HW1

Daniel Wooten

September 11, 2014

1 Problem 1

Please refer to survey submission for proof of problem one completion.

2 Problem 2

2.a

The virtual machine has one processor.

2.b

The virtual machine has 994MiB of memory.

3 Problem 3

3.a

The code that follows can be described as follows.

1. Wget retrieves the file
2. grep pulls the Republican components
3. grep pulls the Democratic components
4. grep pulls the Republican senators
5. grep pulls the Democratic senators
6. sed command to remove extra commas for Republicans
7. sed command to remove extra commas for Democrats
8. sed command to remove various symbols for Republicans
9. sed command to remove various symbols for Democrats

10. cut command to pull the desired information for Republicans
11. cut command to pull the desired information for Democrats
12. sort command to sort by total contributions in reverse order `(-r)`, by numeric value `(-g)`, based upon the 4th field `(--key)` with field seperation denoted by "," `(-t)` for the Republicans
13. sort done same as above for the Democrats
14. head command to display the top 5 for the Republicans
15. head command to display the top 5 for the Democrats

```
wget http://www.fec.gov/data/CandidateSummary.do?format=csv
grep "REP" CandidateSummary.do?format=csv >> R1.csv
grep "DEM" CandidateSummary.do?format=csv >> D1.csv
grep "\"S\"" R1.csv >> R2.csv
grep "\"S\"" D1.csv >> D2.csv
sed 's/\([^",]\),/\1/g' R2.csv > R3.csv
sed 's/\([^",]\),/\1/g' D2.csv > D3.csv
sed 's/[$"]//g' R3.csv > R4.csv
sed 's/[$"]//g' D3.csv > D4.csv
cut -d',' -f 3,5,7,20 R4.csv > R5.csv
cut -d',' -f 3,5,7,20 D4.csv > D5.csv
sort -g -r --key=4 -t , R5.csv > R6.csv
sort -g -r --key=4 -t , D5.csv > D6.csv
head -5 R6.csv
head -5 D6.csv
```

```
## MCCONNELL MITCH,KY,REP,11832783.00
## CORNYN JOHN,TX,REP,9673385.00
## LAND TERRI LYNN,MI,REP,8609528.00
## COTTON THOMAS,AR,REP,6682006.06
## GRAHAM LINDSEY OLIN,SC,REP,6520501.00
## MARKEY EDWARD JOHN MR,MA,DEM,17156080.00
## BOOKER CORY A,NJ,DEM,15498084.00
## HAGAN KAY R,NC,DEM,12744012.00
## GRIMES ALISON  LUNDERGAN,KY,DEM,11056082.00
## UDALL MARK E,CO,DEM,10002963.00
```

3.b Part B

For the desired shell code, please see below.

```
#!/bin/bash
#Script to accomplish part B of question 3

echo "**********************************"
echo "Script Begin"

#We check to see if cn14.zip exists and if not, we download it
if [ ! -e cn14.zip ]; then
      echo "Retreiving file"
      wget ftp://ftp.fec.gov/FEC/2014/cn14.zip
else
      echo "cn14.zip file already downloaded"
fi

#We check to see if indiv14.zip exists and if not, we download it
if [ ! -e indiv14.zip ]; then
      echo "Retreiving file"
      wget ftp://ftp.fec.gov/FEC/2014/indiv14.zip
else
      echo "indiv14.zip file already downloaded"
fi

#We check to see if indiv14.zip is unzipped, if not, do so
if [ ! -e itcont.txt ]; then
      echo "Uniziping file"
      unzip indiv14.zip
else
      echo "File already unziped"
fi

#Receive name to search for from user
echo "Please enter candidate's last name"

read name

echo "You have input candidate" $name

echo "Searching for" $name"'s ID number"

#Search for ID number of given candidate name
ID=$(grep -i -m 1 "|$name" cn.txt | cut -d'|' -f10)

echo $name"'s ID number is" $ID

#This grep will count the number of times that
```

```
# ID shows up, effectively the number of contributions
# nation wide
nat=$(grep -ci "$ID|" itcont.txt)

#This grep will do the same as above with the additional
# critiera of CA as the state
cali=$(grep -i "$ID|" itcont.txt | grep -ic "CA|")

echo "The number of national contributors to "$name"'s campaign is "$nat
echo "The number of California contributors to "$name"'s campaign is "$cali

echo "Script End"
echo "**********************************"
```

With respect to the candidates Mitch McConnell and Alison Lundergan please refer to Table 1.

Table 1: McConnell Vs. Grimes

Region	McConnell	Grimes
National	6911	9461
California	324	1497

4 Problem 4

Please see below for the shell script (with comments). I think that the only parts that need explaining (beyond the comments) are the feeding of foo.txt into the while loop and the wgets. foo.txt contains the grepped txt file names that we want to download. We feed those into the while loop which assigns each text file name, one at a time, to the variable p which we then concatenate with the URL variable to give wget the whole textfile name to download. I thought that was pretty savy.

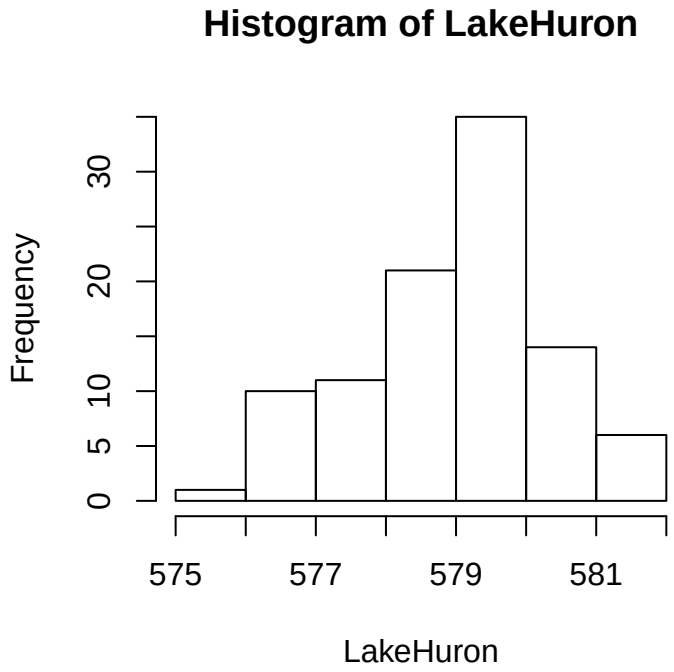
```
#!/bin/bash

echo "**************************************"
echo "Begin Script"

#Set URL shell variable, will be useful

URL=http://www1.ncdc.noaa.gov/pub/data/ghcn/daily/

#Check for the html index file. If absent, retrieve
```

```
 if [ ! -e index.html ]; then
        echo "Index file missing. Retreiving index file"
        wget $URL
 else
        echo "Index file present. Process index file."
 fi

 #Create temporary storage file
 touch foo.txt

 #grep through the index file and feed lines into cut for appropriate trimming
 grep ".txt\"" index.html | cut -d"\"" -f8 > foo.txt

 #Now we loop through the file, foo.txt, which contains the names of all of
 #our text files.

 while read p; do
 #check to see if the file already exists, if not, download
        if [ ! -e $p ]; then
                echo "Retrieving file "$p
                wget $URL$p
        else
                echo "File already exists. Abort download"
        fi
done <foo.txt

 #Clean up the local directory
 rm -f foo.txt

 echo "End Script"
 echo "****************************************"
```

5 Problem 5

The height of the water level in Lake Huron fluctuates over time. Here I 'analyze' the variation using R. I show a histogram of the lake levels for the period 1875 to 1972

```
hist(LakeHuron)
```




```
lowHi <- c(which.min(LakeHuron), which.max(LakeHuron))
yearExtrema <- attributes(LakeHuron)$tsp[1] - 1 + lowHi
```