

# 550.400: Mathematical Modeling and Consulting

## Lecture Notes

**Instructor:**  
Dr. N. H. Lee

JHU AMS 2012 FALL

Last Compiled on September 24, 2012

Notes

---

---

---

---

---

---

---

---

Notes

---

---

---

---

---

---

---

---

Notes

---

---

---

---

---

---

---

---

Notes

---

---

---

---

---

---

---

---

September 24, 2012's Lecture

○○○  
○○○○○  
○○  
○○

## Outline

### September 24, 2012's Lecture

Vim  
Git  
L<sup>A</sup>T<sub>E</sub>X  
Causality & Spurious Correlation  
Math Model Building

2 / 26

September 24, 2012's Lecture

●○○  
○○○○○  
○○  
○○

## Vim

Vim is a *highly customizable* text editor

1. L<sup>A</sup>T<sub>E</sub>X, R, C/C++, Java, Python, Git and etc.
2. Regular expression, syntax coloring, autocompletion
3. <ESC>-mode
  - :-mode, aka., the last line mode
  - i-mode, aka., the insert mode

3 / 26

September 24, 2012's Lecture

○○●  
○○○○○  
○○  
○○

## Vim

- Download & Install GVim or MacVim
- Download & Install tetris.vim
- Download & Install Gundo

4 / 26

## Tutorial: Vim

vi/vim lesson 1 - basic editing

Esc

normal mode

\$

end

^

"soft" buf

O

"hard" buf

W

next word

E

end word

R

replace mode

U

undo

I

insert mode

A

append at end

h

←

j

↓

k

↑

l

→

X

back space

x

delete char

B

prev WORD

b

next WORD

Basics:

**[h][j][k][l]** are vi/vim cursor keys - use them as they are - much closer than regular cursor keys!  
Use **[i]** to enter insert mode, cursor turns from a block into a vertical line, and you can type in text. Use **[Esc]** to return to normal mode.  
Use **[x]** to delete the current character, or **[X]** to delete the one to the left.  
Use **[A]** to go insert text at the end of the line (wherever you are in the line!)  
(Note: insert mode is actually very similar to a regular editor; you can use cursor/navigation keys, backspace, delete...)

Extras:

**[u]** to undo the last action - traditional vi has a single level, while vim supports unlimited undo (CTRL-**[** to redo)  
**[0]** jumps directly to the beginning of the line, **[^]** to the end, and **[\_]** to the first non-blank  
Use **[w][b]** to move along 'words'. A 'word' is a sequence of all alphanumeric or punctuation signs: **[w]** **[b]**  
Use **[W][B]** to move along WORDs. A 'WORD' is a sequence of any non-blank characters: **[W]** **[B]**  
Use **[i]** to enter insert mode with an overstrike cursor, which types over existing characters.  
**[w]** and press enter to save, **[q]** and enter to quit.

moves the cursor, or defines the range for an operation

direct action command, if vi, it enters insert mode

version 1.1

April 26, 08

5 / 26

Notes

## Intro. to Using Git for Off-Line Teamwork I

Places to set up a git for your group work:

- Git Hub
- Dropbox

Why does it matter?

- It allows you to collaborate with others off-line
- You leave a trail of your contributions to the project

In-Class Activities for setting up a github account

- go to github.com
- initiate a git project from github
- set up your local folder
- populate the folder with new contents

Notes

## Intro. to Using Git for Off-Line Teamwork II

as shown in Figure 9-1.

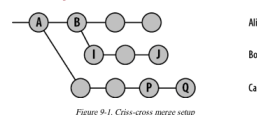


Figure 9-1. Cross-cross merge setup

Imagine that Cal started the project and Alice joined in. Alice worked on it for a while, then Bob joined in. In the meantime, Cal has been working away on his own version.

Eventually, Alice merged in Bob's changes, and Bob kept on working without merging Alice's changes back into his tree. There are now three different branch histories (Figure 9-2).

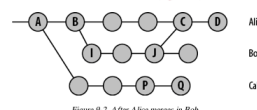


Figure 9-2. After Alice merges in Bob

Let's imagine that Bob wants to get Cal's latest changes. The diagram is looking pretty complicated now, but this part is still relatively easy. Trace up the tree from Bob, through Alice, until

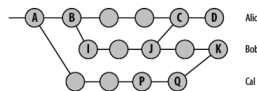


Figure 9-3. After Bob merges in Cal

### TIP

You can always find the merge base between two or more branches by using `git merge-base`. It is possible for there to be more than one equally valid merge base for a set of branches.

So far, so good.

Alice now decides that she, too, wants to get Cal's latest changes, but she doesn't realize Bob has already merged Cal's tree into his. So she just merges Cal's tree into hers. That's another easy operation because it's obvious where the diverged from Cal. The resulting history is shown in Figure 9-4.

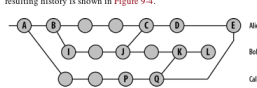


Figure 9-4. After Alice merges in Cal

```

1 git branch alice
2 git checkout alice
3 git merge --no-commit --no-ff bob

```

## Focus Problem

- Objective: your own copy of the poem
- Rule 1: You write one stanza of the poem into the `main.tex` file
- Rule 2: You can collect all the others only by using the following commands:

```

1 git remote
2 git pull
3 git push
4 git fetch
5 git merge

```

Notes

## Adv. Git Mehod for Off-Line Teamwork I

```
1 git format-patch master~2..master
```

Notes

---

---

---

---

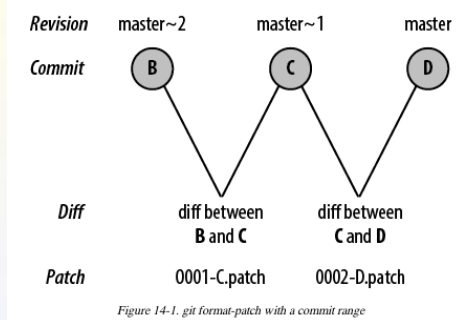
---

---

---

---

## Adv. Git Mehod for Off-Line Teamwork II



Notes

---

---

---

---

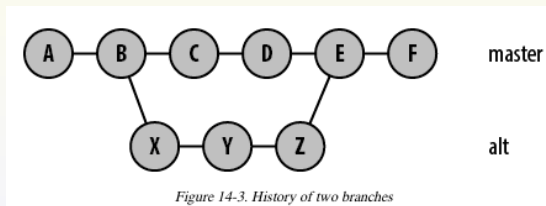
---

---

---

---

## Git Exercise I



Notes

---

---

---

---

---

---

---

---

## Git Exercise II

Create a git folder with the following history

- Each node's label signifies the commit
- The folder contains only one single file `main.txt` throughout the history
- KISS (See WMA for its meaning)

### Class Exercise

Collect all 8 stanzas together with your neighbor.

- You do four of them
- Your teammate do four of them
- Then, you combine yours with your teammate's

Notes

---

---

---

---

---

---

---

---

## Using R to do System Admin Stuff I

```
1 for(itr in 1:8) {
2   stanzaname = paste("stanza",itr,sep="")
3   gitaddress = paste("git://github.com/nhlee/550400.",
4                     stanzaname, ".git", sep="")
5   bashcommand = paste("git remote add ",
6                       stanzaname, " ", gitaddress, sep="")
7   system(bashcommand)
8 }
```

- 1:8 creates a vector that ...
- X = 1 assigns 1 to X
- X <- 1 also assigns 1 to X
- lots of things are done through function
- paste and system are functions that ...

13 / 26

Notes

---

---

---

---

---

---

---

---

## Using R to do System Admin Stuff II

- functions has none or more arguments
- arguments are implicitly ordered but the order can be overridden

```
1 system('ls -ld .*')
2 system('cat .Rprofile')
3 system('cat .bashrc')
4 system('cat .gitignore')
5 system('cat .vimrc')
```

- .xxx files are hidden
- ls -ld .\* show the hidden files
- .Rprofile set up your R behavior
- .bashrc set up your bash behavior
- .gitignore set up your git behavior

14 / 26

Notes

---

---

---

---

---

---

---

---

## Using R to do System Admin Stuff III

- .vimrc set up you vim behavior
- these files are equivalent to Preference part of your GUI software

15 / 26

Notes

---

---

---

---

---

---

---

---

## Intro. to workstatement template I

```
1 \documentclass[12pt,letterpaper][article]
2 \usepackage{amsmath,amsthm,amssymb,amsfonts} # for popular math add-on
3 \usepackage{graphicx} # for inserting png, jpeg, pdf files as figure
4 \usepackage{bm} # for bold math
5 # some preamble stuff omitted (see the actual template)
6 \begin{document}
7 \section{A}
8   \subsection{a}
9   \paragraph{Hello World}
10  \begin{align*}
11    & f(x) = \int_0^1 \sin(u+x) du, \quad \backslash
12    & f(\bm{x}) = \int_0^1 \sin(u+\|\bm{x}\|) du.
13  \end{align*}
14 \end{document}
```

16 / 26

Notes

---

---

---

---

---

---

---

---

# Introduction to beamer I

## Basic Body Layer

```
1 \begin{document}
2 \section{Hello World}
3   \subsection{hello world}
4     \begin{frame}
5       \frametitle{hi world}
6       \begin{columns}
7         \begin{column}{0.5\textwidth}
8           \begin{itemize}
9             \item Alice!
10          \end{itemize}
11        \end{column}
12        \begin{column}{0.5\textwidth}
13          \begin{block}{hey world}
14            Bob!
15          \end{block}
16        \end{column}
17      \end{columns}
18 \end{frame}
```

Notes

---

---

---

---

---

---

---

---

# Introduction to beamer II

```
19 \end{document}
```

## Basic Preambles

```
1 \documentclass[hyperref={colorlinks=false},handout,10pt]{beamer}
2 \usetheme{Singapore}
3 \usecolortheme{lily}
4 \usefonttheme[onlymath]{serif} % What does this do?
```

## OR

```
1 \documentclass[hyperref={colorlinks=false},handout,10pt]{beamer}
2 \usetheme{Berlin}
3 \usecolortheme{wolverine}
4 \usefonttheme[onlymath]{serif} % What does this do?
```

For a more complete array of themes, go to:

► <http://www.hartwork.org/beamer-theme-matrix/>

Notes

---

---

---

---

---

---

---

---

# Introduction to beamer III

SO, how to put a code in the slide? and it looks like codes?

```
\begin{lstlisting}
require(tikzDevice)
x = rnorm(100)
plot.ts(x)
dev.off()
\end{lstlisting}

1 require(tikzDevice)
2 x = rnorm(100)
3 plot.ts(x)
4 dev.off()
```

But, this requires the following in the preamble portion of your tex file:

Notes

---

---

---

---

---

---

---

---

# Introduction to beamer IV

```
\usepackage{listings}
\lstset{
  basicstyle=\footnotesize\ttfamily,
  numbers=left,
  frame=bottomline,
  frametopmargin=50pt,
}
```

Where to get more help:

► <http://en.wikibooks.org/wiki/LaTeX/Presentations>

Notes

---

---

---

---

---

---

---

---

## Spurious Causality I

```
1 CBE <- read.table('http://www.massey.ac.nz/~pscowper/ts/cbe.dat')
2 Ets <- ts(CBE[,3], start = 1958, freq=12)
3 Cts <- ts(CBE[,2], start = 1958, freq=12)
4 plot(as.vector(aggregate(Cts)),as.vector(aggregate(Ets)))
```

```
1 set.seed(10)
2 x <- rnorm(100)
3 y <- rnorm(100)
4 for(i in 2:100) {
5   x[i] <- x[i-1] + rnorm(1)
6   y[i] <- y[i-1] + rnorm(1)
7 }
8 plot(x,y)
```

21 / 26

Notes

---

---

---

---

---

---

---

---

## Spurious Causality II

```
1 xrates <- read.table(
2   'http://www.massey.ac.nz/~pscowper/ts/us_rates.dat')
3 plot(xrates$UK,xrates$EU,pch=4)
4 require(tseries)
5 pp.test(xrates$UK)
6 pp.test(xrates$EU)
```

```
1 x <- y <- mu <- rep(0,1000)
2 for(i in 2:1000) mu[i] <- mu[i-1] + rnorm(1)
3 x <- mu + rnorm(1000)
4 y <- mu + rnorm(1000)
5 adf.test(x)$p.value
6 adf.test(y)$p.value
7 po.test(cbind(x,y))
8
9 po.test(cbind(xrates$UK,xrates$EU))
10 ukeu.lm <- lm(xrates$UK ~ xrates$EU)
11 ukeu.res <- resid(ukeu.lm)
12 ukeu.res.ar <- ar(ukeu.res)
```

22 / 26

Notes

---

---

---

---

---

---

---

---

## Spurious Causality III

```
13 ukeu.res.ar$order
```

23 / 26

Notes

---

---

---

---

---

---

---

---

## How to do software documentation using R

```
1 myfun <- function(x) {x^2}
2 package.skeleton(name='MYPAC',
3   list='myfun',
4   path='/')
5 #Do the documentation
6 system('R CMD check ~/MYPAC')
7 system('R CMD build ~/MYPAC')
8 system('R CMD install MYPAC')
```

24 / 26

Notes

---

---

---

---

---

---

---

---

## A brain-teaser

"To encourage Elmer's promising tennis career, his father offers him a prize if he wins (at least) two tennis sets in a row in a three-set series to be played with his father"

- What is that you wish to know?
- unimportant, exogenous, and endogenous?
- if the model fits the situation, will we be able to use it?
- Test the model

Notes

---

---

---

---

---

---

---

---

## Arguments from Scale I

Cost of Packing

Speed of Racing Shells

Size Effect in Animal

Notes

---

---

---

---

---

---

---

---

Notes

---

---

---

---

---

---

---

---

Notes

---

---

---

---

---

---

---

---