Sphering")
```





Sphering for Multivariate Variables

55/63

- The transformed variables will have a p -dimensional mean of 0 and a covariance matrix given by the identity matrix.

Sliced Inverse Regression for Dimension Reduction

KER-CHAU LI*

© 1991 American Statistical Association
Journal of the American Statistical Association
June 1991, Vol. 86, No. 414, Theory and Methods

$$y = f(\beta_1 \mathbf{x}, \beta_2 \mathbf{x}, \dots, \beta_K \mathbf{x}, \epsilon). \quad (1.1)$$

$$\hat{\Sigma}_{\mathbf{xx}} = \frac{1}{n-1} \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T$$

An $n \times n$ matrix A is **diagonalizable** if there is a matrix V and a diagonal matrix D such that $A = VDV^{-1}$. This happens if and only if A has n **eigenvectors** which constitute a basis for \mathbb{C}^n . In this case, V can be chosen to be the matrix with the n eigenvectors as columns, and thus a square root of A is

$$R = VSV^{-1},$$

where S is any square root of D .

A in $\mathbb{R}^{n \times n}$

- Cholesky Decomposition: $A = U^T U$
- LU Decomposition: $A = LU$
- QR Decomposition: $A = QR$, Q : orthonormal, R : upper triangular
- Singular Value Decomposition: $A = VDV^T$

4. SLICED INVERSE REGRESSION

A scheme for sliced inverse regression operates on the data (y_i, \mathbf{x}_i) ($i = 1, \dots, n$), in the following way:

1. Standardize \mathbf{x} by an affine transformation to get $\tilde{\mathbf{x}}_i = \hat{\Sigma}_{\mathbf{xx}}^{-1/2} (\mathbf{x}_i - \bar{\mathbf{x}})$ ($i = 1, \dots, n$), where $\hat{\Sigma}_{\mathbf{xx}}$ and $\bar{\mathbf{x}}$ are the sample covariance matrix and sample mean of \mathbf{x} respectively.

2. Divide range of y into H slices, I_1, \dots, I_H ; let the proportion of the y_i that falls in slice h be \hat{p}_h ; that is $\hat{p}_h = (1/n) \sum_{i=1}^n \delta_h(y_i)$, where $\delta_h(y_i)$ takes the values 0 or 1 depending on whether y_i falls into the h th slice I_h or not.

3. Within each slice, compute the sample mean of the $\tilde{\mathbf{x}}_i$'s, denoted by \hat{m}_h ($h = 1, \dots, H$), so that $\hat{m}_h = (1/n\hat{p}_h) \sum_{y_i \in I_h} \tilde{\mathbf{x}}_i$.

4. Conduct a (weighted) principal component analysis for the data \hat{m}_h ($h = 1, \dots, H$) in the following way: Form the weighted covariance matrix $\hat{V} = \sum_{h=1}^H \hat{p}_h \hat{m}_h \hat{m}_h'$, then find the eigenvalues and the eigenvectors for \hat{V} .

5. Let the K largest eigenvectors (row vectors) be $\hat{\eta}_k$ ($k = 1, \dots, K$). Output $\hat{\beta}_k = \hat{\eta}_k \hat{\Sigma}_{\mathbf{xx}}^{-1/2}$ ($k = 1, \dots, K$).



Which Transformation? (1/2)

- It is better to use a transformation that other researchers **commonly use in your field**, such as the square-root transformation for count data or the log transformation for size data.
- Even if an obscure transformation that not many people have heard of gives you slightly more normal or more homoscedastic data, it will probably be better to use a more common transformation so people don't get suspicious.
- Remember that your data don't have to be perfectly normal and homoscedastic; parametric tests aren't extremely sensitive to deviations from their assumptions.
- It is also important that you decide which transformation to use before you do the statistical test. **Trying different transformations until you find one that gives you a significant result is cheating.**
- If you have a large number of observations, compare the effects of different transformations on the normality and the homoscedasticity of the variable.
- If you have a small number of observations, you may not be able to see much effect of the transformations on the normality and homoscedasticity; in that case, you should use whatever transformation people in your field routinely use for your variable.

<http://www.biostathandbook.com/transformation.html>



Which Transformation? (2/2)

- The main criterion in choosing a transformation is: **what works with the data?**

It is important to consider as well two questions:

- **What makes physical (biological, economic, whatever) sense**, for example in terms of limiting behaviour as values get very small or very large? This question often leads to the use of logarithms.
- **Can we keep dimensions and units simple and convenient?**
 - Prefer measurement scales that are easy to think about.
 - Simplify: The cube root of a volume and the square root of an area both have the dimensions of length. Reciprocals usually have simple units.



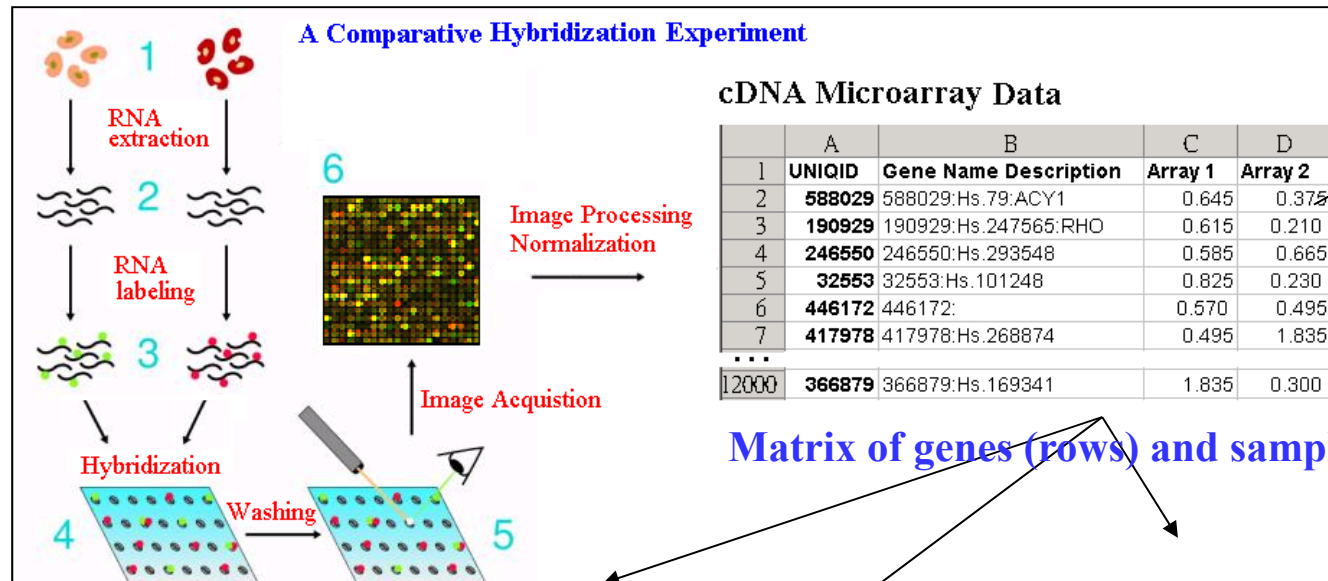
Psychological Comments

- Although transformed scales may seem less natural, this is largely a psychological objection. Greater experience with transformation tends to reduce this feeling, simply because transformation so often works so well.
- In fact, many familiar measured scales are really transformed scales: decibels, pH and the Richter scale of earthquake magnitude are all **logarithmic**.
- However, transformations cause debate even among experienced data analysts:
 - "This seems like a kind of cheating. You don't like how the data are, so you decide to change them."
 - "I see that this is a clever trick that works nicely. But how do I know when this trick will work with some other data, or if another trick is needed, or if no transformation is needed?"
 - "Transformations are needed because there is no guarantee that the world works on the scales it happens to be measured on."
 - "Transformations are most appropriate when they match a scientific view of how a variable behaves."
 - Often it helps to transform results back again, using the reverse or inverse transformation.
- **Back transformation:** Even though you've done a statistical test on a transformed variable, such as the log of fish abundance, it is not a good idea to report your means, standard errors, etc. in transformed units.

cDNA Microarray Gene Expression Data

微陣列資料統計分析 Statistical Microarray Data Analysis

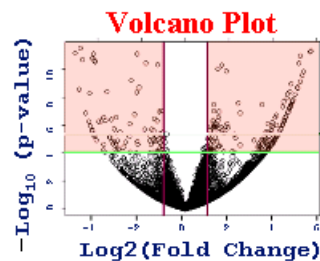
<http://www.hmwu.idv.tw/index.php/mada>



Discovery of differentially expressed genes

Parametric: t-test

Non-parametric: Wilcoxon, M

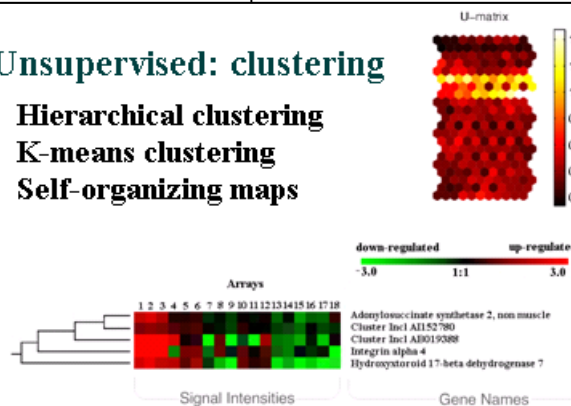


Unsupervised: clustering

Hierarchical clustering

K-means clustering

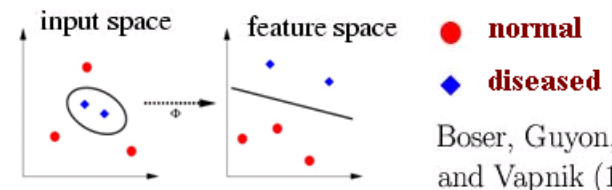
Self-organizing maps



Supervised: classification

- Linear discriminants
- Decision trees
- Support vector machines

Support Vector Classifiers





What is Normalization?

- **Non-biological factor** can contribute to the variability of data, in order to reliably compare data from **multiple probe arrays**, differences of non-biological origin must be minimized.
- Normalization is a process of **reducing unwanted variation** across chips.
- It may use information from multiple chips.
 - Normalization corrects for overall chip brightness and other factors that may influence the numerical value of expression intensity,
 - Enabling the user to more confidently **compare gene expression** estimates between samples.



Why Normalization?

- **Main idea**
 - Remove the systematic bias in the data as completely possible while preserving the variation in the gene expression that occurs because of biologically relevant changes in transcription.
- **Assumption**
 - The average gene **does not change** in its expression level in the biological sample being tested.
 - **Most genes** are not differentially expressed or up- and down-regulated genes roughly **cancel out** the expression effect.



Normalization Methods for cDNA Microarray Data

62/63

- **Within-Array Normalization: location**
 - Correcting for Different Responses of the Cy3 and Cy5 Channels
 - Linear Regression of Cy5 Against Cy3 (**Global Normalization**)
 - Linear Regression of Log Ratio Against Average Intensity
 - Nonlinear Regression of Log Ratio Against Average Intensity (**Lowess Normalization**)
 - Correcting for Spatial Effects
 - Two-Dimensional Lowess Regression
 - Block-Block Loess Regression (**Within Print-tip Group Normalization**)
- **Within-Array Normalization: scale**
- **Between-Array Normalization**
 - To enable comparison of multiple arrays
 - Centering, Scaling, Distribution Normalization
- **Paired-slides Normalization**
(dye swap Experiments) (**Slef-normalization**)



Case Studies Using R/BioConductor

63/63

Statistical Microarray Data Analysis

微陣列資料統計分析

Version: July 01, 2009

Han-Ming Wu (吳漢銘)

Department of Mathematics, Tamkang University

<http://www.hmwu.idv.tw>

hmwu@mail.tku.edu.tw



Statistical Microarray Data Analysis

Chapter 2

Preprocessing Two-Color Spotted Microarray

Statistical Microarray Data Analysis

Appendix 2

Lab: Preprocessing of Two-Color Spotted Microarray

Statistical Microarray Data Analysis

Chapter 3

Preprocessing Affymetrix GeneChip Microarray

Statistical Microarray Data Analysis

Appendix 3

Lab: Preprocessing Affymetrix GeneChip Microarray Data