PS6

Mengying Yang

November 1, 2017

Problem 1

1

The goal of their simulation study is to demonstrate the likelihood ratio statistic is converge to a weighted sum of independent chi-square distribution. They test the number of their components in a normal mixture distribution (a single normal versus a two component mixture versus three-component mixture normal distribution). They are trying to look at two questions:
1. How fast does test statistics converge in distribution to asymptotic distribution?
2. What is the power?

2

The choices the authors must make in designing their simulation study are determine:
1. seperation $D = (u1 - u2)/sigma$
2. sample size
3. number of components from case 1 to case 2
(above three affect power)
4. mixing proportion
5. in addtion to that they could change variance but they control it at here, replication(fix 1000 at here too), and norminal level alpha)
(the rest choices are unclear if they affect power)

3

The table only show the data, and rest tables are not very clear.I think if they can make some graph to show the pattern would be better.

4

The table one shows under normal distribution, generating n samples and calculating 2LR at 0.01 and 0.05 nominal level. They want to show that the larger sample size, the proportion of simulation we rejects are more close to the nominal level, especially under adjusted test which isto make the converge faster. Table 2 add distance level. When D = 3, there is no strong evidence the power will depend on the mixing proportion.

5

I think the 10 times simulation is not enough, it will cause the error large. However, when the number of simulation too large, then the cost will be too much. I think the number of times of simulation should depends on when the result begin to converge. I think 1000 at here is a good tradeoff.

Problem 2

```
library(RSQLite)
drv <- dbDriver("SQLite")
dir <-'C://Users/Renjun/Desktop/STAT243/ps6'
dbFilename <- 'stackoverflow-2016.db'
db <- dbConnect(drv, dbname = file.path(dir, dbFilename))
Only_r <- dbGetQuery(db, "SELECT distinct U.userid, U.displayname FROM Users U join questions Q
                     on U.userid = Q.Ownerid join questions_tags T on Q.questionid = T.questionid
                     WHERE tag = 'r' and userid not in (SELECT distinct U.userid FROM Users U
                     join questions Q on U.userid = Q.Ownerid join questions_tags T on
                     Q.questionid = T.questionid  WHERE tag = 'python')")

head(Only_r)

##    userid     displayname
## 1  575952      Thalecress
## 2 5738949         AMedina
## 3 4802680 Dhwani Dholakia
## 4 3507767     user3507767
## 5 2670641   Reuben Mathew
## 6 4148256            Jake

length(Only_r[,1])

## [1] 18611

dbDisconnect(db)
```

Problem 3

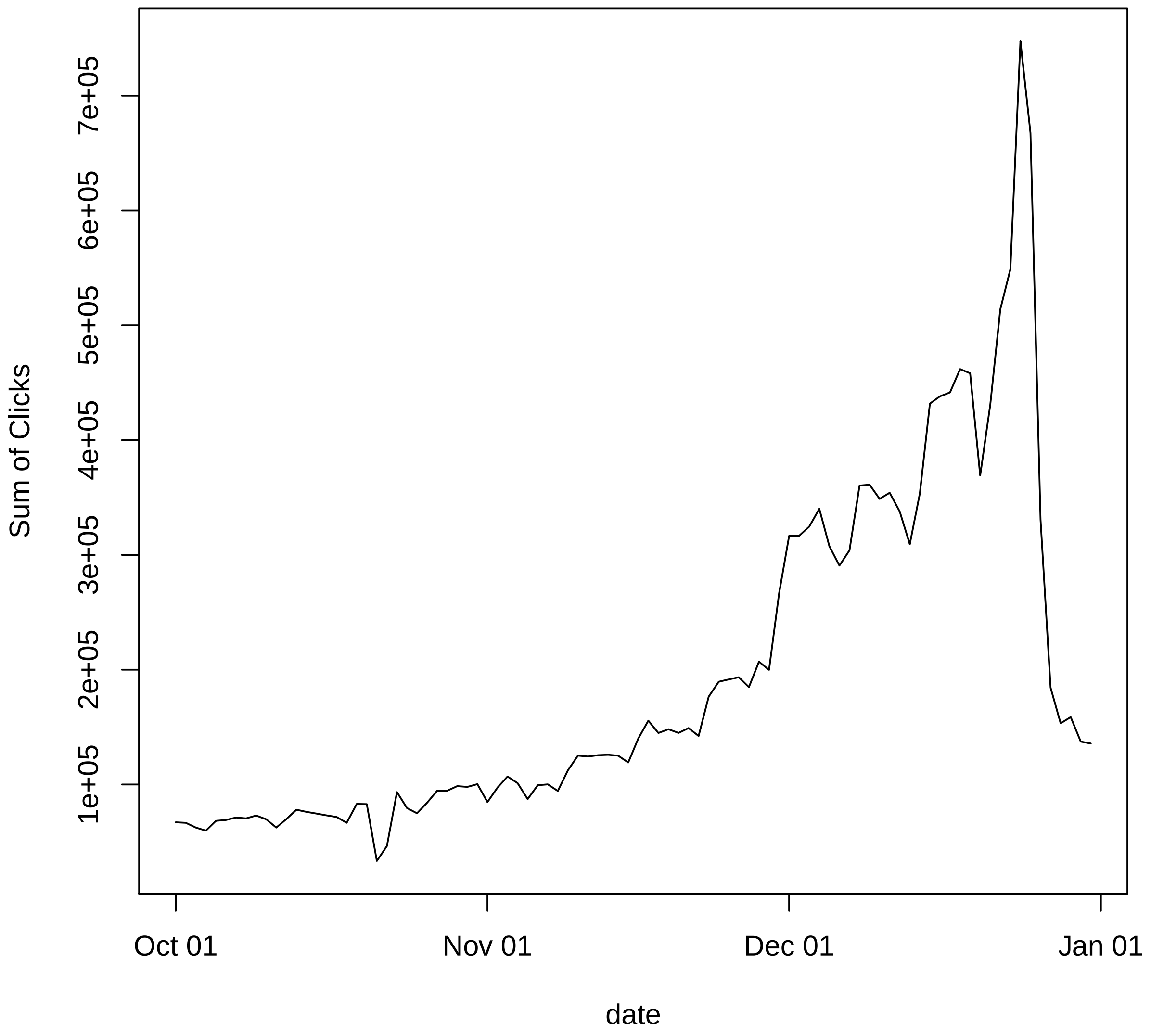
I want to investigate the number of click the Wikipedia that are relevant to Christmas changes over time from October to December. When the most of people begin to think or prepare for Christmas? If the changes over time influence significantly? Does the culture influence?

```
#get acess to the interactive session
ssh mengying_yang@hpc.brc.berkeley.edu
srun -A ic_stat243 -p savio2  --nodes=1 -t 1:00:00 --pty bash

#set up code to get Spark running on these nodes
module load java spark
source /global/home/groups/allhands/bin/spark_helper.sh
spark-start

#set up an interactive python session
module load python/2.6.6
module unload python
pyspark --master $SPARK_URL --executor-memory 60G

dir='/global/scratch/paciorek/wikistats_full'
lines = sc.textFile(dir + '/' + 'dated')
lines.getNumPartitions()
```

```
# filter to the lines related to
import re
from operator import add

#Define the function that can find Christmas related lines
def find(line, regex = "Christmas", language = None):
    vals = line.split(' ')
    if len(vals) < 6:
        return(False)
    tmp = re.search(regex, vals[3])
    if tmp is None or (language != None and vals[2] != language):
        return(False)
    else:
        return(True)

Christmas = lines.filter(find).repartition(480)

def stratify(line):
# create key-value pairs where:
# key = date-time-language
# value = number of website hits
    vals = line.split(' ')
    return(vals[0] + '-' + vals[1] + '-' + vals[2], int(vals[4]))

counts = Christmas.map(stratify).reduceByKey(add)

def transform(vals):
    # split key info back into separate fields
    key = vals[0].split('-')
    return(",".join((key[0], key[1], key[2], str(vals[1]))))

outputdir = '/global/home/users/mengying_yang/Christmas'
counts.map(transform).repartition(1).saveAsTextFile(outputdir)

#check the result and the format is: date, time, language, number of search through Wikipedia
less /global/home/users/mengying_yang/Christmas/part-00000

#copy my file from brc to my local address
scp mengying_yang@dtn.brc.berkeley.edu:/global/home/users/mengying_yang/Christmas/part-00000
~/stat243/ps6/Christmas
```

```
library(stringr)
library(dplyr)

##
## Attaching package:  'dplyr'
## The following objects are masked from 'package:stats':
##
##     filter, lag
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

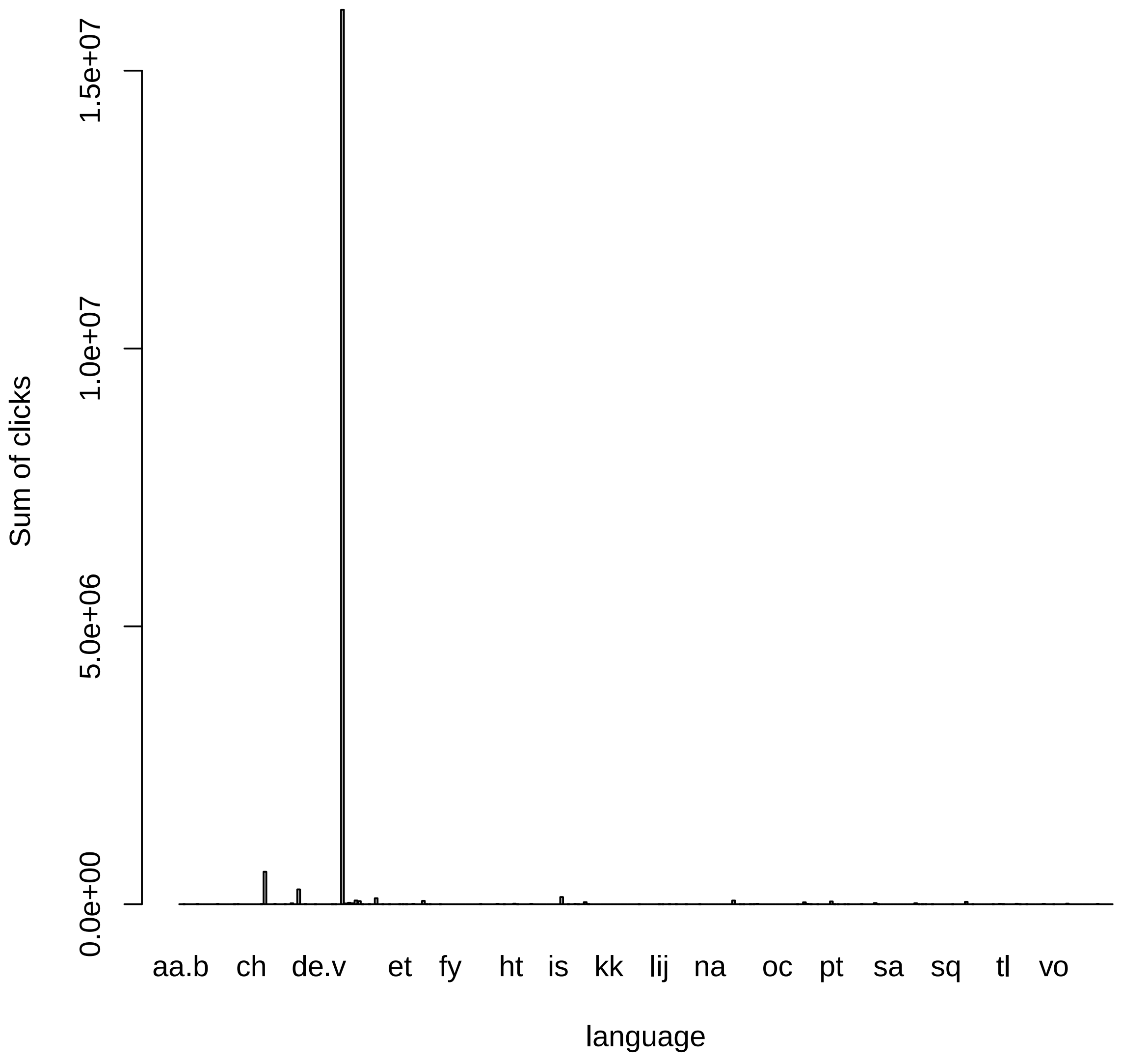
```
Christmas <-read.csv('C://Users/Renjun/AppData/Local/lxss/home/Mengying/stat243/ps6/Christmas',
                     header=FALSE)
names(Christmas) <- c("date", "time", "language", "clicks")

Christmas$date <- as.Date(as.character(Christmas$date),'%Y%m%d')

dateGroup <- Christmas %>% group_by(date) %>% summarise(SumClicks = sum(clicks))
#Group the date therefore we can find how many clicks on the webpage.

tail(dateGroup[order(dateGroup$SumClicks),], n=10 )

## # A tibble: 10 x 2
##          date SumClicks
##        <date>     <int>
##  1 2008-12-21    430773
##  2 2008-12-15    431805
##  3 2008-12-16    438130
##  4 2008-12-17    441545
##  5 2008-12-19    458207
##  6 2008-12-18    461883
##  7 2008-12-22    513952
##  8 2008-12-23    548788
##  9 2008-12-25    667525
## 10 2008-12-24    747523

plot(dateGroup$date, dateGroup$SumClicks, type='l', xlab='date', ylab='Sum of Clicks')
```



```
unique(Christmas$language)

##   [1] es          ru          bn          lmo         it.q
##   [6] no          tr          en.s        simple.q    fi
##  [11] fr.d        fr.v        fi.d        en.b        ja
##  [16] th.s        www         cy          kk          lv
##  [21] eu          de          pl.b        en.d        nl.d
##  [26] ca          pl          it          it.d        ro
##  [31] vi          pl.d        nl          zh          de.d
##  [36] tl          hu          mn          oc          pt
##  [41] ru.d        fr.b        en.v        lt          en
##  [46] fr          en.n        simple      ar          commons.m
##  [51] he          sv          cs.d        nl.s        et
##  [56] hu.d        id          nn          sq          da
##  [61] el.d        hr          th          pt.d        fa
##  [66] pt.s        eo.d        en.q        ms          cs
##  [71] pdc         nl.q        el          hsb         bat
##  [76] lb          ca.d        nl.n        bs          sl
##  [81] tr.d        fy          vo          hi          commons
##  [86] es.d        nb          mk          ka          uk
##  [91] pl.q        ang         pt.b        wo          no.d
##  [96] simple.d    fr.s        pt.q        te          ro.d
## [101] ko          la          ja.d        ta          hr.s
## [106] bg          id.d        sv.d        en2         fo
## [111] eo          de.q        az          nah         meta.m
## [116] ar.d        es.q        is          sr          ko.d
## [121] lij         lad         it.s        ga          sk
## [126] it.b        pt.n        vec         hu.q        ml
## [131] jv          de.s        de.v        zh.d        hif
## [136] species.m   vi.d        el.v        sh          pam
## [141] pag         io.d        it.n        sw          af
## [146] fy.d        kn          ch          gl          sr.d
## [151] www.n       li          tpi         qu          gd
## [156] fur         ga.d        nostalgia   es.b        pa
## [161] chr         ro.s        es.s        pl.s        da.d
## [166] nov         nl.b        uk.d        hy          de.b
## [171] lt.d        sq.d        fr.n        fr.q        mr
## [176] an          sl.s        sa          se          nrm
## [181] br          www.s       no.n        ru.q        sco
## [186] wuu         my          ilo         als         bg.d
## [191] www.d       km          cv          mt          it.v
## [196] oc.d        si          dk          cbk         sources
## [201] su          sv.s        ia          de.n        bar
## [206] ht          na          ur          be          zh.s
## [211] cz          rm          os          tet         io
## [216] tw          vi.b        gn          nds         kw
## [221] bpy         jp          es.n        yi          id.s
## [226] lo          li.s        war         ps          incubator.m
## [231] kr          sv.b        roa         www.w       ku
## [236] la.d        uz          tp.d        xal         cdo
## [241] dv          sk.d        stq         to          cr
## [246] id.q        mg          new         ksh.d       pl.n
## [251] ru.s        zea         ee          ne          pms
## [256] pt.v        jbo         www.q       or          hr.d
## [261] haw         es.v        sah         zh.b        ka.d
## [266] gv          fa.d        th.n        aa.b        zh.q
## [271] nan         hi.d        co          sc          he.d
## [276] crh         scn
## 277 Levels: aa.b af als an ang ar ar.d az bar bat be bg bg.d bn bpy ... zh.s
```

```
LanguageGroup <- Christmas %>% group_by(language) %>% summarise(SumClicks = sum(clicks))
tail(LanguageGroup[order(LanguageGroup$SumClicks),], n=5 )
```

```
## # A tibble: 5 x 2
##    language SumClicks
##      <fctr>     <int>
## 1        es    107185
## 2        it    128041
## 3        de    266185
## 4 commons.m    583430
## 5        en  16096252

barplot(LanguageGroup$SumClicks, names.arg = LanguageGroup$language
        ,xlab='language', ylab='Sum of clicks')
```



As we can see from first the plot, it is very clear that the search that is related to Christmas increase significantly starting from the mid Nov until the Christmas day and the drop rapidly after the Christmas Day. It is very interesting that the change is huge after the day of Christmas. and from the language group we can find English speaking people search much more about Christmas than non-English speaking people. Therefore, the culture can influence the result.

1 Problem 4

1.1 (a)

```
ssh mengying_yang@hpc.brc.berkeley.edu
srun -A ic_stat243 -p savio2  --nodes=1 -t 1:59:59 --pty bash
scp ~/stat243/ps6/ProblemSet6Q4.R mengying_yang@dtn.brc.berkeley.edu:~/ProblemSet6Q4.R
scp ~/stat243/ps6/ProblemSet6Q4.sh mengying_yang@dtn.brc.berkeley.edu:~/ProblemSet6Q4.sh

module load r/3.2.5
module load stringr foreach doParallel

sbatch ProblemSet6Q4.sh
squeue -j 1879337
squeue -u mengying_yang
squeue -A ic_stat243

R
```

```r
#ProblemSet6Q4.R

library(parallel)
library(doParallel)
library(foreach)

nCores <- as.integer(Sys.getenv("SLURM_CPUS_ON_NODE"))
registerDoParallel(nCores)

nSub <- 959

result <- foreach(i = 0:nSub,
                .packages = c("readr","dplyr", "stringr"),
                # libraries to load onto each worker
                .combine = rbind, # how to combine results
                .verbose = TRUE) %dopar% {

                  file <- paste("/global/scratch/paciorek/wikistats_full/dated_for_R/part-",
                                str_pad(i, width=5, side="left", pad="0"), sep="")
                  data <- readr::read_delim(file, delim =" ", col_names=FALSE)
                  data <- as.data.frame(data)
                  Obamaline <- grep("Barack_Obama", data$X4, ignore.case = TRUE)
                  Obamadata <- data[Obamaline,]
                }

dimresult <- dim(result)
headresult <- result[1:10,]

write.table(dimresult, file="/global/home/users/mengying_yang/dimresult.txt")
write.table(headresult, file="/global/home/users/mengying_yang/headresult.txt")
write.table(result, file="/global/home/users/mengying_yang/result.txt")
```

```
#the separate job script file: ProblemSet6Q4.sh

#!/bin/bash
# Job name:
#SBATCH --job-name=test
#
# Account:
#SBATCH --account=ic_stat243
#
# Partition:
#SBATCH --partition=savio
#
# Wall clock limit (30 minutes here):
#SBATCH --time=02:59:59
#
## Command(s) to run:
module load r/3.2.5 doParallel dplyr stringr
R CMD BATCH --no-save PS6_Q4.R PS6_Q4_new.out
```

After downloading the results from BRC,I find we have 433895 ’Barack$_Obama'relatedpagesforOctober - December2008from960files.$

b

When I run the Spark code, it takes me around one and half hours minutes 1 cores to create the filtered dataset that just has the Obama-related webpage traffic using Spark. Assuming that I can achieve perfect scalability (i.e., that if you ran your code in part (a) on 4 times as many cores, it would go 4 times as fast), it Will probablily takes me approximately 25 minutes. PySpark is more effective than parallelized R when doing filter problem.

c

In our case, we have 960 files that have similar size. Dynamic allocation will cost more time on connmunication, therefore prescheduling will be more efficient. However,if I don’ have many files as this case or the tasks with highly variable completion times, the preschedule will not very helpful to improve load-balancing.