Final Review

Stats 506, F18

12/11/2018

Final Exam

- Monday December 17, 1:30-3:30 PM in this room
- 200 points
- Two 100 point parts
- Part 1, four 25-point questions
- Part 2, four 50-point questions (choose 2)
- You may bring one two-sided piece of 8.5 x 11 inch set of notes

Part 1 Topics

Questions in part 1 will be drawn from the following topics

- Regular Expressions (Passive and Active)
- data.table (Passive and Active)
- dplyr (Passive and Active)
- SQL (Passive and Active)
- SAS (Passive)
- Re-sampling methods: bootstrap or cross validation (Active)
- R basics: matrix operations, vectorization, etc (Active)

Part 2 Topics

- Any topic from part 1 can appear in multiple parts
- Each of the following may appear in at most one question:
  - SAS (Active)
  - Permutation testing
  - Matrix decomposition (SVD and QR)
  - Parallel computing: mclapply, foreach, and/or futures
  - S3 methods

Regular Expressions

- You will be provided with the "help" sheet from regexcrosswords
- Be able to match regular expressions to their strings and write simple regular expressions to solve tasks.
- Know how to use bracketed expressions, or |, negation, bracket keyword groups, anchors, and repetition statements.
- Here is the regexp for a valid object name in R, let's unpack it:

```
^([[:alpha:]]|[.][._[:alpha:]])[._[:alnum:]]*$
```

R basics

- Know how to define R functions and use default parameters.
- Recognize and be able to use the functions `rep()` and `seq()` and `:`.
- Know the arithmetic operators `%%` and `%/%`
- Know and be able to use the `sample()` and `quantile()` functions
- Recognize and be able to use basic string manipulation functions: `paste()`/`paste0()` and `sprintf()`

R matrices and vectorization

- Know that matrices are stored in column major order and how this interacts with the concept of
- Be familiar with matrix indexing
- Know the matrix operations: `%*%`, `t()`, `outer()`, `colSums()`/`colMeans()`, `rowSums()`/`rowMeans()`.
- Recognize and be able to use `apply` functions: `apply`, `sapply`, `lapply` and their relations to `for` loops.

SQL

- Know and be able to use the core statements:

```
SELECT
 FROM
 WHERE
 GROUP BY
 HAVING
 ORDER BY
```

- Understand and be able to use `INNER JOIN` and `LEFT JOIN`
- Recognize aliases

```
SELECT a.some_key as ID, b.value as VALUE
FROM tableA a
LEFT JOIN tableB b
 ON a.some_key = b.some_key
```

SAS

- Know how to use a `libname` statement.
- Understand multilevel file handles, i.e. `mylib.table1`, `work.data0`.
- Recognize and (part two only) be able to use key data step statements.
- Recognize and (part two only) be able to use `proc summary`.
- Know how to use `proc sql` (part two only).
- There will not be SAS regular expressions.

dplyr and tidyr

- Know and be able to use the core dplyr verbs:
  - `filter`
  - `mutate`
  - `transmute`
  - `select`
  - `rename`
  - `group_by`
  - `summarize`
  - `left_join`
- Recognize `spread` and `gather` from tidyr and be able to interpret their output
- Know and be able to use pipes `%>%`

data.table

- Recognize and be able to use the `[i, j, by]` parameters
- Know and understand the difference between `by` and `keyby`
- Recognize when `dt[ , .(j), .(by)]` will have one or more than one row per group
- Understand and be able to update by reference using `:=`
- Recognize and use the special symbols `.N` and `.SD` and know how `.SDcols` interacts with `.SD`
- Recognize and be able to use the `dcast` and `melt` methods for a data.table

Matrix decompositions

- Understand the SVD and QR decomposition
- Know and be able to use syntax for these decomposition in R
- Be able to solve the least squares problem using these decomposition

S3 methods

- Know how to define an S3 method for a given class, i.e. `summary.lm`
- Recognize and be able to use method dispatch
- Know how an S3 generic is defined

```
summary = function(x, ...){
 UseMethod("summary")
}
```

Resampling methods

- Cross validation and the bootstrap are fair game for part 1. One of the two will appear there and the other is likely to appear in part 2.
- Permutation testing will not appear in part 1, but may appear in part 2.
- Be able to write code to implement these methods.
- Know and be able to implement the percentile method for determine bootstrap standard errors.
- Understand how to use the bootstrap with aggregate data.

Parallel computing

- Recognize and be able to use syntax from at least one of packages for parallel computing we have discussed:
  - `parallel::mclapply`
  - `doParallel` and `foreach`
  - `futures`
- Be able to reason about how tasks are divided among parallel processes and to appropriately chunk tasks.