STAT 215A - Week 2

Zoe Vernon (08/31/2018)

Thanks to Rebecca Barter for sharing her slides

Announcements

Lab 0 example code available on my STAT-215A-Fall-2018 Github

Sign up for piazza

Make sure your stat-215-a repository is **private** (try uploading image of your ID instead of using your email address to get student account)

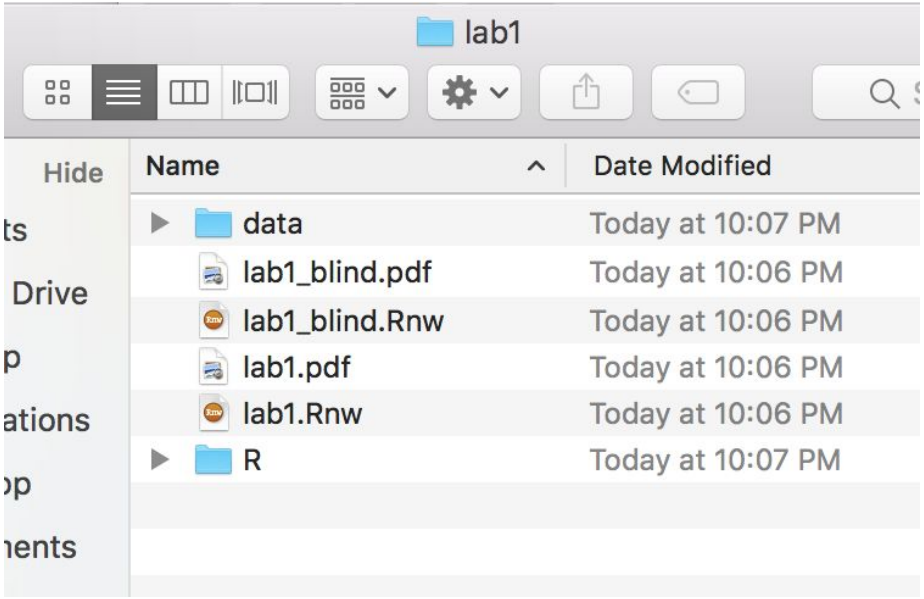
Lab 1 will be released on Bcourses today at the end of the lab section. **Due: Thursday September 13 at 11:59pm**

Github repositories I have access to

adelsonchua
alonamid
Joyyyyyyyyyy
arminaskari
edwardguo61
benjilu
Bassel-Sadek
DWSoriano
DanniDeng
imbrogliodc
ffandong
gaoweichen
yhrhelen
Ibrahimjitani
jpdunc23
JiajianLu
kexinhuang0216
kunaljaydesai
leizhangucb
WaverlyWei
MariusWiggert
Menglingliu
michelleyu1
nicholassim
PhilBoileau
qichen0729
RuiLovesMango
Shunili
sonaliii
taejooahn
tmtang15
tfaulk13
duke00xjunyuangao
xyzjayne
yuchen0105
yulinliu101
ytwangyolanda
lzyaml
zihaochen

Source: http://www.manageartworks.com/features/artwork-approval-workflows/

Project folder structure



Project folder structure cont'd

R/

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

```
> loadData(path_to_data)
```

clean.R - a file containing a functions(s) for cleaning load data

```
> cleanData(loaded_data)
```

data/

Contains the dataset(s)

Not uploaded to GitHub

Project folder structure cont'd

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.

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?

More tips at http://swcarpentry.github.io/r-novice-gapminder/16-wrap-up/

General tips cont'd

Keep your code modular

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

In doing so you create a bank of useful functions that you can load into any analysis script for your project (or future projects)

Use different R scripts for different groups of functions (i.e. load.R, clean.R, etc.)

General tips cont'd

Break down each problem into bite-sized 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

If you find yourself copying and pasting similar lines often, write a function

Code style

Follow Google's R Style Guide for writing your code
https://google.github.io/styleguide/Rguide.xml

Variable names

All lowercase: separate words by "." or "_" (be consistent with the one you choose)

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

Code style cont'd

Function names

Camel-case: use verbs in function name

Good: `CalculateAvgTemp, calculateAvgTemp`
Bad: `calculate_avg_temp, calculateavgtemp`

Line length: Maximum line length 80 characters

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

Code style cont'd

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 `<-` instead of `=`

Code style cont'd

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.
…
}
```

Code style cont’d

Most importantly, be consistent!

About Gapminder: https://www.gapminder.org/about-gapminder/

Resources for this tutorial:

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

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

See gapminder_example in week2 folder on my GitHub

Lab 1 Introduction

Source: http://www.redwoodhikes.com/JedSmith/BoyScout1.jpg

Lab 1 Introduction

Read the paper carefully **sensys05-TollePolastreEtAl-redwoods.pdf** in the lab1 folder on Bcourses

The lab1 folder on Bcourses will also contain a template to follow when putting together your lab as well as loading and cleaning functions that

The explore.Rmd file was put together by last year's GSI (Rebecca) to get you started looking at the data. Do **not** push this file (or your own explore files) to your stat-215-a repo

Lab 1 Introduction

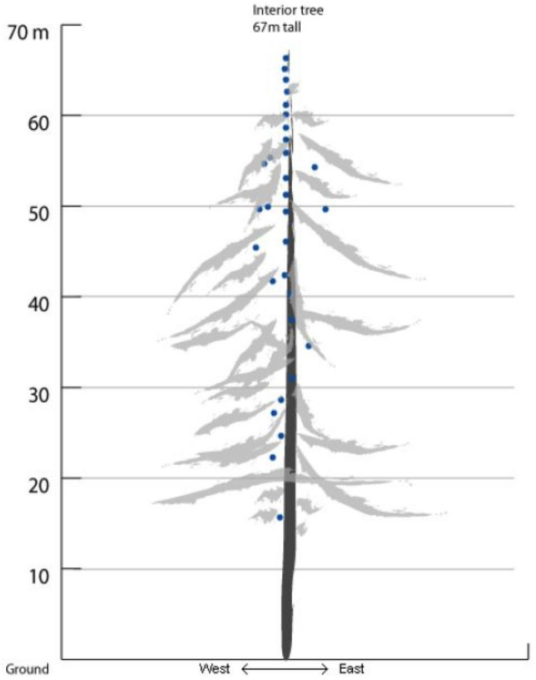


Figure 1: The placement of nodes within the tree

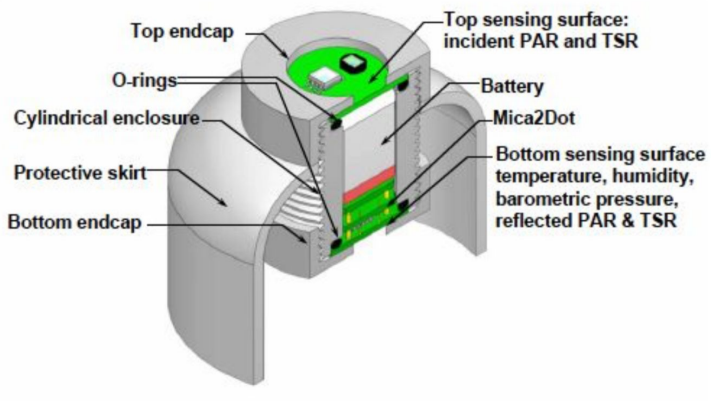


Figure 2: Sensor node and packaging

Lab 1 Rubric

68 points total (8 points for HW and 60 points for the lab)

- Readability and grammar - 10 points
- Reproducibility of report - 4 points
- Readability of code - 4 points
- Data cleaning (description and validity) - 7 points
- Figures that are not for the findings (relevance and quality) - 6 points
- Graphical critique - 3 points
- Three findings (interestingness and quality of figure) - 6 points per finding
- Overall quality and level of detail of report - 8 points