STAT 215A Fall 2017 Week 2

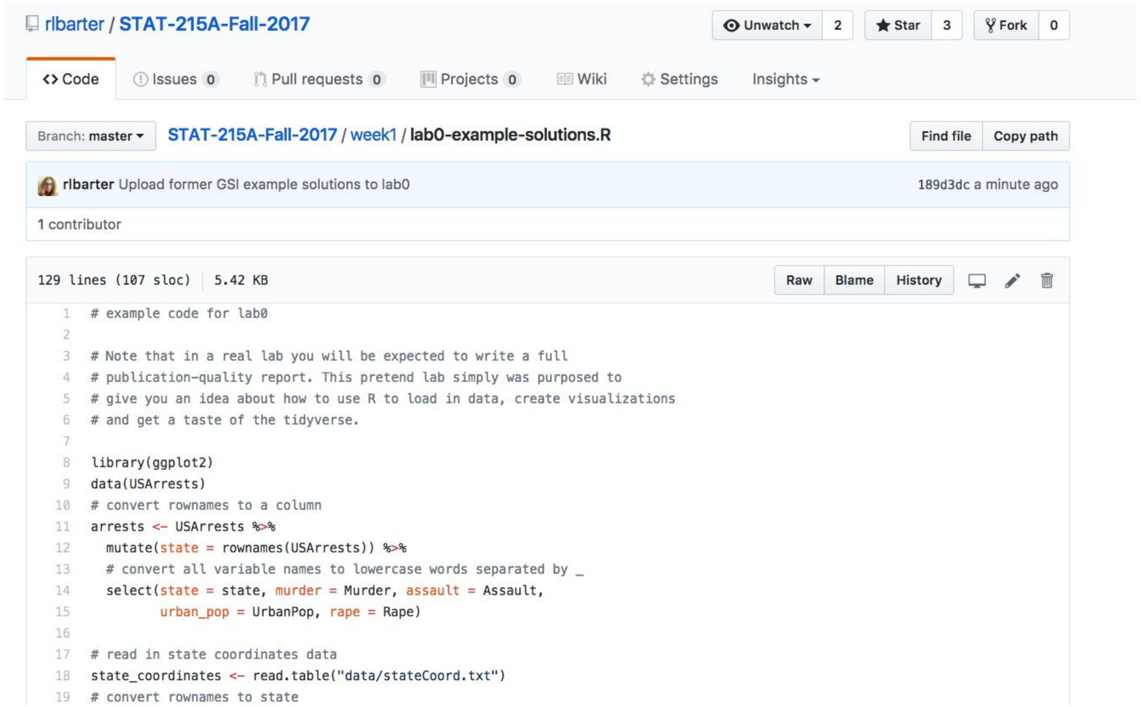
Rebecca Barter
09/01/2017

Announcements

List of github repositories I have access to

actang
Akesari12
alandong93
AlexBrandt
alicegold
amyko
AzizKhiyami
bpark738
candytaco
Chiao-Yu
conniedudu
csinva
dagcilibili
djgrieb
ebenmichael
ehforscher
erickim
feiding92
feynmanliang
g131002
glavrentiadis
gxlilyBerkeley
HectorRDB
Hongxuma
jake-soloff
joeborja
jonathanma1000
jorpro
liv810
Mingjia-Chen
MiyabiIshihara
mostafaharb
mshin03
mxndrwgrdnr
Ningning1992
NYangBerkeley
rneuhausler
Runjing-Liu120
ruonan-hao
rzrsk
shuang0321
spopham
sslama
weichengkuo
xialiyu1995
xiaoli31
xudastar
xurao
xz0416
yaozhewei
yizhouzhao
yzyelse
zoevernon

Lab 0 example code available on GitHub



STAT-215A-Fall-2017 / week1 / lab0-example-solutions.R

rlbarter Upload former GSI example solutions to lab0 189d3dc a minute ago

1 contributor

129 lines (107 sloc) 5.42 KB

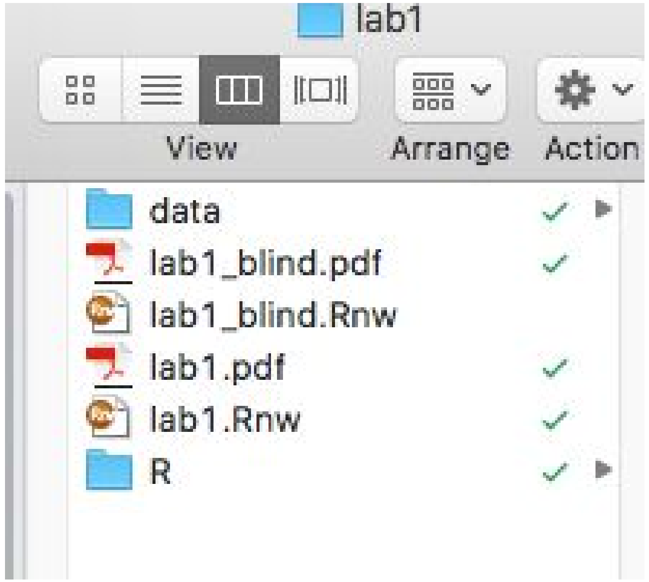
```r
# example code for lab0

# Note that in a real lab you will be expected to write a full
# publication-quality report. This pretend lab simply was purposed to
# give you an idea about how to use R to load in data, create visualizations
# and get a taste of the tidyverse.

library(ggplot2)
data(USArrests)
# convert rownames to a column
arrests <- USArrests %>%
  mutate(state = rownames(USArrests)) %>%
  # convert all variable names to lowercase words separated by _
  select(state = state, murder = Murder, assault = Assault,
         urban_pop = UrbanPop, rape = Rape)

# read in state coordinates data
state_coordinates <- read.table("data/stateCoord.txt")
# convert rownames to state
```

Lab 1 will be released on BCourses today.

Due date: Friday September 15

workflow

Workflow: project folder structure



data ✓
lab1_blind.pdf ✓
lab1_blind.Rnw
lab1.pdf ✓
lab1.Rnw ✓
R ✓

Workflow: project folder structure

R/

load.R - a file containing a function for reading in the data

```
> loadData(path_to_data)
```

clean.R - a file containing a function(s) for cleaning your loaded data

```
> cleanData(loaded_data)
```

data/

Contains the dataset(s)

Workflow: project folder structure

lab1.Rnw: your final report combining code (not printed in output) and text.

Should be written like a paper.

Focus on communicating well and producing high quality **explanatory** figures.

explore.Rmd: a separate (optional) .Rmd file that contains your exploratory figures.

A useful place for exploring the data and saving avenues of exploration that you don’t necessarily want to include in your final report.

Workflow: project folder structure

A quick blog post I wrote on the cleaning workflow project structure:

http://www.rebeccabarter.com/blog/2017-08-16-data-science-workflow/

Workflow: general tips

Make code readable

Be kind to both your peer reviewers and your future self!

Documentation

Write lots of comments in your code. Answer the following questions:

- what does this function do?
- why are you writing this particular piece of code?

Source for tips: http://swcarpentry.github.io/r-novice-gapminder/16-wrap-up/

Workflow: general tips

Keep your code modular

Separate your functions from your analysis file (lab1.Rmd) and store them in R/

You can then load your repository of useful functions into any analysis script for your project (DRY principle)

Group together related functions in the same .R script

Workflow: general tips

Break down each problem into bite-size pieces

Try to solve your problem first for a simple case and then write a generalizable implementation for all cases.

Test your code

Write tests to make sure that your functions are doing the right thing!

Don’t repeat yourself (DRY)

If you find yourself copy-pasting similar lines of code through your project, write a reusable function instead.

Workflow: code style guide

Write your code according to the (modified) Google Style Guide (https://google.github.io/styleguide/Rguide.xml)

Variable names

all lowercase: separate words by "." or "_" (choose one - be consistent!)

- Good: avg.temp, avg_temp,
- Bad: AvgTemp

Workflow: code style guide

Function names

Camel-case, make function names verbs

- Good: CalculateAvgClicks, calculateAvgClicks
- Bad: calculate_avg_clicks, calculate.avg.clicks

Workflow: code style guide

Line length

Maximum line length is 80 characters

Indentation

When indenting your code, use two spaces (rather than tabs)

Workflow: code style guide

Spacing

Place spaces around all binary operators (=, +, -, <-, etc)

- Good:
```
tab.prior <- table(df[df$days.from.opt < 0, "campaign.id"])
```
- Bad:
```
tab.prior=table(df[df$days.from.opt<0,"campaign.id"])
```

Assignment

Use `<-` not `=` for assignment

Workflow: code style guide

Curly Braces

An opening curly brace should never go on its own line; a closing curly brace should always go on its own line.

Always begin the body of a block on a new line.

- Good:
```
if (is.null(ylim)) {
  ylim <- c(0, 0.06)
}
```
- Bad:
```
if (is.null(ylim)) ylim <- c(0, 0.06)
```

Workflow: code style guide

Function Documentation

Functions should contain a comments section immediately below the function definition line.

```
CalculateSampleCovariance <- function(x, y, verbose = TRUE) {
  # Computes the sample covariance between two vectors.
  # Args:
  #   x: One of two vectors whose sample covariance is to be calculated.
  #   y: The other vector. x and y must have the same length, greater than one,
  #      with no missing values.
  #   verbose: If TRUE, prints sample covariance; if not, not. Default is TRUE.
  # Returns:
  #   The sample covariance between x and y.
...
}
```

Workflow: code style guide

The most important tip: be consistent!

tidyverse

GAPMINDER

About Gapminder:

http://www.gapminder.org/about-gapminder/

Resources for this tutorial:

http://swcarpentry.github.io/r-novice-gapminder/08-plot-ggplot2/
http://swcarpentry.github.io/r-novice-gapminder/13-dplyr/

Follow along in **gapminder.Rnw / gapminder.Rmd**

Lab 1 introduction

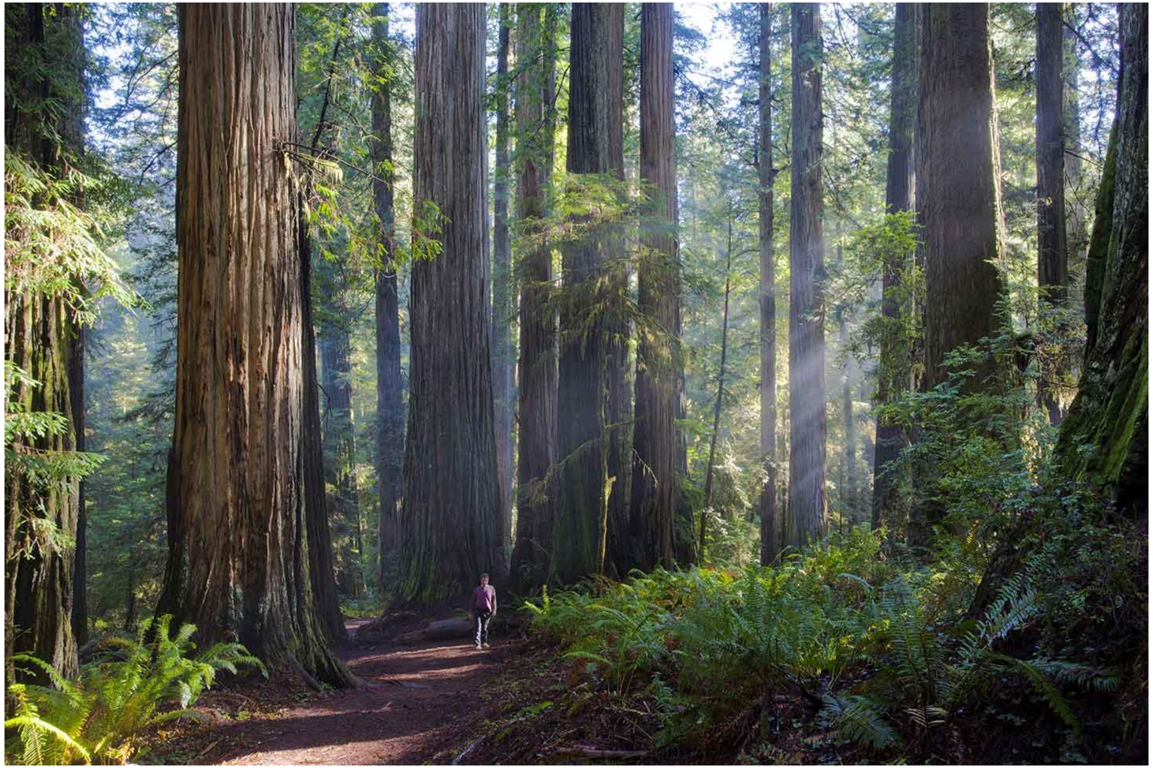
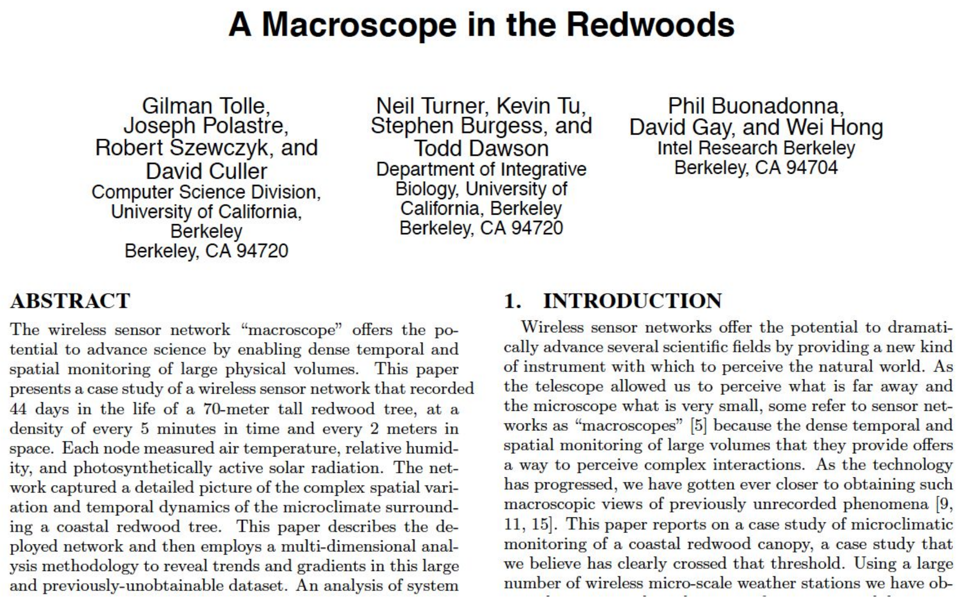


Image source: http://www.redwoodhikes.com/JedSmith/BoyScout1.jpg

Lab 1 introduction

Read the paper carefully: **sensys05-TollePolastreEtAl-redwoods.pdf**



A Macroscope in the Redwoods

Gilman Tolle, Joseph Polastre, Robert Szewczyk, and David Culler
Computer Science Division, University of California, Berkeley
Berkeley, CA 94720

Neil Turner, Kevin Tu, Stephen Burgess, and Todd Dawson
Department of Integrative Biology, University of California, Berkeley
Berkeley, CA 94720

Phil Buonadonna, David Gay, and Wei Hong
Intel Research Berkeley
Berkeley, CA 94704

ABSTRACT

The wireless sensor network "macroscope" offers the potential to advance science by enabling dense temporal and spatial monitoring of large physical volumes. This paper presents a case study of a wireless sensor network that recorded 44 days in the life of a 70-meter tall redwood tree, at a density of every 5 minutes in time and every 2 meters in space. Each node measured air temperature, relative humidity, and photosynthetically active solar radiation. The network captured a detailed picture of the complex spatial variation and temporal dynamics of the microclimate surrounding a coastal redwood tree. This paper describes the deployed network and then employs a multi-dimensional analysis methodology to reveal trends and gradients in this large and previously-unobtainable dataset. An analysis of system

1. INTRODUCTION

Wireless sensor networks offer the potential to dramatically advance several scientific fields by providing a new kind of instrument with which to perceive the natural world. As the telescope allowed us to perceive what is far away and the microscope what is very small, some refer to sensor networks as "macroscopes" [5] because the dense temporal and spatial monitoring of large volumes that they provide offers a way to perceive complex interactions. As the technology has progressed, we have gotten ever closer to obtaining such macroscopic views of previously unrecorded phenomena [9, 11, 15]. This paper reports on a case study of microclimatic monitoring of a coastal redwood canopy, a case study that we believe has clearly crossed that threshold. Using a large number of wireless micro-scale weather stations we have ob-

Lab 1 introduction

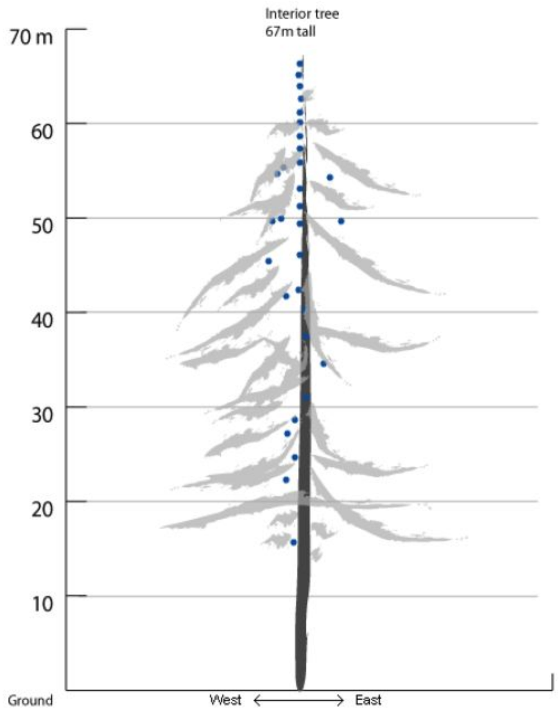


Figure 1: The placement of nodes within the tree

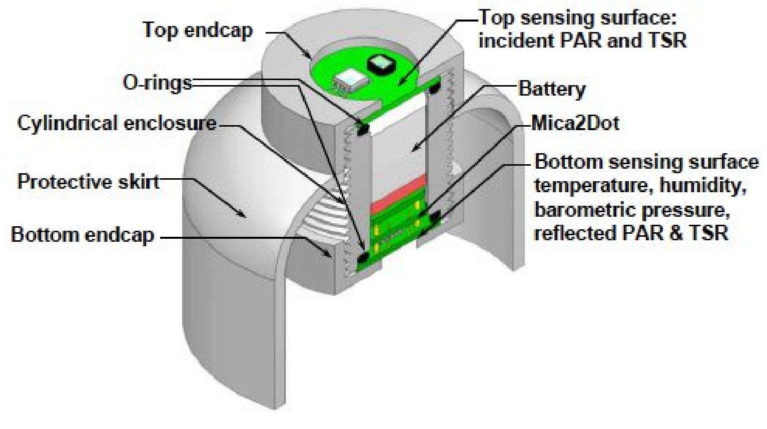
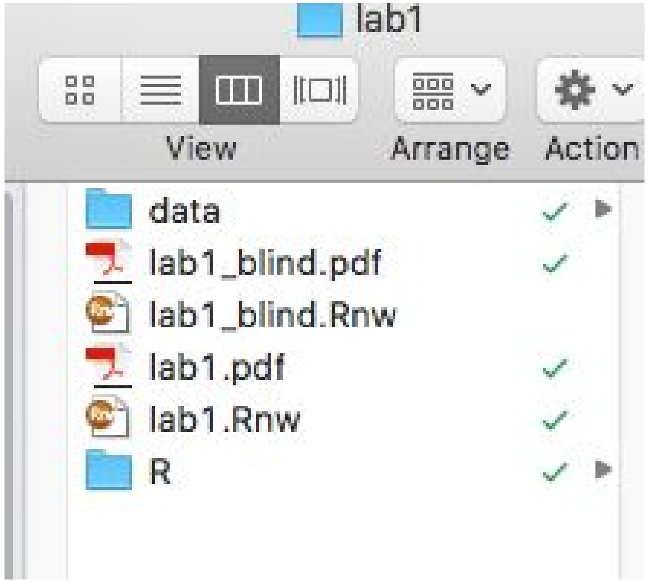

Top endcap
O-rings
Cylindrical enclosure
Protective skirt
Bottom endcap
Top sensing surface: incident PAR and TSR
Battery
Mica2Dot
Bottom sensing surface temperature, humidity, barometric pressure, reflected PAR & TSR

Figure 2: Sensor node and packaging

Workflow: project folder structure



data ✓
lab1_blind.pdf ✓
lab1_blind.Rnw
lab1.pdf ✓
lab1.Rnw ✓
R ✓