lecture_notes

Samantha Alger

January 17, 2017

Lecture Notes

1/17/17

Goals

- think on paper
- computational tools
- core programming methods
- speaking skills

Lab- Wednesdays: 1-3:30

Git Hub

Working locally on your computer, you will push or pull versions to and from GitHub. (push sync to do both). Then, you can 'commit' to create a version that you can later return to if needed.

If your code is not working, you can use the 'Revert' button and Git will send you back to the last commit (previous version of your code).

Homework:

Send NG an email when done setting up your repository:

- github webpage site
- github repository
- Fun Fact- something about me that is memorable

1/19/17

Thinking on Paper!!

Inductive method

observation->hypothesis->prediciton->observation

Hypothesis-Deductive method

Observation with multiple hypotheses and test predictions and refute predictions.

Very uncommon to see people progressing this way. Our starting viewpoint is instead, "Nature is complex". So we shy away from stating simple hypotheses and we see this is in the ecology literature. For example: "relative importance of factors x, y, and z". We don't usually see a quantification of the relative importance of these factors. But, we do see a "complex mix of biotic, abiotic, and historical factors". Another that is popping up is "context-dependent". Of course everything is context dependent. None of these get at a clear hypothesis. Nick calls this "Tangled Bank Fever".

We do not make good progress by embracing this complexity initially.

4 tools for 'thinking on paper'

- path diagram (cause and effect flow chart)
- logic tree
- look up table or DIP switch
- mock data

2017-1-24

Plain-text files-WYSIWYW (what you see is what you want/need) WYSIWYG (what you see is what you get)

plain-text types:

- LaTex- has thousands of typesetting commands, complex
- markdown- very simple version of LaTex (dozen commands instead of thousands of commands)
- Rmarkdown- we can mix in lines of R code and get it to run (filename.Rmd) we can "*knit*" to make file: .pdf, .html, or .docx

yaml- "yet another markdown language" at the top of the page (meta data information)

to knit: use Cntrl + Shift + k

To learn markdown language-

- Help ->Markdown Quick Reference!

This text has *italics*.

This text is **bold**.

Header 1

Header 2

Header 3

Header a million

A friend once said:

> It’s always better to give than to receive.

These words are widely spaced

One line
Next line
Third line

lists are super easy!!!

- first item
- second item
  - what?!
  - double what?!

Yo! let’s use a link:

Alex’s website[1]

Insert a photo here [in all his glory] [Burnham.JPG]

footnote at the bottom of the page[2]

> quoted text is formatted this way

First Header	Second Header
Content Cell	Content Cell
Content Cell	Content Cell

`plaintext` shows this way

```
All the text that we add in here will be exactly like plain text, and not converted to markdown languag
```

Back to markdown formatting

in line r code 6.1415927

To automatically insert R chuck, Use command + alt + i:

```
#Use comments profusely in your code!

Time <- seq(1,10) #seq makes integer sequence
print(Time)       #show contents
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

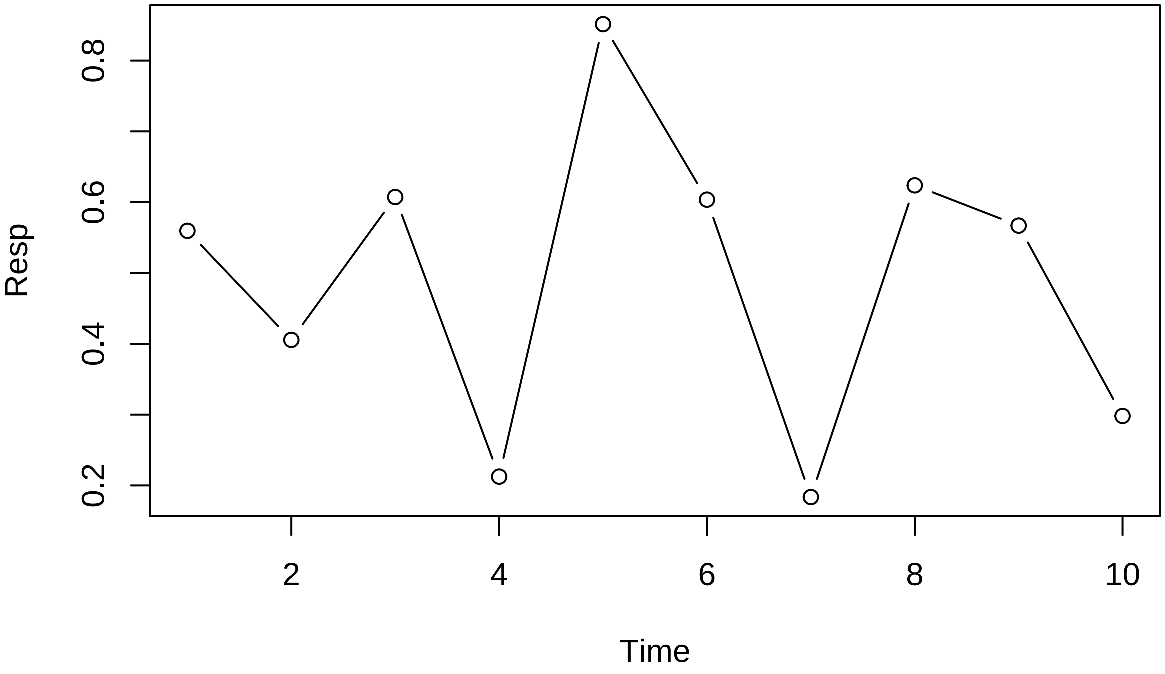
[1] yeah, he’s kind of a big deal

[2] FOOTNOTE!

```
Resp <- runif(10) #random uniform- create 10 random numbers froma uniform distribution
print(Resp)
```

```
##  [1] 0.5595652 0.4055383 0.6073194 0.2124563 0.8514330 0.6036229 0.1835984
##  [8] 0.6237807 0.5669125 0.2980996
```

```
plot(x=Time, y=Resp, type="b")
```


Resp
Time