PS1 - STAT 243

Alexander Fred-Ojala
Student ID: 26958060

September 11th 2015

1 Problem 1

1.1 1. a)

First we download the zip file with wget (and the option flag -O to name it apricots.zip). Then we unzip it using unzip (with -p option so that nothing but file data is sent to stdout):

```
wget -O apricots.zip
"http://data.un.org/Handlers/DownloadHandler.ashx?DataFilter=itemCode:526&
DataMartId=FAO&Format=csv&c=2,3,4,5,6,7&s=countryName:asc,elementCode:asc,year:desc"

unzip -p apricots.zip > apricots.csv
```

To make the sorting easier we remove all the quotation characters and also the four zeros after every figure in the data file. This is done with sed, and then we extract the world data (indicated by a '+'). The country data as the exception (using flag -v with grep), also we do not want to include the first line and the footer, so we eliminate those with with the tail and head commands. According to below:

```
sed 's/"//g' apricots.csv | sed 's/.0000//g' | grep "+" > region.csv

sed 's/"//g' apricots.csv | sed 's/.0000//g' | grep -v "+" | tail -n +2
| head -n -7 > country.csv
```

Then we subset the countries.csv data file to the year 2005 and we grep for 'Area Harvested'. To obtain the result we sort for numerical values in column 6 (delimiter/seperator ',') and then we use tail to see the top 5 countries:

```
grep "2005" country.csv | grep "Area Harvested" | sort -n -t, -k6,6 | tail -n5
```

The procedure can be automated with the following lines put into two scripts (ps1a1.sh, ps1a2 both made executable with *chmod uog+x*). Procedure: To setup the analysis we run '*./ps1a1.sh*' once, then to compare different years we run *./ps1a2.sh*

ps1a1.sh consists of - BASH CODE:

```
wget -O apricots.zip
"http://data.un.org/Handlers/DownloadHandler.ashx?DataFilter=itemCode:526
&DataMartId=FAO&Format=csv&c=2,3,4,5,6,7&s=countryName:asc,elementCode:asc,year:desc"

unzip -p apricots.zip > apricots.csv

sed 's/"//g' apricots.csv | sed 's/.0000//g' | grep "+" > region.csv

sed 's/"//g' apricots.csv | sed 's/.0000//g' | grep -v "+" | tail -n +2
| head -n -7 > country.csv
```

ps1a2.sh (that can be used for automated and repeated analysis, since it only takes user input for the year to analyze and then prints out the result in a list) consists of - BASH CODE:

```
read -p "Enter the year you want to examine : " year

echo "$(grep $year country.csv | grep "Area Harvested" | sort -n -t, -k6,6 | tail -n5)"
```

ANSWER: The rankings have changed much over the years.

1.2 1. b)

We define the function ID. Then we take a user input for the variable number, with the function read. We input the item number in the URL with ${number} to specify that it is a variable and we want the output. We unzip the data and then echo the output on the screen. The stdout of the function can then be used with other commands and stored in other data files. It is just to use e.g. '$ ID > 526data.csv' in order to store the output in a file.

Bash code:

```
wget -O apricots.zip

function ID {

read -p "Enter ID : " number

wget -O file.zip "http://data.un.org/Handlers/DownloadHandler.ashx?
DataFilter=itemCode:${number}
&DataMartId=FAO&Format=csv&c=2,3,4,5,6,7&s=countryName:asc,elementCode:asc,year:desc"

unzip -p file.zip > data.csv

echo "$(cat data.csv)"
}
```

2 Problem 2

We place the code below in a shell script called txtdownload.sh, we change the file with `$ chmod uog+x txtdownload.sh` so that it is executable. We run the file in the terminal with ./txtdownload.sh

Procedure:

First we download the html file with wget (specifically only looking for .html, wget is always -q now so that the only output is what is downloaded). Then we specify the file names and assign them to a variable using grep at the html file (looking for those specific strings that ends in .txt and has a pattern before them, in this case ' href=" ', we remove the patterns around to obtain the file names)

After that we store the url in a variable and then create a for loop where wget fetches the file names stored in the variable and then prints out what file is being downloaded.

—- BASH CODE INSIDE SCRIPT: txtdownload.sh —-

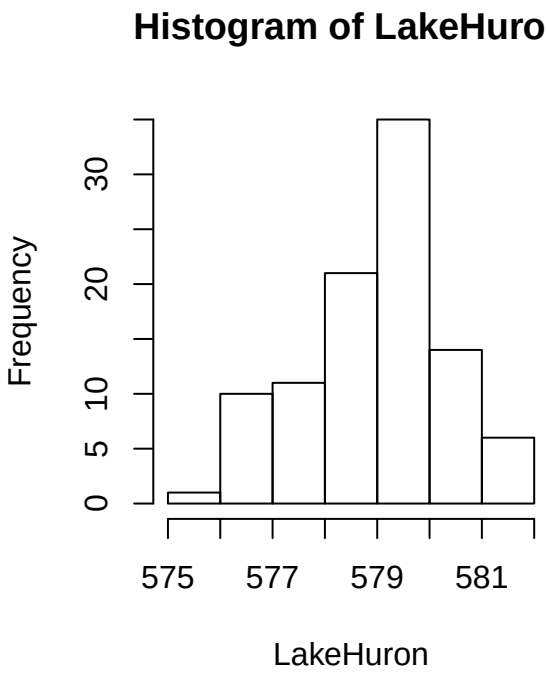
```
wget -q -A html http://www1.ncdc.noaa.gov/pub/data/ghcn/daily/

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

filenames=$(grep -o -P '(?<=href=").*(?=.txt)' index.html | grep -o -P '.*(?=")')

for filenames in ${filenames}
do

    echo "Now downloading ${filenames}"

    wget -q ${url}/${filenames}

done
```

3 Problem 3 (see result at next page)

The result of your solution to Problem 3 should look like this page

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
```