# Problem set 1

Dongyu Lang
SID: 24174288

September, 07, 2017

## 1 Question 2 (a)

Explaination: I first download the file with wget and unzip it. From the data I know that all the countries do not have the symbol ”+”, and regions have it. So I use grep to seperate the data. Furthermore, I use awk to get all the 2005 in the forth column and grep the Area Harvested and get the top five. For the last part of the question, I perform a for loop, and within each loop, I perform the same thing described above, but change the year. The result shows that the ranking for the top 5 countries keeps changing, but most of them stay in the top five position constantly.

```
wget -O p2.csv ”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 p2.csv > q2a.csv
#change the comma in the first column to underscore
sed -e 's/, /_/g' q2a.csv > q1.csv
#create one file for countries, and the other for regions
grep + q1.csv > region.csv
grep -v + q1.csv > country.csv

#get the top 5 countries in year 2005
top5=$(awk -F',' '$4 ~  ”2005”' country.csv | grep 'Area
    Harvested' | sed 's/\”//g' | sort -rn -t',' -k 6 | cut
     -d',' -f1 | sed -n '1,5p' | sed -e 's/_/, /g')

echo $top5

#perform a for loop for each year
for i in $(seq 1965 10 2005)
do
```

```
        echo "${i}: $(awk -F',' '$4 ~  '${i}'' country.
            csv | grep 'Area Harvested' | sed 's/\"//g' |
            sort -rn -t',' -k 6 | cut -d',' -f1 | sed -n
            '1,5p' | sed -e 's/_/, /g')" >> output.txt
done

cat output.txt

## --2017-09-07 19:08:04--  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
## Resolving data.un.org... 85.159.207.229
## Connecting to data.un.org|85.159.207.229|:80...
   connected.
## HTTP request sent, awaiting response... 200 OK
## Length: 68264 (67K) [application/zip]
## Saving to:   p2.csv
##
##      0K .......... .......... .......... ..........
   .......... 75%  230K 0s
##     50K .......... ......
                                       100% 97.1K=0.4s
##
## 2017-09-07 19:08:07 (171 KB/s) -   p2.csv   saved
   [68264/68264]
##
## Turkey Iran, Islamic Republic of Pakistan Uzbekistan
   Algeria
## 1965: USSR
## Turkey
## United States of America
## Spain
## Tunisia
## 1975: USSR
## Turkey
## Spain
## Tunisia
## Italy
## 1985: Turkey
## USSR
## Spain
## Iran, Islamic Republic of
## Tunisia
## 1995: Turkey
## Iran, Islamic Republic of
## Spain
## Ukraine
## Tunisia
## 2005: Turkey
## Iran, Islamic Republic of
## Pakistan
## Uzbekistan
## Algeria
```

The result shows that the top 5 countries keeps changing, but most of them are relatively stable to be in top 5 for these years

# 2 Question 2 (b)

Explaination: If the number of argument is not one, then return invalid number of arguments; if the argument is ”-h”, return the usage; else, download the file and unzip it; if there is no data except title, then return the code is wrong; otherwise read the data.

```
{ function showCSV () { if [ $# != ”1” ] ; then echo ”
   invalid number of arguments”; elif [ $1 = ”-h” ] ;
   then echo ”usage: myfun [num]” ; else wget -O showCSV.
   csv 'http://data.un.org/Handlers/DownloadHandler.ashx?
   DataFilter=itemCode:'$1'&DataMartId=FAO&Format=csv&c
   =2,3,4,5,6,7&s=countryName:asc,elementCode:asc,year:
   desc'; unzip -p showCSV.csv > temp.csv; tail -n +2
   temp.csv > content.csv; check=$(cut -d',' -f1 content.
   csv | wc -l); if [ $check -eq ”1” ] ; then echo ”there
    is no csv file under the item code”; else less temp.
   csv; fi; fi; }; }
```

# 3 Question 3

Explaination: I first get all the txt file name in the URL, and adding the file name after the URL to get the URL for the txt file. I perform a for loop, and in each loop, I download a file and echo that I have downloaded the file.

```
havetxt=$(curl -s https://www1.ncdc.noaa.gov/pub/data/
   ghcn/daily/ | grep -o '\[.*.txt' | cut -c 26- | grep -
   o '.*”' | sed 's/”//g')
```

```
for i in $havetxt
do
        wget -O $i 'https://www1.ncdc.noaa.gov/pub/data/
           ghcn/daily/'$i''
        echo ”Status: finished downloading '$i'”
done
```

# 4 Question 4 code

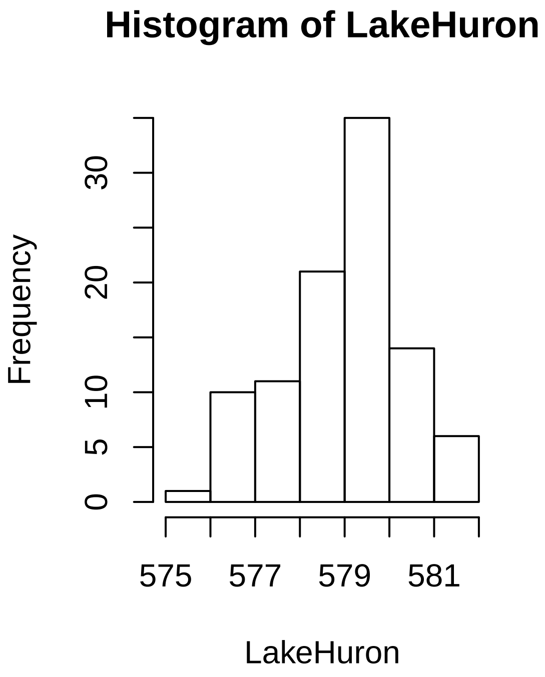
The code for section 4 is here:

```
#\let\oldsection\section

#\renewcommand\section{\clearpage\oldsection}

#\section{Question 4}

#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 \Sexpr{start(LakeHuron)[1]} to \Sexpr{end
   (LakeHuron)[1]}.

#<<r-plot, fig.width=3, fig.height=4>>=

#hist(LakeHuron)

#lowHi <- c(which.min(LakeHuron), which.max(LakeHuron))

#earExtrema <- attributes(LakeHuron)$tsp[1]-1 + lowHi

#@
```

## 5 Question 4

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
```