550.400: Mathematical Modeling and Consulting

Lecture Notes

Instructor:
Dr. N. H. Lee

JHU AMS 2012 FALL

Last Compiled on September 24, 2012

Notes

Outline

September 24, 2012's Lecture
- Vim
- Git
- LaTeX
- Causality & Spurious Correlation
- Math Model Building

Notes

Vim

Vim is a *highly customizable* text editor

1. LaTeX, R, C/C++, Java, Python, Git and etc.
2. Regular expression, syntax coloring, auto-completion
3. <ESC>-mode
   - :-mode, aka., the last line mode
   - i-mode, aka., the insert mode

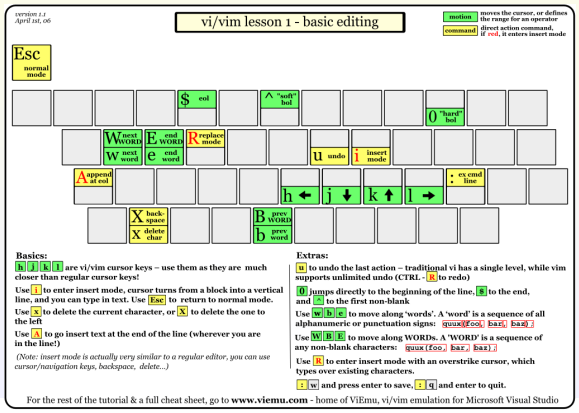
Notes

Vim

- Download & Install GVim or MacVim
- Download & Install tetris.vim
- Download & Install minibufexpl.vim
- Download & Install Gundo
- Download & Install Vim-LaTeX

Notes

Vim



Notes

Git Exercise I

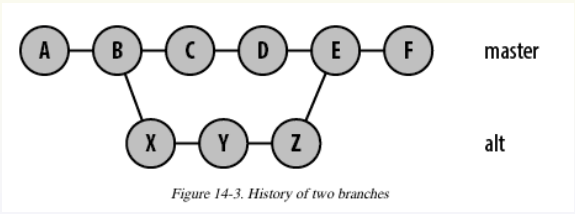


Figure 14-3. History of two branches

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

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

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?

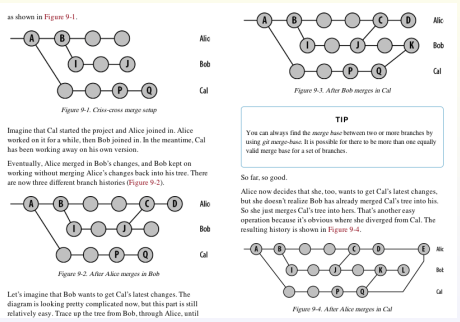
- It allows you to collaborate with others off-line
- You leave a trail of your contributions to the project

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. Criss-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 she diverged from Cal. The resulting history is shown in Figure 9-4.

Figure 9-4. After Alice merges in Cal

Notes

Intro. to Using Git for Off-Line Teamwork III

```
1  git branch alice
2  git checkout alice
3  git remote add altorigin git://git.com/do/not/copy/and/paste/this
4  git push origin alice
5  git branch -D alice
```

```
1  git://github.com/nhlee/550400.stanza1.git
2  git://github.com/nhlee/550400.stanza2.git
3  git://github.com/nhlee/550400.stanza3.git
4  git://github.com/nhlee/550400.stanza4.git
5  git://github.com/nhlee/550400.stanza5.git
6  git://github.com/nhlee/550400.stanza6.git
7  git://github.com/nhlee/550400.stanza7.git
8  git://github.com/nhlee/550400.stanza8.git
```


Notes

Adv. Git Mehod for Off-Line Teamwork I

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

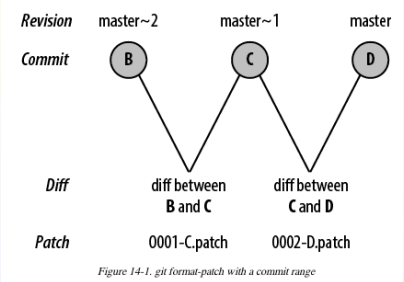


Figure 14-1. git format-patch with a commit range

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 ...

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

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

Notes

Intro. to work-statement template I

```
1  \documentclass[12pt,letterpaper][aritcle]
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, \\
12             &f(\bm x) = \int_0^1 \sin(u+\|\bm x\|) du.
13         \end{align*}
14 \end{document}
```


Notes

Introduction to beamer I

Basic Body Layer

```
\begin{document}
\section{Hello World}
    \subsection{hello world}
        \begin{frame}
            \frametitle{hi world}
            \begin{columns}
                \begin{column}{0.5\textwidth}
                    \begin{itemize}
                        \item Alice!
                    \end{itemize}
                \end{column}
                \begin{column}{0.5\textwidth}
                    \begin{block}{hey world}
                        Bob!
                    \end{block}
                \end{column}
            \end{columns}
        \end{frame}
```


Notes

Introduction to beamer II

```
\end{document}
```

Basic Preambles

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

OR

```
\documentclass[hyperref={colorlinks=false},handout,10pt]{beamer}
\usetheme{Berlin}
\usecolortheme{wolverine}
\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}
```

```
require(tikzDevice)
x = rnorm(100)
plot.ts(x)
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,
framextopmargin=50pt,
}
```

Where to get more help:

http://en.wikibooks.org/wiki/LaTeX/Presentations

Notes

Spurious Causality I

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

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


Notes

Spurious Causality II

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

```
x <- y <- mu <- rep(0,1000)
for(i in 2:1000) mu[i] <- mu[i-1] + rnorm(1)
x <- mu + rnorm(1000)
y <- mu + rnorm(1000)
adf.test(x)$p.value
adf.test(y)$p.value
po.test(cbind(x,y))
```


Notes

Spurious Causality III

```
po.test(cbind(xrates$UK,xrates$EU))
ukeu.lm <- lm(xrates$UK ~ xrates$EU)
ukeu.res <- resid(ukeu.lm)
ukeu.res.ar <- ar(ukeu.res)
ukeu.res.ar$order
```


Notes

How to do software documentation using R

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


Notes

A Word Problem

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 and the club champion alternately: father-champion-father or champion-father-champion, according to Elmer's choice. The champion is a better player than Elmer's father. Which series should Elmer choose?

- 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