Linux: basic usage

Isheng Jason Tsai

Introduction to NGS Data and Analysis
Lecture 2-1

Lecture objective

Introduction

Linux intro
Some basics

R intro
Some basics

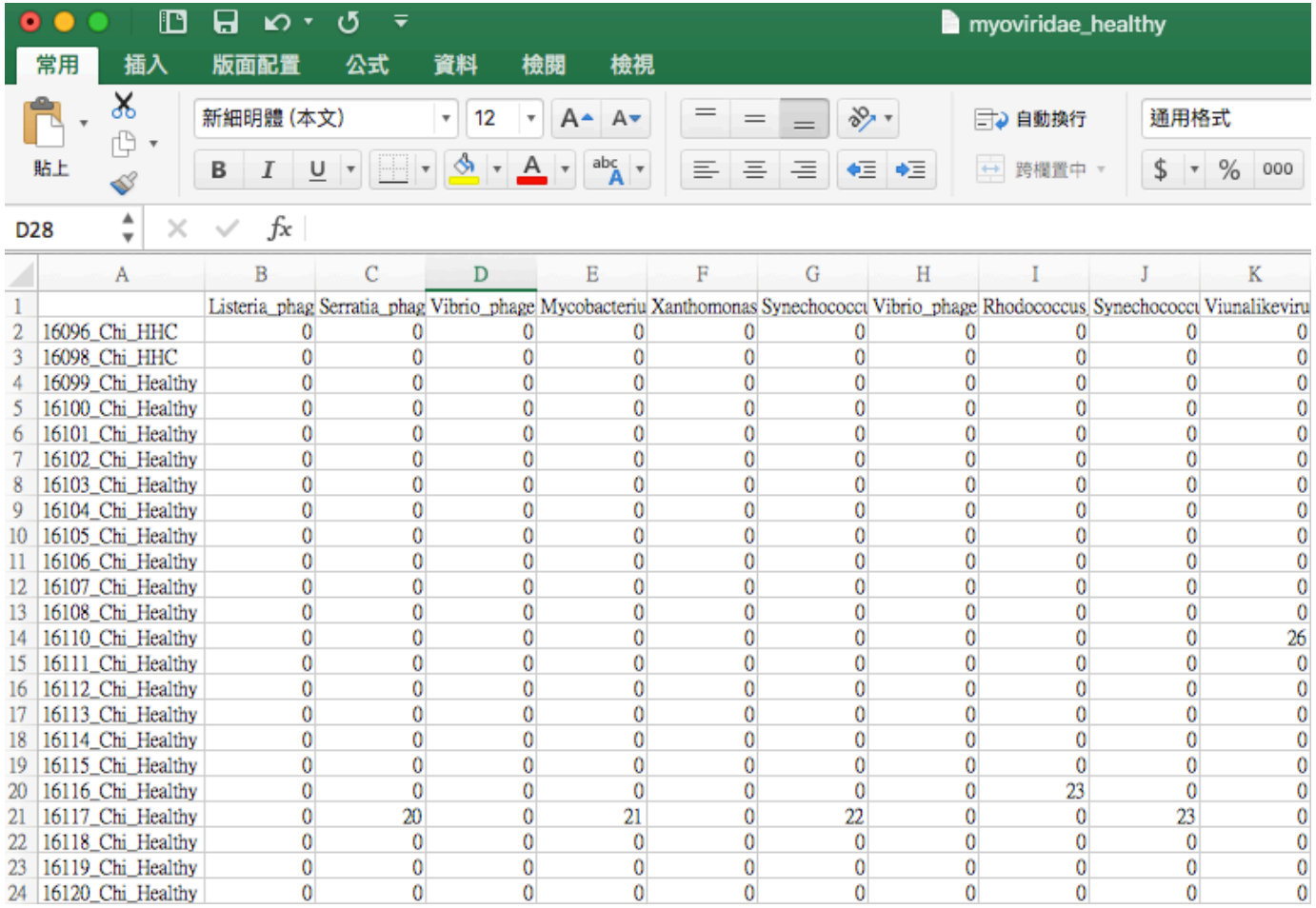
What do we actually do everyday?

- You have got some new sequenced data!
  - **(1) Need to understand, QC, and analyse the data**. How?

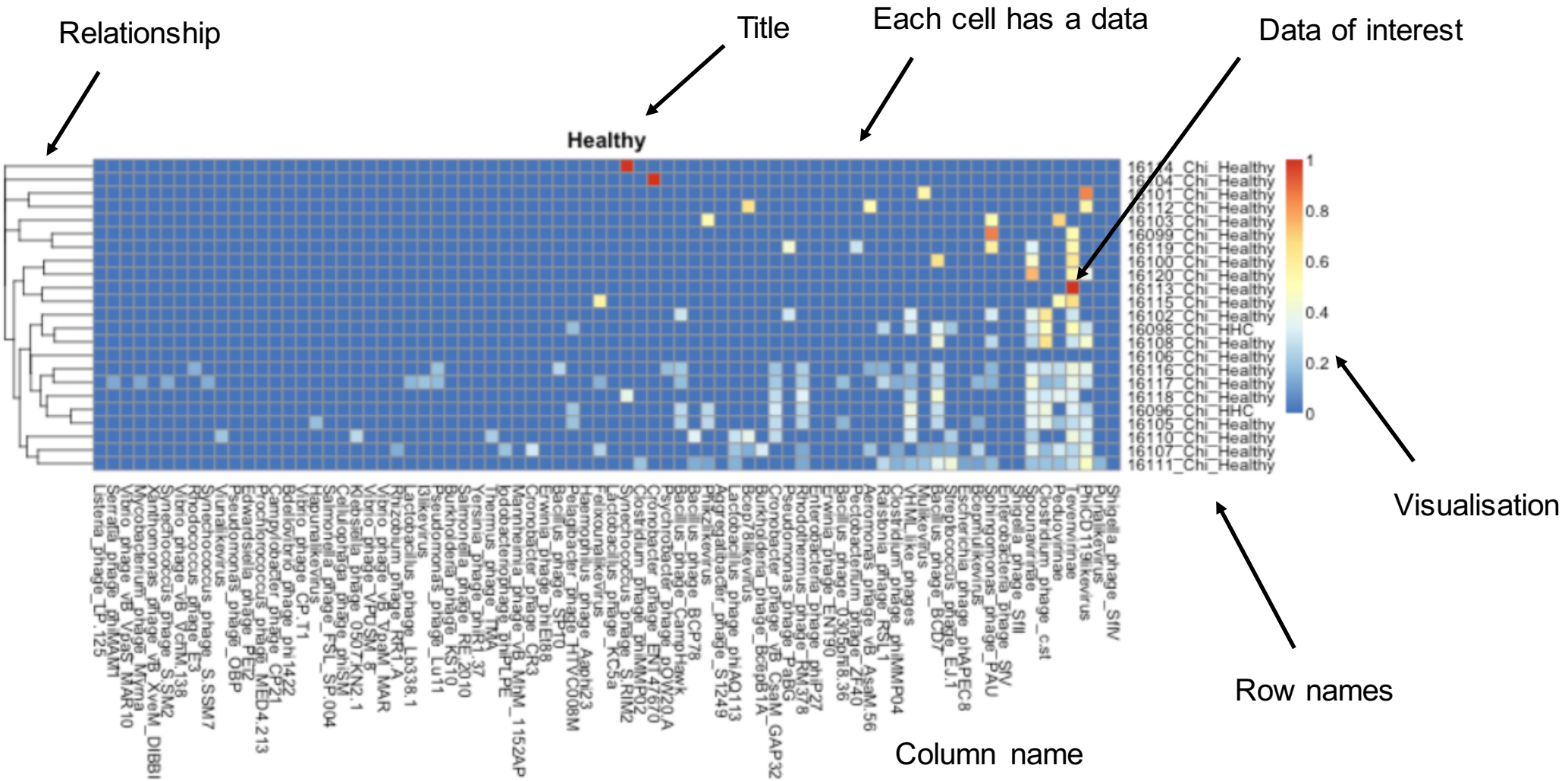
- Once the data has been analysed, you need to compare against published ones
  - **(2)** You need to survey, and download the right dataset
  - Move to step **(1)**

- **(3)** Then you need to **visualise** and **present**
- Does it answer your question? There are times when you need to
  (4) develop new/better algorithms and
  (5) generate more data

When most people think of visualisation

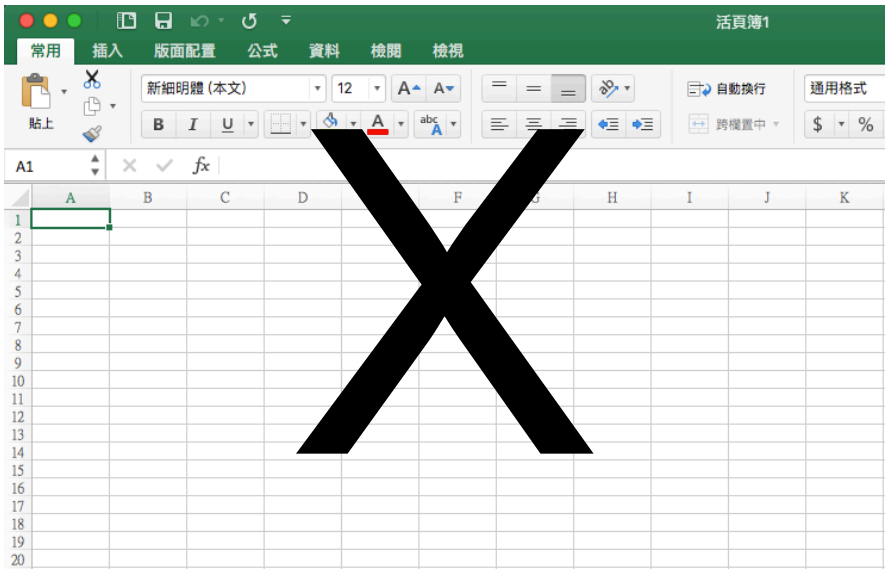


	A	B	C	D	E	F	G	H	I	J	K
1		Listeria_phag	Serratia_phag	Vibrio_phage	Mycobacteriu	Xanthomonas	Synechococcu	Vibrio_phage	Rhodococcus	Synechococcu	Viunalikeviru
2	16096_Chi_HHC	0	0	0	0	0	0	0	0	0	0
3	16098_Chi_HHC	0	0	0	0	0	0	0	0	0	0
4	16099_Chi_Healthy	0	0	0	0	0	0	0	0	0	0
5	16100_Chi_Healthy	0	0	0	0	0	0	0	0	0	0
6	16101_Chi_Healthy	0	0	0	0	0	0	0	0	0	0
7	16102_Chi_Healthy	0	0	0	0	0	0	0	0	0	0
8	16103_Chi_Healthy	0	0	0	0	0	0	0	0	0	0
9	16104_Chi_Healthy	0	0	0	0	0	0	0	0	0	0
10	16105_Chi_Healthy	0	0	0	0	0	0	0	0	0	0
11	16106_Chi_Healthy	0	0	0	0	0	0	0	0	0	0
12	16107_Chi_Healthy	0	0	0	0	0	0	0	0	0	0
13	16108_Chi_Healthy	0	0	0	0	0	0	0	0	0	0
14	16110_Chi_Healthy	0	0	0	0	0	0	0	0	0	26
15	16111_Chi_Healthy	0	0	0	0	0	0	0	0	0	0
16	16112_Chi_Healthy	0	0	0	0	0	0	0	0	0	0
17	16113_Chi_Healthy	0	0	0	0	0	0	0	0	0	0
18	16114_Chi_Healthy	0	0	0	0	0	0	0	0	0	0
19	16115_Chi_Healthy	0	0	0	0	0	0	0	0	0	0
20	16116_Chi_Healthy	0	0	0	0	0	0	0	23	0	0
21	16117_Chi_Healthy	0	20	0	21	0	22	0	0	23	0
22	16118_Chi_Healthy	0	0	0	0	0	0	0	0	0	0
23	16119_Chi_Healthy	0	0	0	0	0	0	0	0	0	0
24	16120_Chi_Healthy	0	0	0	0	0	0	0	0	0	0


Relationship
Title
Each cell has a data
Data of interest
Healthy
16114_Chi_Healthy
16104_Chi_Healthy
16101_Chi_Healthy
16112_Chi_Healthy
16103_Chi_Healthy
16099_Chi_Healthy
16119_Chi_Healthy
16100_Chi_Healthy
16120_Chi_Healthy
16113_Chi_Healthy
16115_Chi_Healthy
16102_Chi_Healthy
16098_Chi_HHC
16108_Chi_Healthy
16106_Chi_Healthy
16116_Chi_Healthy
16117_Chi_Healthy
16118_Chi_Healthy
16096_Chi_HHC
16105_Chi_Healthy
16110_Chi_Healthy
16107_Chi_Healthy
16111_Chi_
1
0.8
0.6
0.4
0.2
0
Visualisation
Row names
Column name

So what do you need?

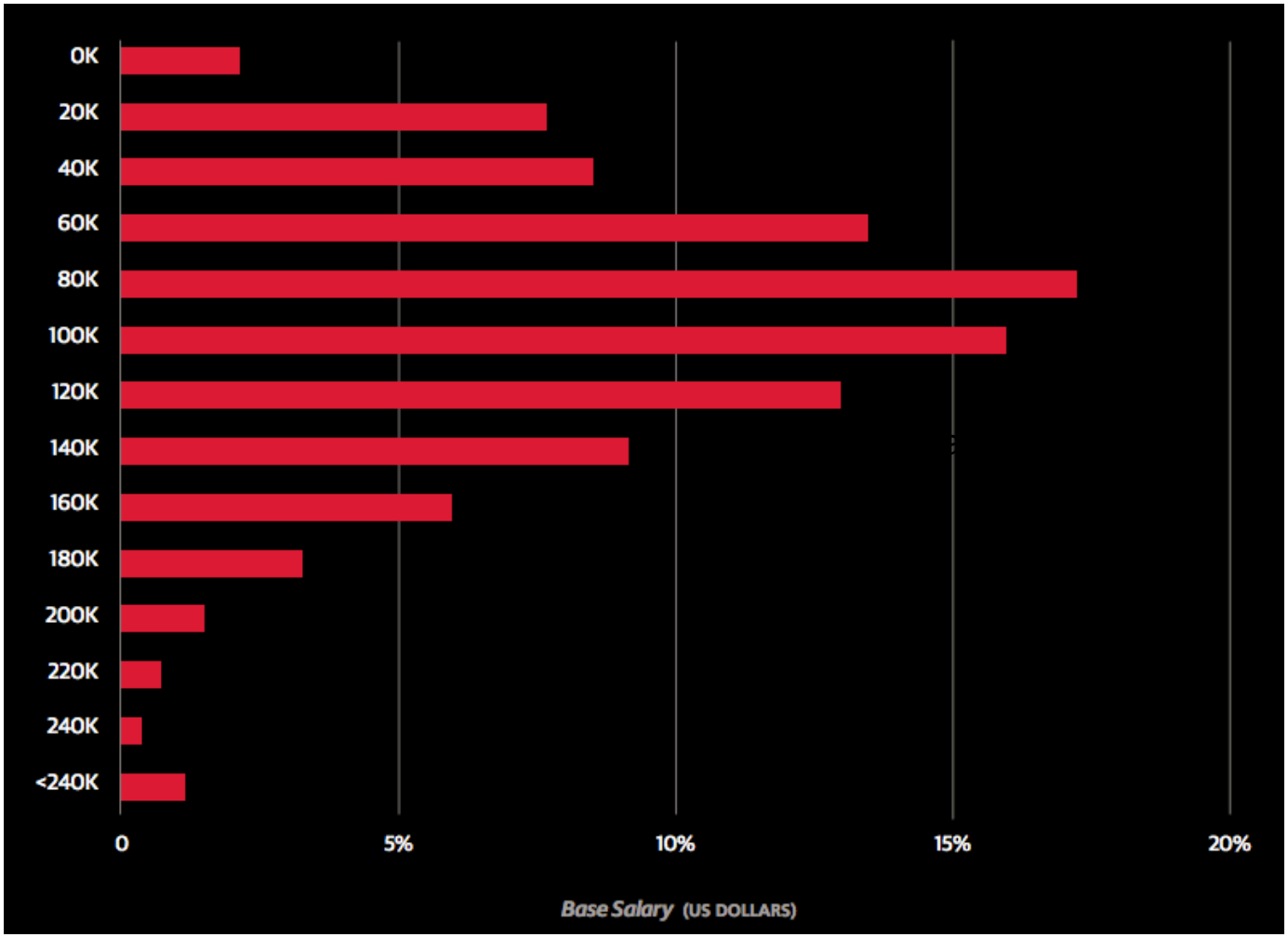
You need a platform to rearrange, tidy, subset, merge data easily



Recommendation:
R and Python in a linux environment

2015 Data science survey

In Taiwan

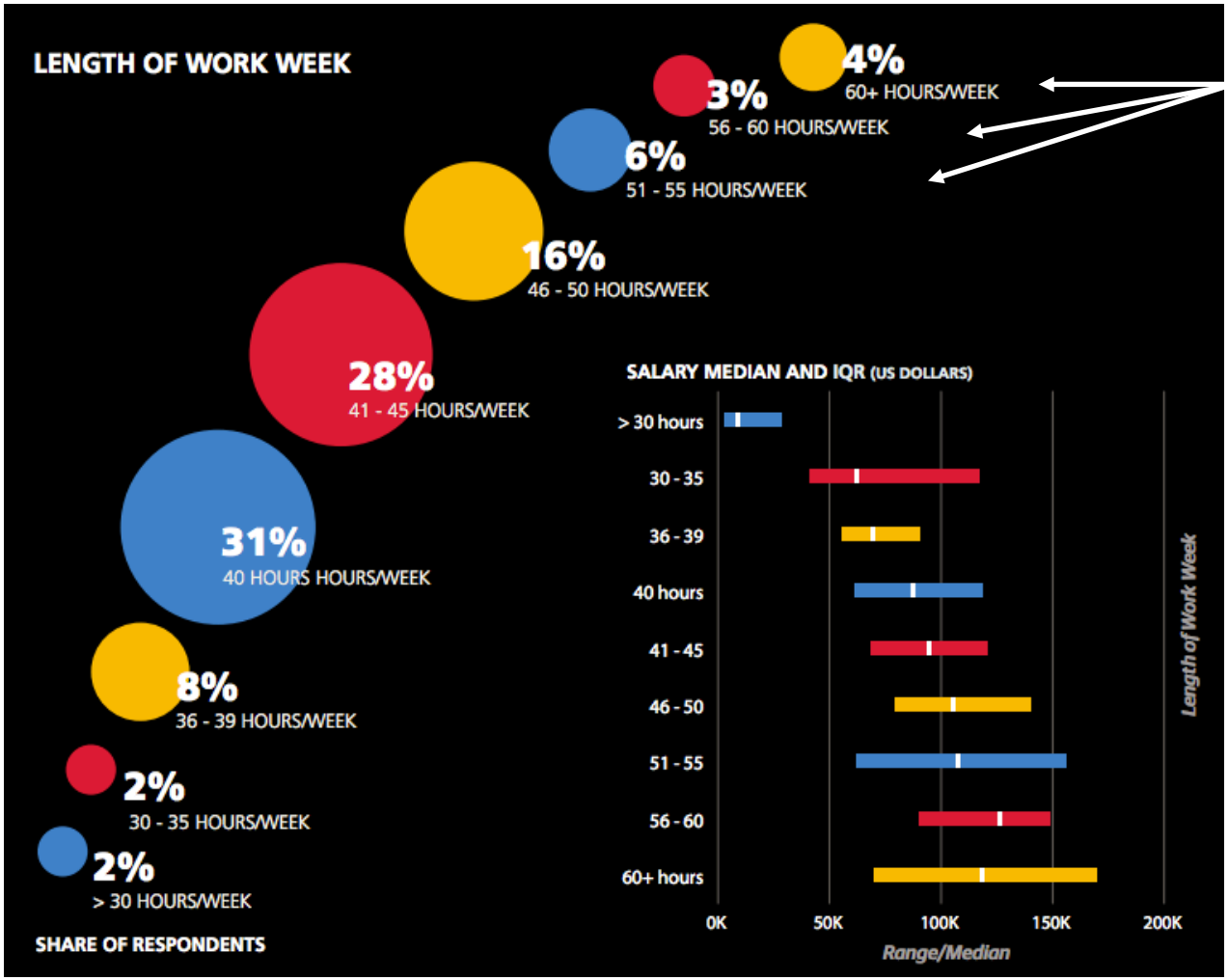


← PI salary in Taiwan

← TSMC PhD 33 salary

https://www.oreilly.com/ideas/2015-data-science-salary-survey

2015 Data science survey

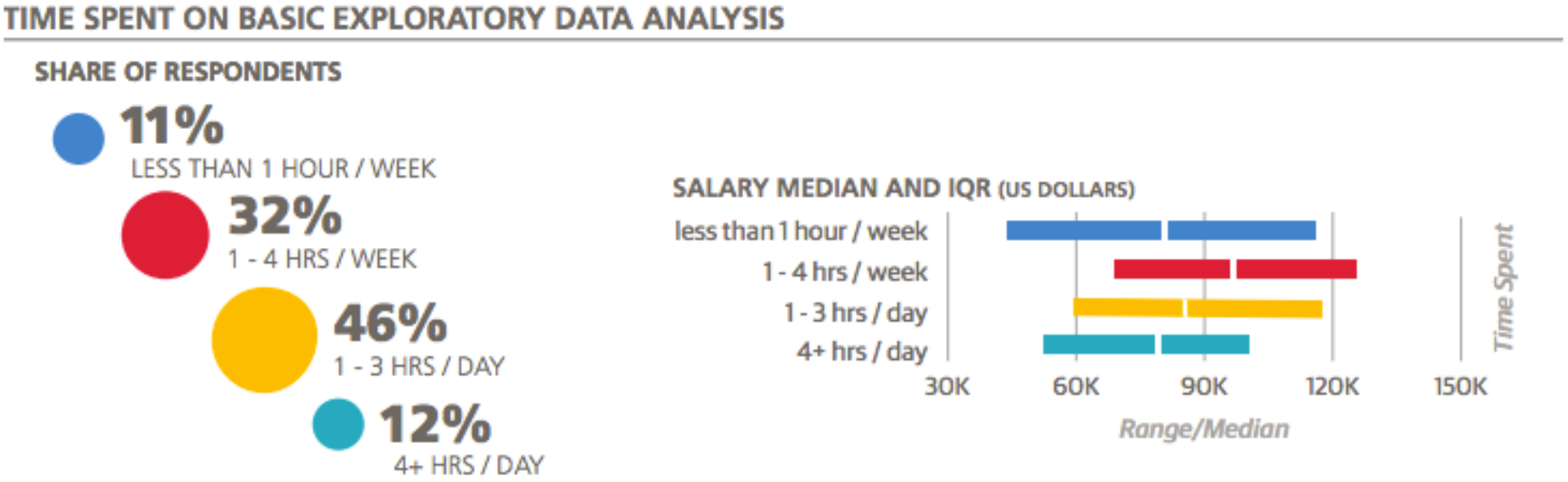


https://www.oreilly.com/ideas/2015-data-science-salary-survey

In Taiwan

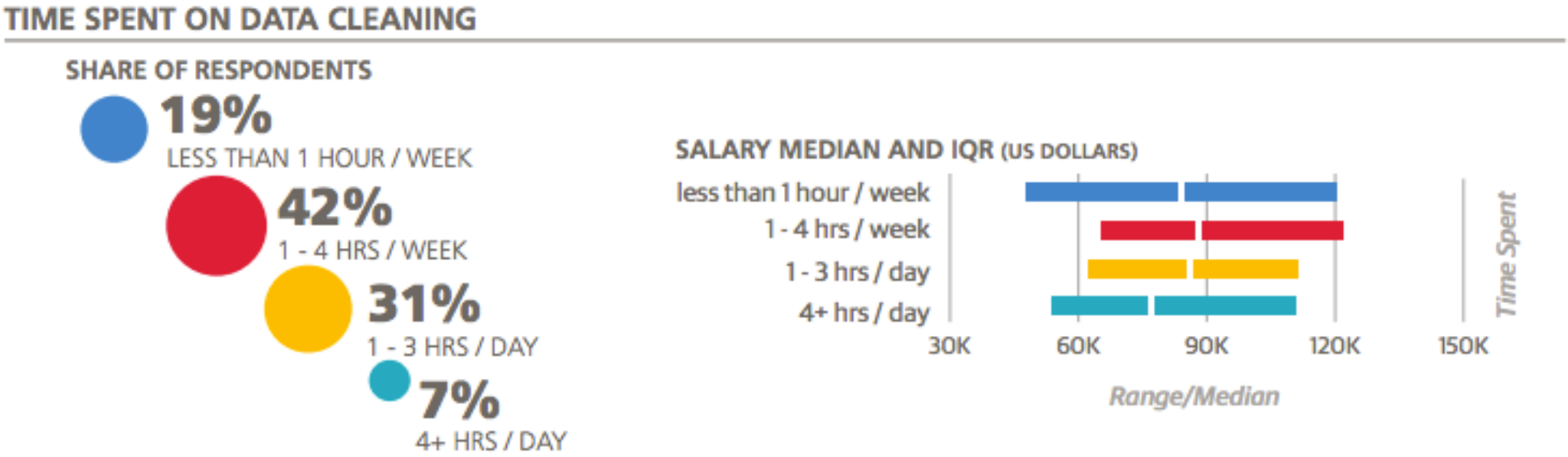
PhD/RA/PI working hour

2015 Data science survey



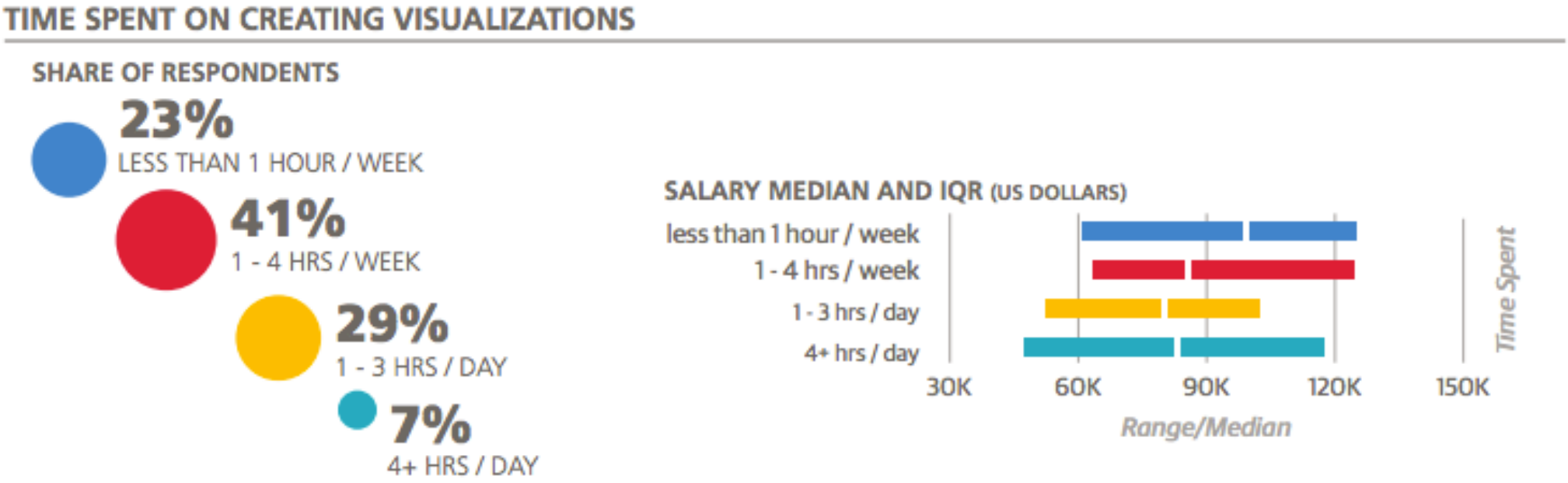
https://www.oreilly.com/ideas/2015-data-science-salary-survey

2015 Data science survey



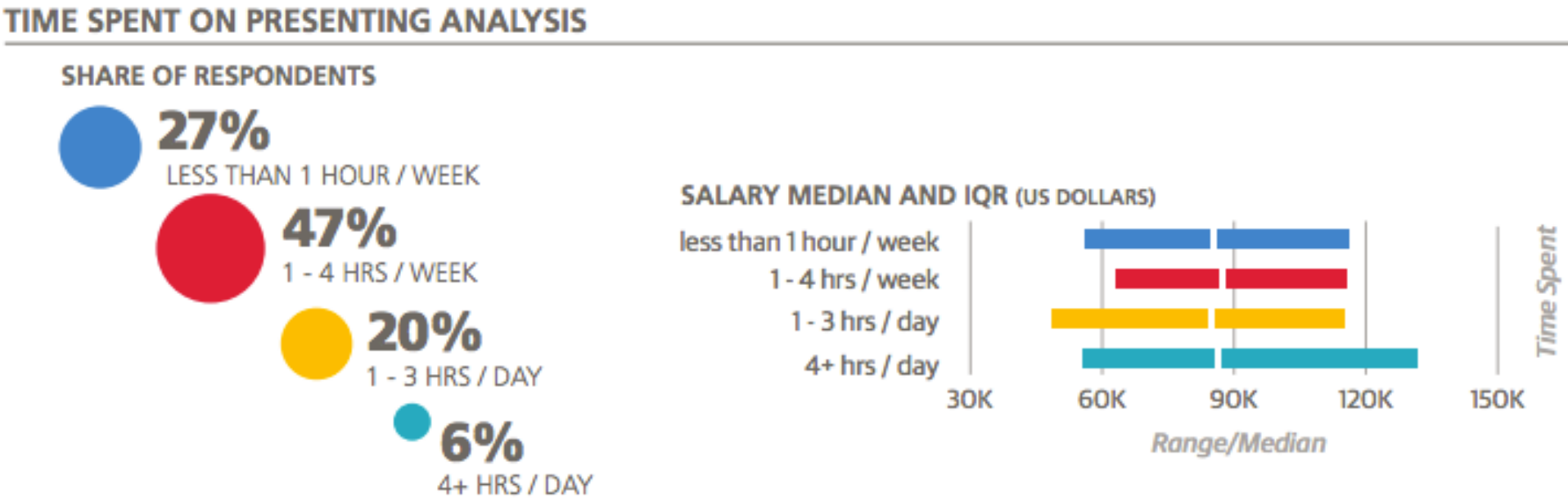
https://www.oreilly.com/ideas/2015-data-science-salary-survey

2015 Data science survey



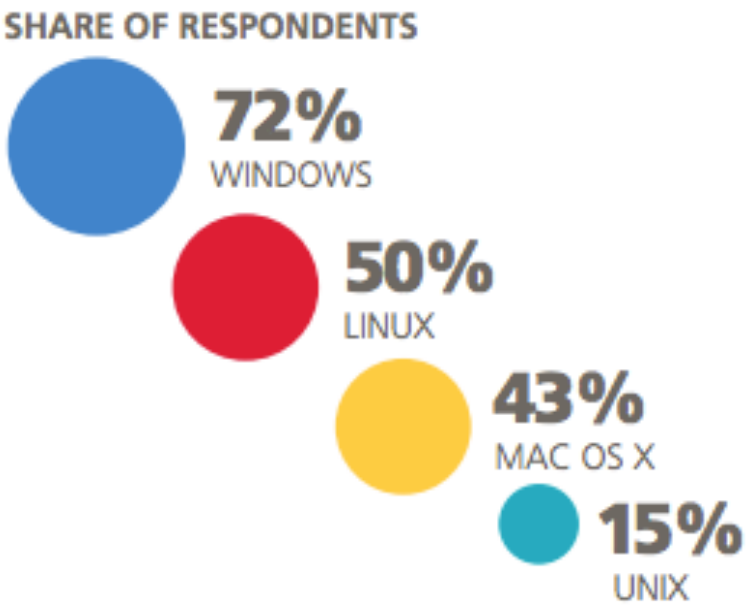
https://www.oreilly.com/ideas/2015-data-science-salary-survey

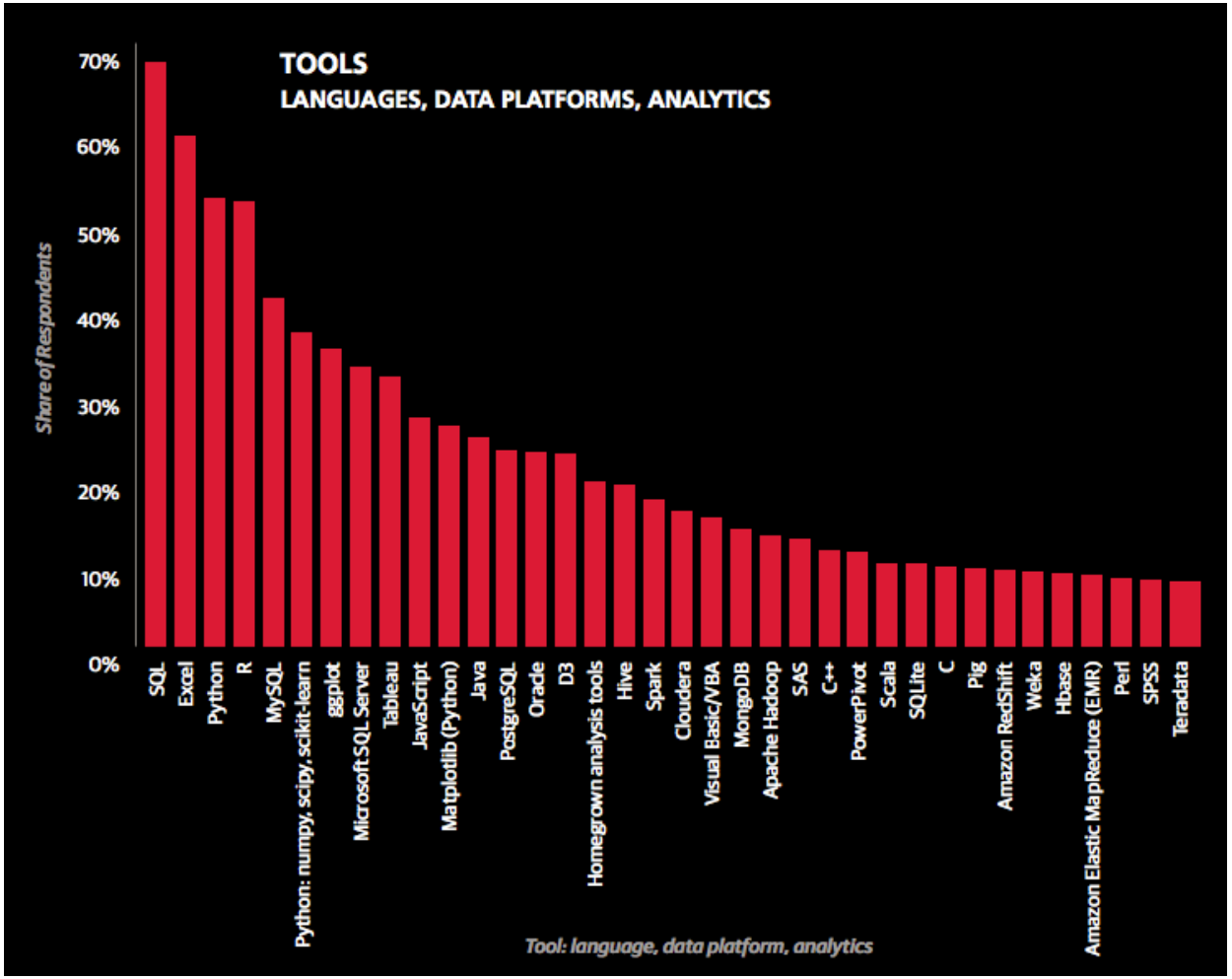
2015 Data science survey



https://www.oreilly.com/ideas/2015-data-science-salary-survey

2015 Data science survey





https://www.oreilly.com/ideas/2015-data-science-salary-survey

Some observations

A day of a data scientist /bioinformatician / biologist with lots of data:

- **Less than 1 to 4 hours** to quickly explore data (78%)
- **Less than 1 to 4 hours** to do data cleaning (74%)
- **Less than 1 to 4 hours** to visualise data (70%)
- **Less than 1 to 4+ hours** to present analysis (73%)

= 4 – 16 hours to finish your daily task

Some observations (my own opinions)

- Data scientist are needed everywhere
- Bioinformatician / data scientist in Biology field are poorly paid in relative to other field,
  - even more apparent in Taiwan -> so not many bioinformatician left in Taiwan

This will result in

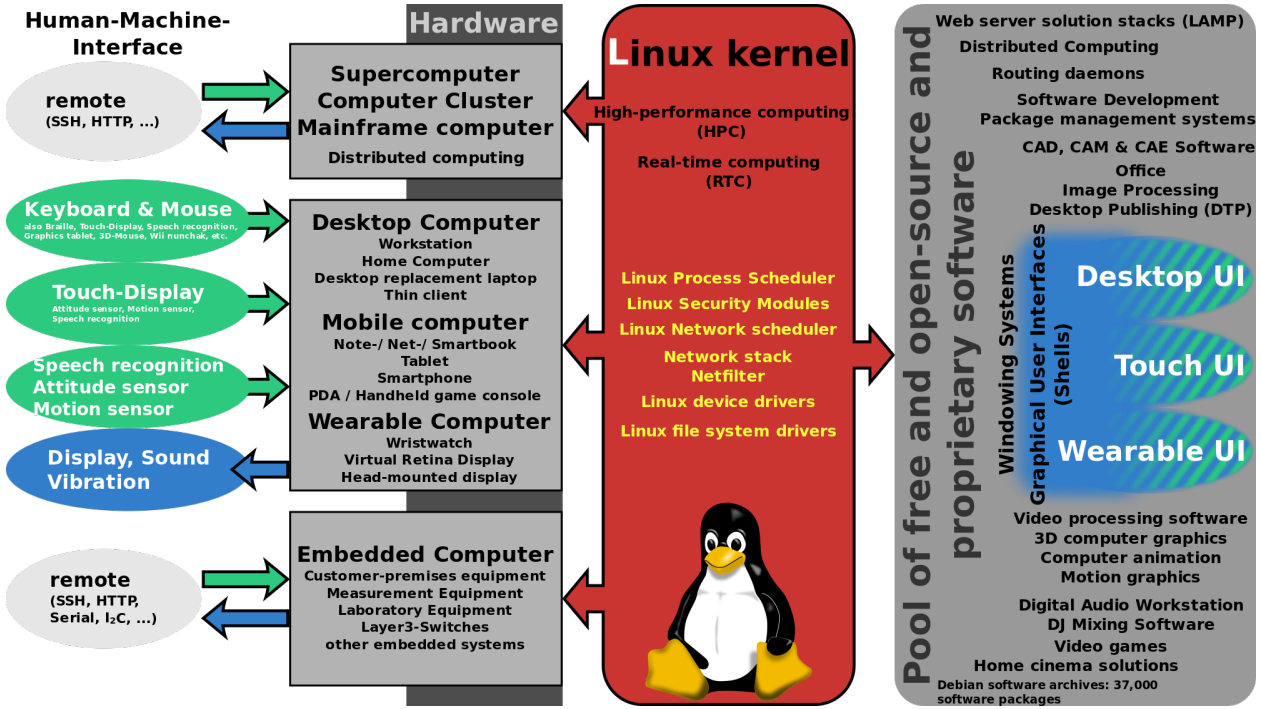
- Either all high throughput data are outsourced to companies outsourced to companies abroad (BGI?) -> **Taiwan will not gain the experience**
- A few labs can enjoy deal with all the data in Taiwan -> also not good as no energy to initiate novel projects

Bioinformaticians / biologists will data skills should be more appreciated

Linux

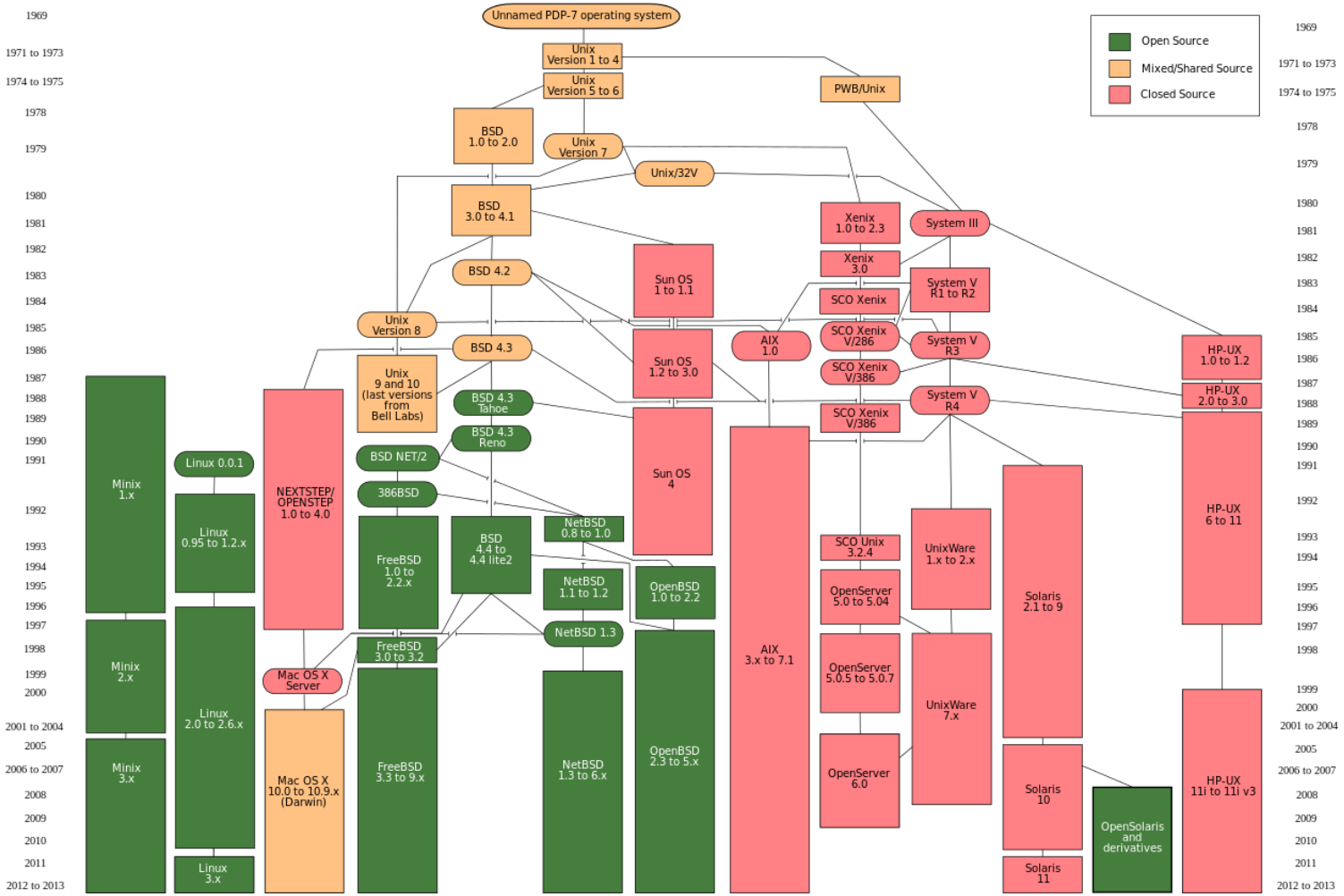
What is Linux?

Linux is a **Unix-like** computer **operating system (OS)** assembled under the model of free and open-source software development and distribution.



Wiki

History of Unix



Wiki

Linux distributions

A **Linux distribution** (often called a distro for short) is an operating system made from **a software collection**, which is based upon the Linux kernel and, often, a package management system.

網頁參觀排名		
資料範圍: Last 6 months 出發		
名次	發行版	H.P.D*
1	Mint	3169–
2	Debian	2100–
3	Ubuntu	1647▲
4	openSUSE	1513▲
5	Fedora	1158–
6	Manjaro	1075▲
7	Mageia	973▼
8	CentOS	890–
9	Arch	829–
10	Android-x86	797–
11	Zorin	737▲
12	Kali	711▼
13	PCLinuxOS	637▲
14	LXLE	616▼
15	Puppy	614▼
16	deepin	604▲
17	Lite	598–
18	Ubuntu MATE	596▲
19	elementary	566▲
20	Lubuntu	566–
21	antiX	503▲
22	Slackware	477–

Wiki

Console and Command-line interface

Computer terminal or system consoles are the **text entry and display device** for system administration messages, particularly those from the BIOS or boot loader, the kernel, from the init system and from the system logger. It is a **physical device consisting of a keyboard and a screen**.

A **command-line interface** is a means of interacting with a computer program where the **user** issues **commands** to the program (putty, terminal) in the form of successive lines of text (command lines).

http://www.ictlounge.com/html/operating_systems.htm Wiki

Console and Command-line interface

Lab

Matrix

Wiki

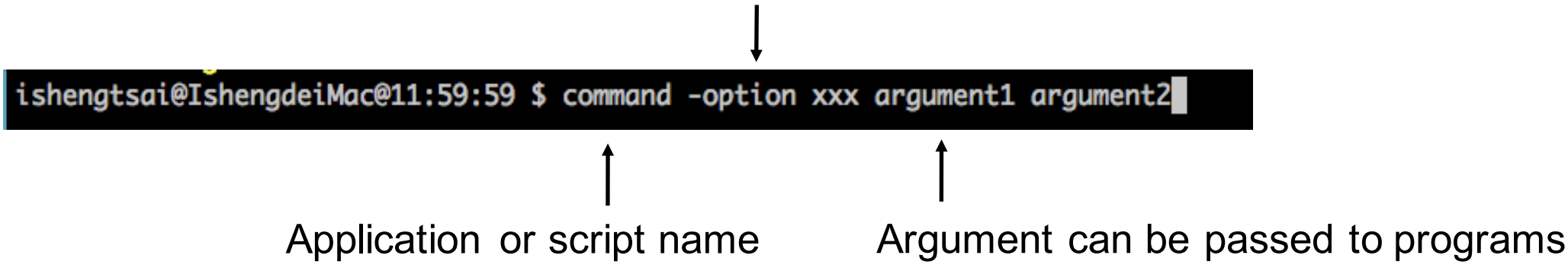
When you first start

```
IshengdeMacBook-Pro:~ ishengtsai$
```

```
/home/ishengtsai/
ishengtsai@t351@22:06:16 $
```

A typical command

Options always start with '-', and often expect to receive an option (xxx)



Special characters in bash

CHARACTER	MEANING
SPACE	Separate commands and arguments
# POUND	Comment
; SEMICOLON	Command separator two run multiple commands
. DOT	Source command OR filename component OR current directory
.. DOUBLE DOTS	Parent directory
' ' SINGLE QUOTES	Use expression between quotes literaly
, COMMA	Concatenate strings
\ BACKSLASH	Escape for single character
/ SLASH	Filename path separator
* ASTERISK	Wild card for filename expansion in globbing
>, <, >> CHARACTERS	Redirection input/outputs
| PIPE	Pipe outputs between commands

Special characters in bash

```
$ command xxxx yyyyy
```

Linux treats xxxx and yyyy as two arguments of the command

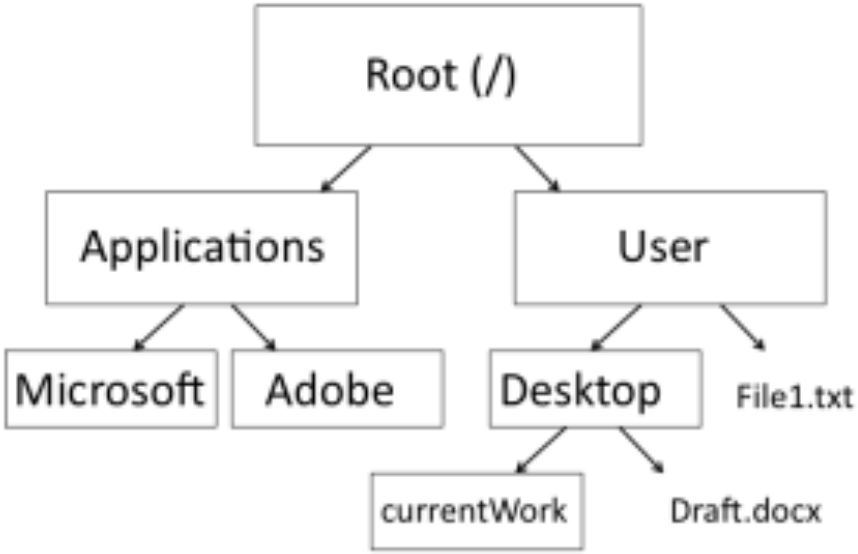
```
$ command ‘xxxx yyyyy’
$ command xxxx\ yyyy
```

You can uses single quotes or escape to distinguish special characters (in this case: space)

Short cut and emergency command in linux

SHORTCUT	MEANING
Tab	Autocomplete files or folder names
↑	Scroll up to the command history
↓	Scroll down to the command history
Ctrl + A	Go to the beginning of the line that you are typing
Ctrl + D	Go to the end of the line that you are typing
Ctrl + U	Clear all the line (or until the cursor position)
Ctrl + R	Search previously used commands
* Ctrl + C	Kill the process that you are running
Ctrl + D	Exit the current shell
Ctrl + Z	Put the running process to the background. Use command fg to recover it.

Directory structure



Try:
ls (list segment)
cd (change directory)
rm (abbreviation for remove)
mkdir (make directory)
pwd (print working directory)

Directory structure is like a tree

From /home/ishengtsai/

Relative path:
cd fungi # moves into **fungi** folder
  # now you are in /home/ishengtsai/fungi/
  # you can only do this successfully when you are in /home/ishengtsai/

cd .. # you go up one directory
  # now you are in /home/

Or absolute path:
cd /home/ishengtsai/fungi/ ;

Files commands **

COMMAND	USE	EXAMPLE
less	Open a file with less. Q to exit. Arrows to scroll	less myfile
touch	Create an empty file	touch myfile
mv	Move file between dirs. Change name	mv myfile yourfile
rm	Remove file	rm youfil
cat	Print file content as STDOUT	cat myfile
head	Print first 10 lines as STDOUT	head myfile
tail	Print last 10 lines as STDOUT	tail myfile
grep	Print matching lines as STDOUT	grep 'ATG' myfile
cut	Cut columns and print as STDOUT	cut -f1 myfile
sort	Sort lines and print as STDOUT	sort myfile
sed	Replace ocurrences, print lines STDOUT	sed 's/ATG/CTG/' myfile
wc	Word count	wc myfile

awk https://en.wikipedia.org/wiki/AWK

Compression commands

COMMAND	USE	EXAMPLE
gzip	Compress a file using gzip	gzip -c test.txt > test.txt.gz
gunzip	Uncompress a file using gzip	gunzip test.txt.gz
bzip2	Compress a file using bzip	bzip2 -c test.txt > test.txt.bz2
bunzip2	Uncompress a file using gzip	bunzip2 test.txt.bz2
tar	Archive files usint tar	tar -cf sample.tar sample/*.txt
tar -zcvf	Archive using tar and compress using gzip	tar -zcvf samples.tar.gz sample/*.txt
tar -zxvf	Unarchive using tar and uncompress using gunzip	tar -zxvf samples.tar.gz
tar -jcvf	Archive using tar and compress using bzip2	tar -jcvf samples.tar.bz2 sample/*.txt
tar -jxvf	Unarchive using tar and uncompress using bunzip2	tar -jxvf samples.tar.bz2

Redirection of input / output

The result of the **ls** command will be output and saved into **out.txt**

```
$ ls > out.txt
```

The result of the **ls** command will be output and **append** into **out.txt**
If the file **out.txt** already exists, then the original content will not be **replaced**, and
the new information will be added into the file

```
$ ls >> out.txt
```

Pipeline

… a **pipeline** is a set of **processes** chained by their **standard streams**, so that the output of each process (stdout) feeds directly as input (stdin) to the next one.

program1 | program2 | program3

Special character to **pipe** the results

Example:

ls -l | grep key | less

Demonstration I: daily tasks

1. Login into a terminal
2. Go to a specific directory that contains your data
3. Inspect your **fasta** files
   $ less ref.fa | grep '>' | less
   $ less ref.fa | grep '>' | wc -l

4. How about **fastq** file?
   - how many sequences?

5. How about gff file?
   - how many exons? How many genes?
   - how many genes that are expressed in the forward strand?

6. Check if command is successful

Installation

1. You need a bioinformatics program
    1. Download binaries and it should be ready to execute
    2. Or you have to compile
    3. Most modern program now deposit their program in **github**

```
cd /home/ijt/NGScourse/
git clone https://github.com/relipmoc/skewer.git
cd skewer
make
/home/ijt/NGScourse/skewer/skewer
```

compile

Ready to run!

Jiang *et al. BMC Bioinformatics* 2014, **15**:182
http://www.biomedcentral.com/1471-2105/15/182

METHODOLOGY ARTICLE Open Access

Skewer: a fast and accurate adapter trimmer for next-generation sequencing paired-end reads

Hongshan Jiang[1*], Rong Lei[1], Shou-Wei Ding[2] and Shuifang Zhu[1]

Demonstration II: daily tasks

1. Downloaded some sequenced data ; mapped to genome and you want to start looking at it.

2. Look at sam file
   $ samtools view xxx.bam | less

3. Okay, how about if I want to check the insert size of properly mapped reads? What filter to use? (https://broadinstitute.github.io/picard/explain-flags.html)

4. You have a file that you want to visualize, what next?

Keep a track of your science

We use evernote


[B303S1] Mapping and SNP calling from assembly of your choice [v1] — Evernote Plus

0. B303 lab book 按一下以新增標籤

已建立: 2015年4月8日 已更新: 2016年2月6日

您正在瀏覽與 7 人 共用的記事

[B303S1] Mapping and SNP calling from assembly of your choice [v1]

#You need a fasta file of reference genome
 # Looks like this...

```
>PNOK.scaff0001
AGTATGGTAATCTTCAGGCTCATCCACCATCTCTGTGATCTTCATGACTACTTTTGGGTTAACCTTCTCTCTAAGAATAGAGATAAGATATTCATCATGG
TCAGCTTTCTCATGAGGATCATCAAAAGTCTCTTTAAGCCTGGCTAGAAAAGTTTCCCACTCATCAGTGATAACCATGTCACACTCCTCCAAGTAGTTATC
TCAAGGATTTCAGAAAAGGTTTGAAGTCCTTTCTTTGTCCTGAGAATGGGTTGGGTTTAGCTACATGGATCTGTCTATGCAGTCTTCTTTTTCCCTTCAAT
TGAGTGAAGGTACTACTAGTCTAGAAAAAGTAAAAGGTAAAAAATAAAGGGAATAAGTGAATATCTACTCTAGGCTATCTAGTGTCTAATTCTAAGAGAAC
TTTGACTATTATTCACTAAGTAACAGGATTTAAACTAGTCTCCAGGCACAAGGCCCTTAAAATGAGAATCTAGCCAAGGGATAGTGTGTTATGTATCTAAC
TTACTCTTATAGGAATAGTGGTGAGAAAGTGGTAGAAGGCTTTAAAGGGTATGTCTAATAGAAAGTGAACAAGGGTGGGGAAAGATGATTGGTGGAAACAC
AGAATGCTTAGTCAGCAGGGTCTGAAGACAAGGCACAACCAGCATACACTGATATAGACAGAGACAGAGGAGAGGAGACAAGAGGATGTATTAGAAGG
TCGGCAGGGACTATGGTTAAACTATTAGGCTAAACTAAGCTGTGCTAGATGAAACAAAGTAAATAGGCTAACTACAGGCTGAGCTAGGCTATGCTAGCTG
TGTTCTGGAAGGTCTCAGAATATTCTAGAACAAGTGAGTAAGTAGTAAAGATAAGAGAGTGATAAGATAAGAGAGGATGCTTAGCAAGACAGCTGAGTCA
ATCGACTTATGTAAGATGATTATGTAAGTGCTTAGTCATATGATATGATTATGTAAGCTGCTTAGTAAGCAAAAAGCGAAAATACACTTGGTTTCATAAC
GAAGATCACTAGCTATATAGCAAGTGCAAATACGTCTCTCAACACGGATGAGAAGAAGATTCGTTTCGTATGCTCATATATGAAAGAAAAAGCCGGAGTA
GAAATCGCTCGAAACGAGATAGCAACAATACAGCAAGGGAACACGCGAGCAGAAGAATTCTTCACCAAACTAGCCTCACTCAGGATCCGAGCCGAGTACA
TTGAAGGATGGAGAAATCGGGTTATTGCAGTCGAAGAAGCACTCTATGCACACAAACCCAAGTATGAGGTCAAAAACTCGTTTAAACCCTCTACGACCTC
```

 # You also need a pairs of fastq files
 # In most cases you copy into the server
 # If you have fastq files on server already, skip this step
 # sftp into the server first
sftp ijt@140.109.143.135

#Copy fastq files to server
get /home/ishengtsai/fungi/Phellinus/fastqs/BRC/*PEtrimQ10* /Users/ishengtsai/Documents/Phellinus/data/fastqs/

BWA mapping (version 0.7.12-r1039)

 # you need to index the genome first using bwa index
bwa index reference.fa -p genome

```
ijt@mgb1@16:52:44 $ bwa index PNOK.fa -p genome
[bwa_index] Pack FASTA... 0.82 sec
[bwa_index] Construct BWT for the packed sequence...
[BWTIncCreate] textLength=63496440, availableWord=16467668
[BWTIncConstructFromPacked] 10 iterations done. 27163448 characters processed.
[BWTIncConstructFromPacked] 20 iterations done. 50180408 characters processed.
[bwt_gen] Finished constructing BWT in 27 iterations.
[bwa_index] 34.30 seconds elapse.
[bwa_index] Update BWT... 0.56 sec
[bwa_index] Pack forward-only FASTA... 0.42 sec
[bwa_index] Construct SA from BWT and Occ... 16.84 sec
[main] Version: 0.7.12-r1039
[main] CMD: bwa index -p genome PNOK.fa
[main] Real time: 52.946 sec; CPU: 52.948 sec
```

 # Map using bwa mem
 # Need to add Readgroup ID (RG), Sample ID (SM) and Library (LB) tag
*** Illumina/454/IonTorrent paired-end reads longer than ~70bp:**
bwa mem -t 8 -R '@RG\tID:1\tLB:GE01\tSM:GE01\tPL:ILLUMINA' genome PE_1.fq.gz PE_2.fq.gz > aln-pe.sam

Screenshot to log results

Comment your code (what was the purpose)

All the command can be reused (copy and paste!)

- Store everything online
- Share with people, with the world

Keep a track of your science

- Ten simple rules series in PLOS Computational Biology

PERSPECTIVE

Ten Simple Rules for Creating a Good Data Management Plan

William K. Michener*

EDITORIAL

Ten Simple Rules for a Computational Biologist's Laboratory Notebook

Santiago Schnell[1,2,3]*

What you needed to know but not taught in this lecture

- File permissions
- Ssh
- Simple scripting

Useful website:

- http://linux.vbird.org/linux_basic/ (Chinese ; extremely useful) ****
- https://evomics.org/learning/unix-tutorial/
- http://www.ark-genomics.org/events-online-training-eu-training-course/introduction-linux
- http://linuxcommand.org/

R : basic usage

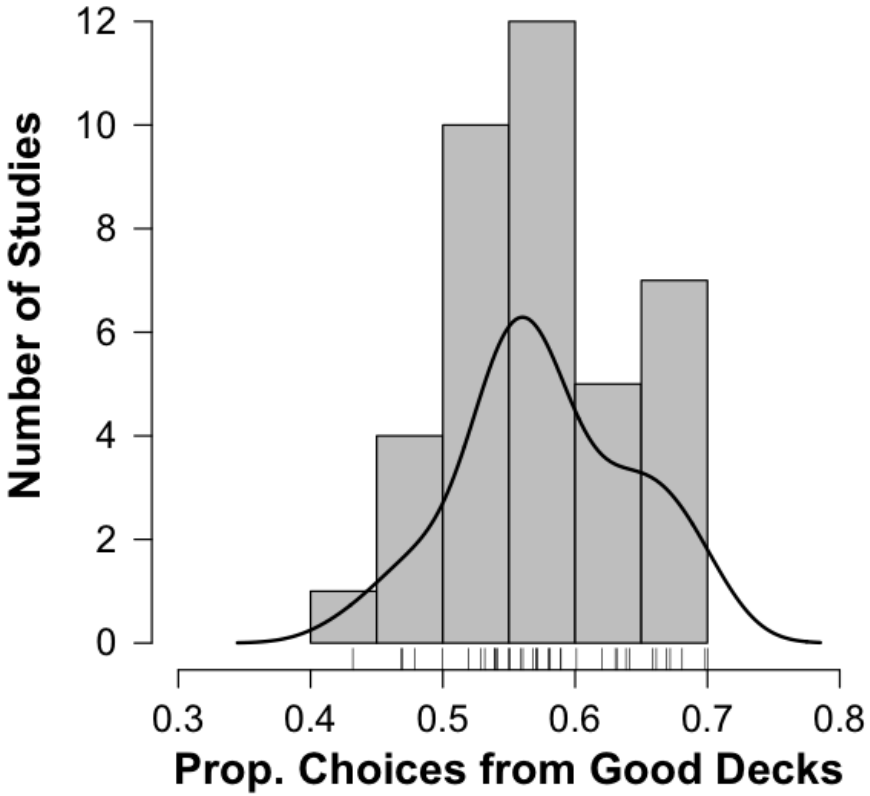
Isheng Jason Tsai

Introduction to NGS Data and Analysis
Lecture 2-2

R is a programming environment

- It's **free**
  - Hence R is supported by a large user network
  - R is open source
- Can be run on Windows, Linux and Mac
- Provides an unparalleled platform for programming new statistical methods in an easy and straightforward manner.
- **Excellent graphics capabilities**

Sharing and tutorials online

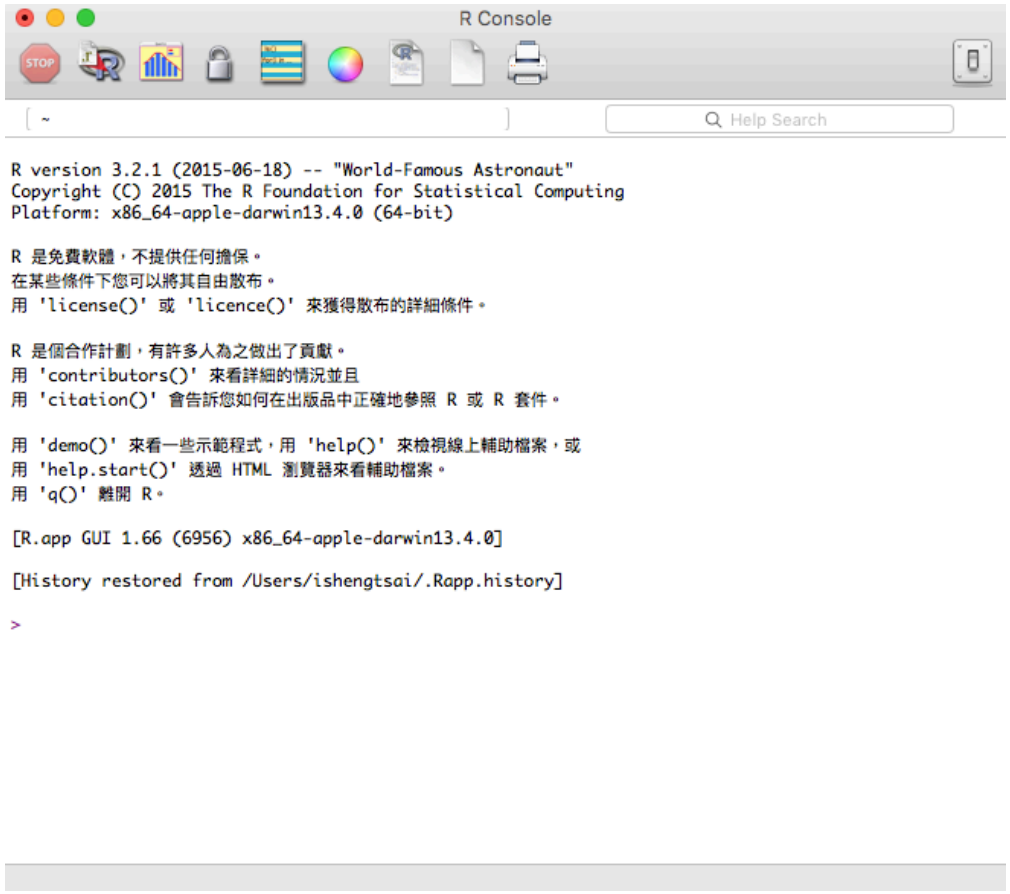



```
# rm(list = ls()) # Data: Proportion of choices from the
good decks as reported in 39 studies
good.choices <-
c(.43, .47, .47, .48, .50, .52, .53, .53, .54, .54, .54, .54, .55,
 .55, .55, .56, .56, .57, .57, .57, .57, .58, .58, .58, .59, .59, .
60, .62, .63, .63, .64, .64, .66, .66, .67, .67, .68, .70, .70)
par(cex.main = 1.5, mar = c(5, 6, 4, 5) + 0.1, mgp = c(3.5,
1, 0), cex.lab = 1.5 , font.lab = 2, cex.axis = 1.3, bty = "n",
las=1)
hist(good.choices, main = "", xlab = "", ylab = " ", ylim =
c(0, 13), xlim = c(.30, .80), axes = FALSE, col = "grey")
axis(1, seq(.30, .80, by = .1))
axis(2, seq(.00, 12, by = 2))
rug(jitter(good.choices))
mtext("Prop. Choices from Good Decks", side = 1, line =
2.5, cex = 1.5, font = 2)
mtext("Number of Studies", side = 2, line = 3, cex = 1.5,
font = 2, las = 0)
lines(density(good.choices), lwd = 2)
```

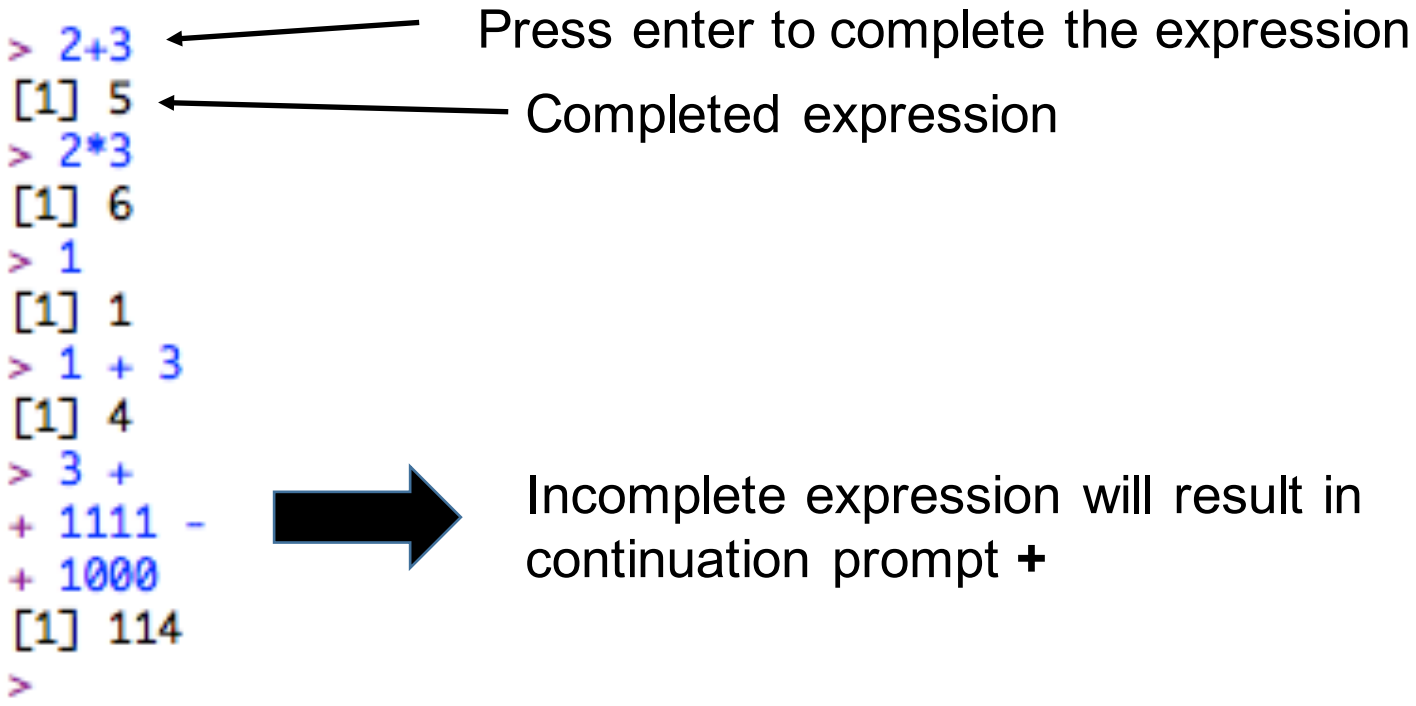
Download R

http://www.r-project.org

R can be executed anywhere



R as a calculator



Assignment

```
> x <- 5
> x
[1] 5
> y <- 10
> y
[1] 10
> x+y
[1] 15
> X <- 10
> X
[1] 10
> x
[1] 5
> x <- 100
> x
[1] 100
> z <- x + y + X
> z
[1] 120
```

<- is the assignment operation

R is case sensitive ; **x** does not equal to **X**

Original value is replaced

New value can be assigned as the result of calculation

Boolean assignment

```
student <- 30000
phd <- 56000
```

```
student > phd
```

```
[1] FALSE
```

```
student < phd
```

```
[1] TRUE
```

```
student != phd
```

```
[1] FALSE
```

```
student + student > phd
```

#Two heads are better than one

```
[1] TRUE
```

Vector is the simplest data structure in R

```
x<- c(1,2,3,4,5,6,7,8,9,10)
```

c = combine
In this case, we assign a **vector** of 10 numbers into x

```
x * 2
x /10 + 1
```

Selection

```
x<- c(1,2,3,4,5,6,7,8,9,10)
names(x)<-c("A","B","C","D","E","F","G","H","I","J")
```

```
x[x>5]
x[1:3]
x[1]
x[-1]
x[c("C","D")]
x[c("Z")]
x[x %in% c(7,9)]
x[x %in% c(7,13)]
```

```
> x[c("C","D")]
C D
3 4
> x[c("Z")]
<NA>
  NA
```

```
> x[x %in% 5]
E
5
> x[x %in% 10]
 J
10
> x[x %in% c(7,9)]
G I
7 9
> x[x %in% c(7,13)]
G
7
```

```
> x[x>5]
 F  G  H  I  J
 6  7  8  9 10
> x[1:3]
A B C
1 2 3
> x[1]
A
1
> x[-1]
 B  C  D  E  F  G  H  I  J
 2  3  4  5  6  7  8  9 10
```

Different types of vectors

```
x<- c(1,2,3,4,5,6,7,8,9,10)
strings <- c("AS","BRC")

typeof(x)
typeof(strings)
```

```
> typeof(x)
[1] "double"
> typeof(char)
[1] "character"
> typeof(strings)
[1] "character"
```

This matters when one data type is numbers, and you want to sort them categorically

Function

function (arg1, arg2, arg3… , option1=,option2=...)

```
x<- c(1,2,3,4,5,6,7,8,9,10)
y<- c(3,6,9,10,13,30,20,100)
```

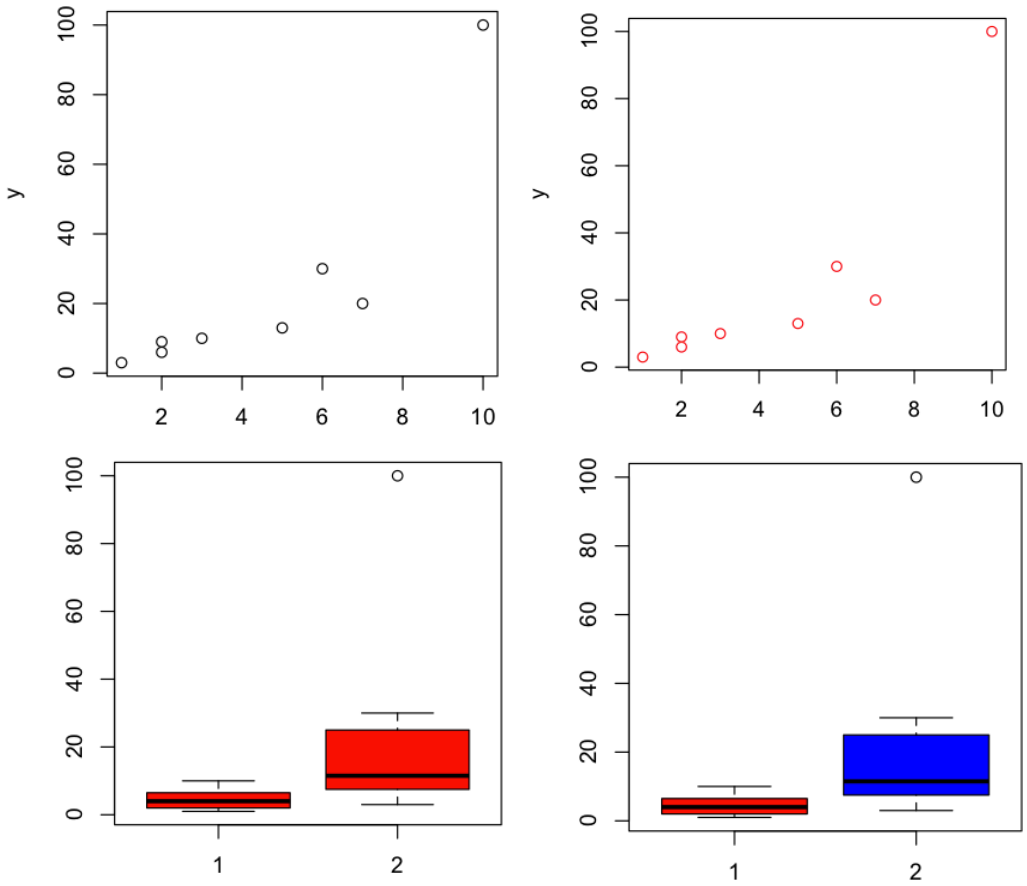
```
mean(x)
mean(y)
median(x)
max(x)
```

```
> x<- c(1,2,2,3,5,6,7,10)
> y<- c(3,6,9,10,13,30,20,100)
> mean(x)
[1] 4.5
> mean(y)
[1] 23.875
> median(x)
[1] 4
> median(y)
[1] 11.5
> max(x)
[1] 10
> min(y)
[1] 3
```

- Must have **assigned names**
- Applies using **round brackets**
- Takes **argument** and options

R simple plot I

```
x<- c(1,2,3,4,5,6,7,8,9,10)
y<- c(3,6,9,10,13,30,20,100,220,100)

plot(x,y)
plot(x,y,col="red")

boxplot(x,y,col="red")
boxplot(x,y,col=c("hotpink", "yellow"))

boxplot(x,y,col=c("hotpink", "yellow"),main="Lec2")
```



R simple plot II

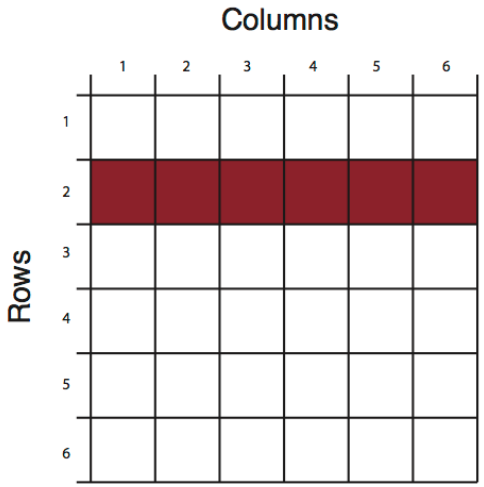
Follow examples here:
http://al2na.github.io/compgenr/intro_to_r/plotting_in_r.html

Matrices are a collection of vectors of the same type

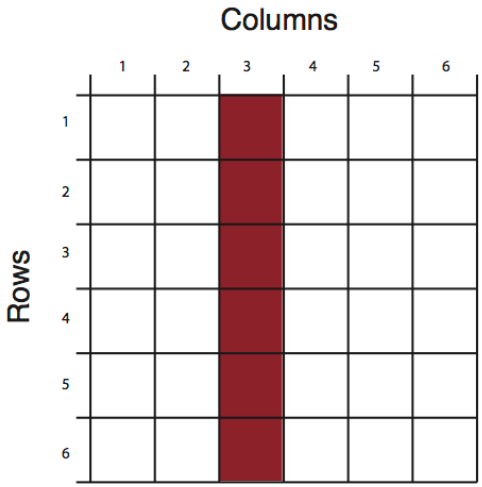
```
mat <- matrix(c(1, 3, 2, 5, -1, 2, 2, 3, 9), nrow = 3)
rownames(mat) <- c("a","b","c")
colnames(mat) <- c(”x",”y",”z")
```

```
  [,1] [,2] [,3]
a    1    5    2
b    3   -1    3
c    2    2    9
```

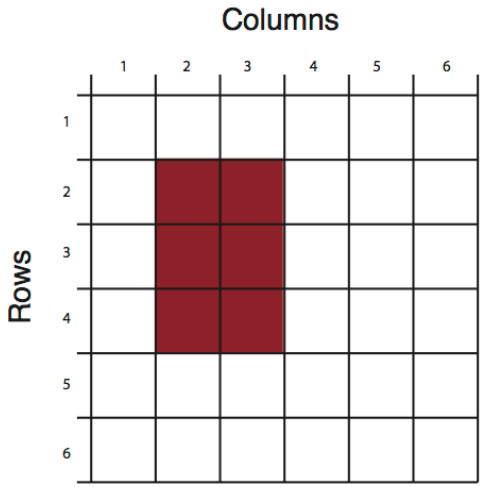
`mat[2,]`



`mat[,3]`



`mat[2:4,2:3]`



http://al2na.github.io/compgenr/intro_to_r/data_types.html

Matrices - summary

- Each row and column must have data of the **same** type (numeric, character etc)
- Most useful when do linear algebra (e.g. PCA,)

```
> mat * 2
     [,1] [,2] [,3]
[1,]    2   10    4
[2,]    6   -2    6
[3,]    4    4   18
```

- If you want **different** data types, need to use objects called data.frames

Data frames

- Think of these like Excel spreadsheets
- **All the values of the same variable must go in the same column**
  - E.g., age, sex, RPKM, numbers
- **Rows represent samples**
  - E.g., sample A collected in Taiwan, sample B collected in Japan

- Like matrices but different types of data are allowed

R has some pre-installed data frames

```
iris
head(iris)
```

Or you can read into data

```
worms <- read.table("worms.txt", header=T)
head(worms)
```

```
> worms
          Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
1        Nashs.Field  3.6    11  Grassland     4.1 FALSE             4
2     Silwood.Bottom  5.1     2     Arable     5.2 FALSE             7
3      Nursery.Field  2.8     3  Grassland     4.3 FALSE             2
4        Rush.Meadow  2.4     5     Meadow     4.9  TRUE             5
5    Gunness.Thicket  3.8     0      Scrub     4.2 FALSE             6
6           Oak.Mead  3.1     2  Grassland     3.9 FALSE             2
7       Church.Field  3.5     3  Grassland     4.2 FALSE             3
8            Ashurst  2.1     0     Arable     4.8 FALSE             4
9        The.Orchard  1.9     0    Orchard     5.7 FALSE             9
10     Rookery.Slope  1.5     4  Grassland     5.0  TRUE             7
11       Garden.Wood  2.9    10      Scrub     5.2 FALSE             8
12      North.Gravel  3.3     1  Grassland     4.1 FALSE             1
13      South.Gravel  3.7     2  Grassland     4.0 FALSE             2
14 Observatory.Ridge  1.8     6  Grassland     3.8 FALSE             0
15        Pond.Field  4.1     0     Meadow     5.0  TRUE             6
16      Water.Meadow  3.9     0     Meadow     4.9  TRUE             8
17         Cheapside  2.2     8      Scrub     4.7  TRUE             4
18        Pound.Hill  4.4     2     Arable     4.5 FALSE             5
19        Gravel.Pit  2.9     1  Grassland     3.5 FALSE             1
20         Farm.Wood  0.8    10      Scrub     5.1  TRUE             3
```

Selection in data frames

Square brackets

- **dat[i ,]** would select the *i*-th row (which is a **vector**)
- **dat[, j]** would select the *j*-th column (which is a **vector**)
- **dat[i, j]** would select the value from the *i*-th row and *j*-th column

```
worms[,1]
worms[1,]
worms[1,1]
```

dollar ($) operation (for columns only)

```
worms$Area
```

subset (not discussing today)

Some combinations of it

Square brackets

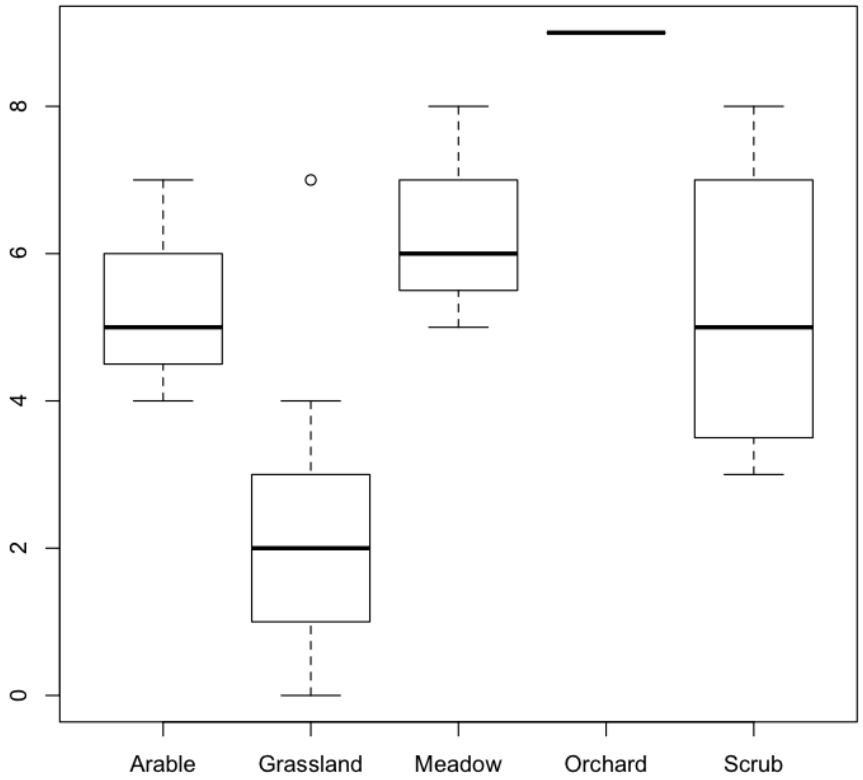
- **dat[i ,]** would select the *i*-th row (which is a **vector**)
- **dat[, j]** would select the *j*-th column (which is a **vector**)
- **dat[i, j]** would select the value from the *i*-th row and *j*-th column

```
worms[worms$Area < 3,]
worms[(worms$Area < 3) & (worms$Worm.density <4),]
worms[(worms$Area < 3) & (worms$Worm.density <4),]$Soil.pH
```

```
> worms
          Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
1        Nashs.Field  3.6    11  Grassland     4.1 FALSE            4
2     Silwood.Bottom  5.1     2     Arable     5.2 FALSE            7
3      Nursery.Field  2.8     3  Grassland     4.3 FALSE            2
4       Rush.Meadow  2.4     5     Meadow     4.9  TRUE            5
5    Gunness.Thicket  3.8     0      Scrub     4.2 FALSE            6
6           Oak.Mead  3.1     2  Grassland     3.9 FALSE            2
7       Church.Field  3.5     3  Grassland     4.2 FALSE            3
8            Ashurst  2.1     0     Arable     4.8 FALSE            4
9        The.Orchard  1.9     0    Orchard     5.7 FALSE            9
10     Rookery.Slope  1.5     4  Grassland     5.0  TRUE            7
11      Garden.Wood  2.9    10      Scrub     5.2 FALSE            8
12     North.Gravel  3.3     1  Grassland     4.1 FALSE            1
13     South.Gravel  3.7     2  Grassland     4.0 FALSE            2
14 Observatory.Ridge  1.8     6  Grassland     3.8 FALSE            0
15       Pond.Field  4.1     0     Meadow     5.0  TRUE            6
16     Water.Meadow  3.9     0     Meadow     4.9  TRUE            8
17        Cheapside  2.2     8      Scrub     4.7  TRUE            4
18       Pound.Hill  4.4     2     Arable     4.5 FALSE            5
19       Gravel.Pit  2.9     1  Grassland     3.5 FALSE            1
20        Farm.Wood  0.8    10      Scrub     5.1  TRUE            3
```

More plot from dataframes

```
plot(worms$Area,worms$Slope,col=as.numeric(worms$Vegetation))
plot(worms$Area,worms$Slope,col=as.numeric(worms$Vegetation),pch=as.numeric(worms$Vegetation))
boxplot(worms$Worm.density ~ worms$Vegetation)
```




```
> worms
           Field.Name Area Slope Vegetation Soil.pH  Damp Worm.density
1         Nashs.Field  3.6    11  Grassland     4.1 FALSE            4
2      Silwood.Bottom  5.1     2     Arable     5.2 FALSE            7
3       Nursery.Field  2.8     3  Grassland     4.3 FALSE            2
4        Rush.Meadow  2.4     5     Meadow     4.9  TRUE            5
5     Gunness.Thicket  3.8     0      Scrub     4.2 FALSE            6
6           Oak.Mead  3.1     2  Grassland     3.9 FALSE            2
7        Church.Field  3.5     3  Grassland     4.2 FALSE            3
8             Ashurst  2.1     0     Arable     4.8 FALSE            4
9         The.Orchard  1.9     0    Orchard     5.7 FALSE            9
10      Rookery.Slope  1.5     4  Grassland     5.0  TRUE            7
11       Garden.Wood  2.9    10      Scrub     5.2 FALSE            8
12      North.Gravel  3.3     1  Grassland     4.1 FALSE            1
13      South.Gravel  3.7     2  Grassland     4.0 FALSE            2
14 Observatory.Ridge  1.8     6  Grassland     3.8 FALSE            0
15        Pond.Field  4.1     0     Meadow     5.0  TRUE            6
16      Water.Meadow  3.9     0     Meadow     4.9  TRUE            8
17         Cheapside  2.2     8      Scrub     4.7  TRUE            4
18        Pound.Hill  4.4     2     Arable     4.5 FALSE            5
19        Gravel.Pit  2.9     1  Grassland     3.5 FALSE            1
20         Farm.Wood  0.8    10      Scrub     5.1  TRUE            3
```

More useful functions here

```
y<-abs(-20)
x<-Sum(y+5)
Z<-Log(x)
round(x,1)
summary(worms)
head(worms)
tail(worms)
ncol(worms)
nrow(worms)
```

Statistics

```
# Simulate two normal distributions one at mean =4, and another at  6
x <- rnorm(500,4)                    # mean at 4
y <- rnorm(500,6)                    # mean at 6

# Plot histogram
plot(hist(x), col=rgb(0,0,1,1/4), xlim=c(0,10))
plot(hist(y), col=rgb(1,0,0,1/4), xlim=c(0,10), add=T)
t.test(x,y)
```

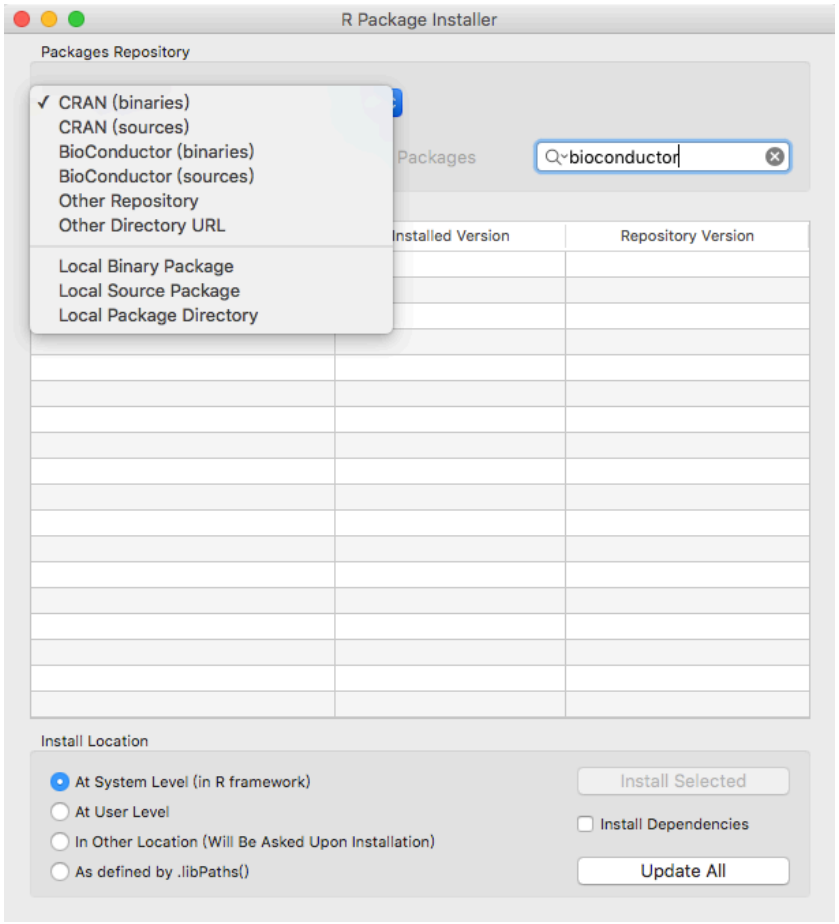
```
# Simulate two normal distributions at mean =3

x <- rnorm(500,3)
y <- rnorm(500,3)
t.test(x,y)
```

Running out of functions to use?

Use Packages

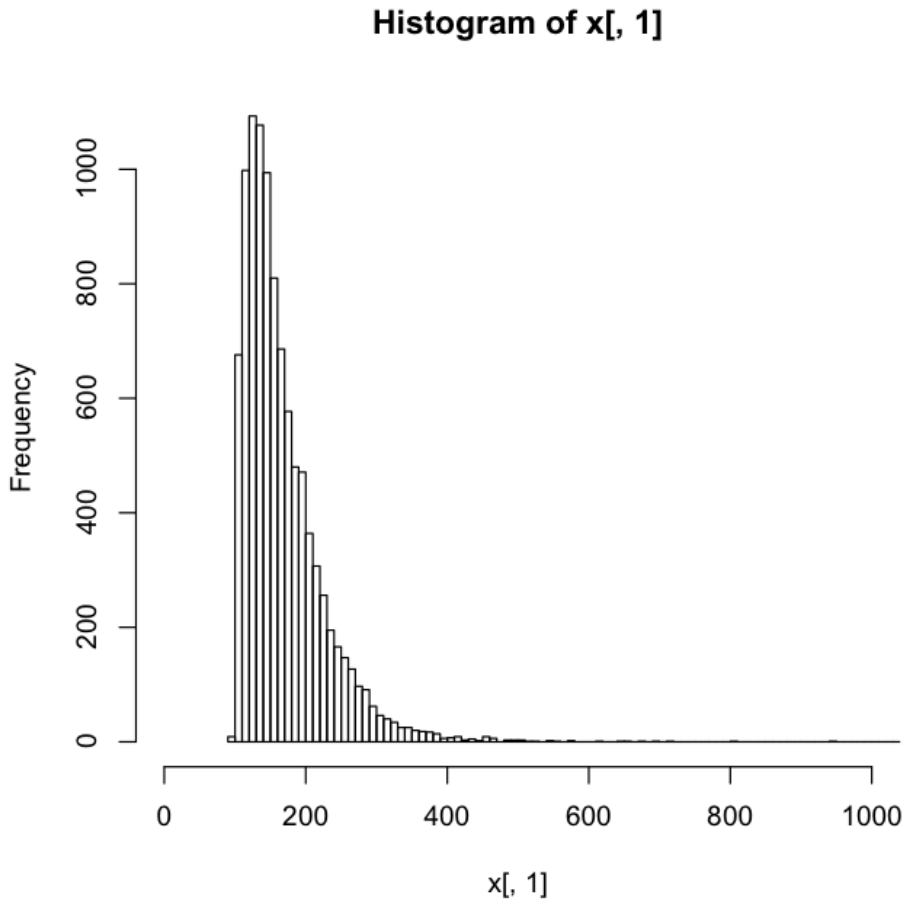
- R consists of a **core** and **additional packages**.
- Collections of R functions, data, and compiled code
- Well-defined format that ensures easy installation, a basic standard of documentation, and enhances portability and reliability

Install R packages



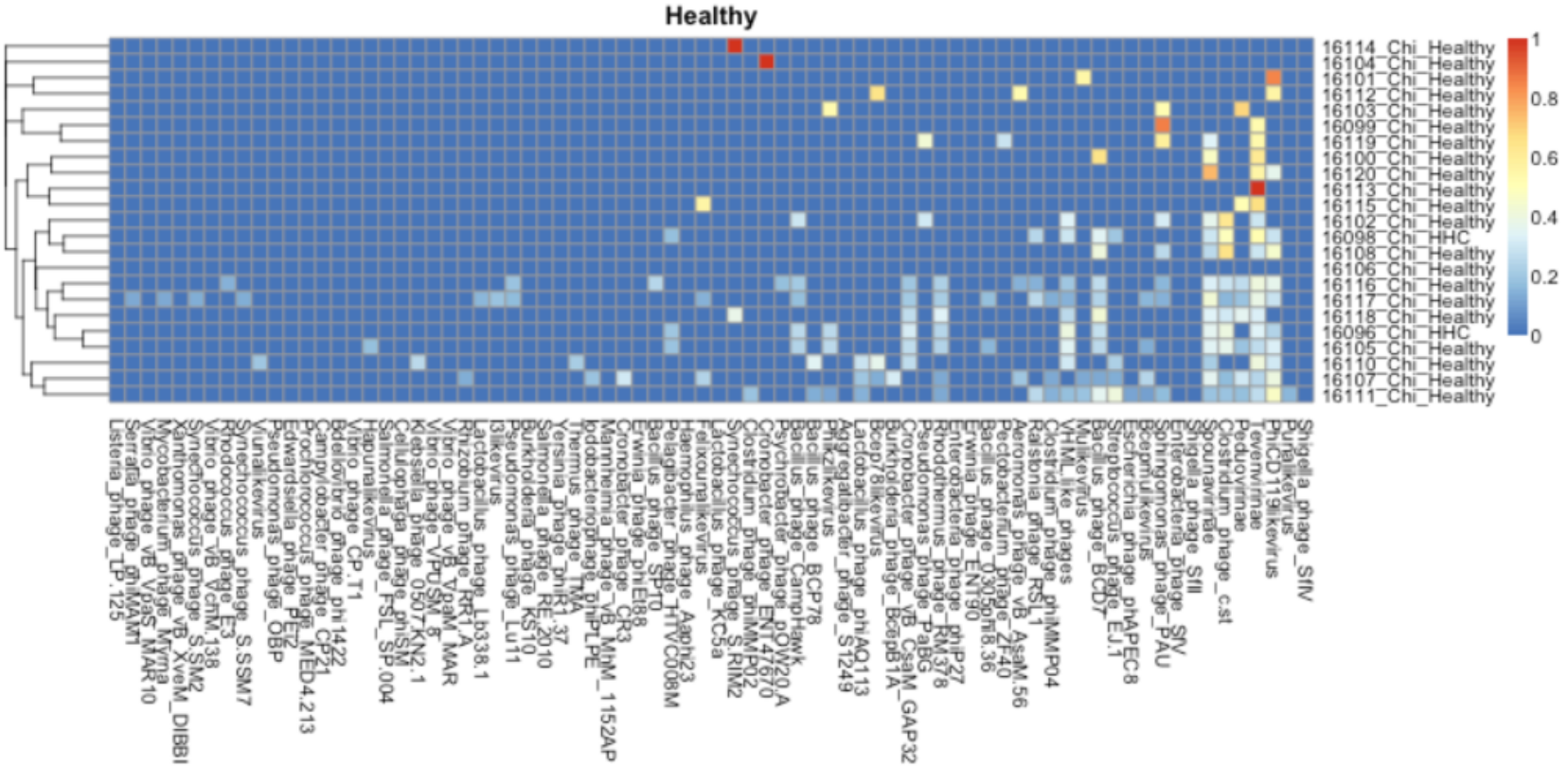
Also can install in command line

Example I




```
x<- read.table("/Users/ishengtsai/Dropbox/NGSLectureNotes/insert.txt")
length(x[,1])
hist(x[,1],xlim=c(0,1000),breaks=50000)
```

Example II




```
library("pheatmap")
library("vegan")
healthy <- read.table("/Users/ishengtsai/Dropbox/NGSLectureNotes/myoviridae_healthy.txt")
healthy_hellinger <- decostand(healthy, method="hellinger")
pheatmap(healthy_hellinger, cluster_cols=FALSE, cellwidth=8, cellheight=8, main="Healthy")
```

Example III

```
x <-
read.table("/Users/ishengtsai/Dropbox/NGSLecture
Notes/BRC.txt",header=T,row.names = 1)

# Exploration
head(x)
plot(x$Postdoc_RA,x$Student,col=x$Level)

# More pretty plot
require(reshape2)
df.m <- melt(x, id.var = "Level")
require(ggplot2)
ggplot(data = df.m, aes(x=variable, y=value)) +
geom_boxplot(aes(fill=Level))


#heatmap
x_hellinger <- decostand(x[,1:3], method="hellinger")
pheatmap(x_hellinger, cluster_cols=FALSE,
cellwidth=8, cellheight=8, main="BRC")
```

中央研究院生物多樣性研究中心人員名單及電話表

總機：2789-9621 Fax：2789-9624
傳真：2789-9624(跨領域)、2787-2235(新溫室)
警衛：2787-3299(跨領域)、2787-2236(新溫室)

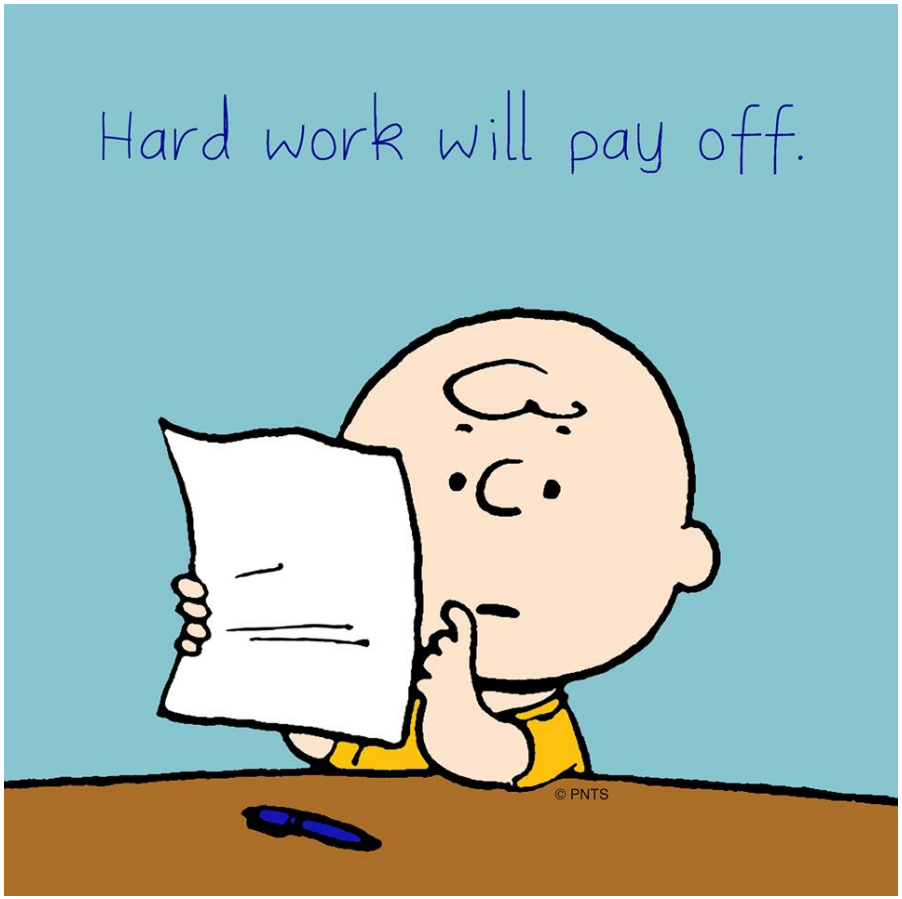
2016.02.02 更新

主任	電話	行政人員	電話	行政人員	電話	行政人員	電話	行政人員	電話
李文雄	2787-2256	葉欣宜	2787-2201	周麟丞	2787-2205	林妙穗	2787-2209	顏梅花	2787-2215
		劉建華	2787-2202	黃光隆	2787-2206	馮欣蓉	2787-2210	蕭彰志	2787-2216
		陳淑敏	2787-2203	洪小鳳	2787-2207	陳怡樺	2787-2212	吳莉珍	2787-2234
		徐曼慈	2787-2204	許文良	2787-2208	張玉玲	2787-2213	陳捷教	2789-9623

Lab位置	主持人	分機/傳真	博士後研究及助理	研究生	其他人員	助理分機
跨領域 A202	趙淑妙	2787-1155	吳宗賢	許智堯·徐銘鴻·陳意娜·王婷淇		2787-1021
跨領域 A203	李文雄	2787-2256 2787-2261(Fax)	吳俊揚·彭小平	蔡哲文·杰羅姆	林思安·周書仔	2787-2254
			陳湘君·黃禹馨·吳賽華·吳筱曼	李明璇·劉意霖		2787-2257
			林浩業·程凱若·林玉儒·高慧瑩	安馬瑞·皮宏偉	張瑞仁	2787-2259 2787-2274
			黃貞祥·劉文裕·班佐伊·黃啟發	林進之·黃志仁	張家銘	2787-2262
			王瑋騰·蘇建豪·張耀明·游竣評	陳志冠·古麥斯		2787-2263
跨領域 A204	王子元	2787-2258			陳素蘩	2787-2268
跨領域 B210	編輯室	2789-9529	吳佳祈			
跨領域 C206	沈聖峰	2787-2280	劉彥廷·吳士瑋·詹仕凡·詹偉平·袁子能·林彥成·林靖淳·張勝心	陳伯飛·劉育菁·汪瑞瑋·龍悅生·施怡伸·張昌祐·蔡祥瑜		2787-2281 2787-2282 2787-2283(B203)

In summary

- Start practicing
- There are so much data out there
- Going through tutorials
- **Learn through real case scenarios**



Useful websites

Useful website:

- http://shinyapps.org/apps/RGraphCompendium/index.php
- https://plot.ly/r/
- http://www.statmethods.net/
- Always search for slideshare

Chinese lecture websites for reference

- http://web.ntpu.edu.tw/~cflin/