Problem Set 1

HSIEH CHENG HAN
henry50618@berkeley.edu

2.(a)

```
curl "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" -o "data.zip"
# Download the zip file from the url link and rename it as "data.zip"
unzip -p data.zip > data.csv
# Decompress the zip file as data.csv
grep + data.csv > region.csv
# Extract the regions of the world (ex:Africa) into a new file since those names are followed by “+”
grep -v + data.csv > country.csv
# Extract the individual countries into a new file since those names are not followed by “+”
grep 2005 country.csv > 2005.csv
# Subset the country-level data to the year 2005
grep Area 2005.csv > area.csv
# Subset the data of which the element is "Area Harvested"
sed 's/\"//g' area.csv | sort -n -t ',' -k 6 | tail -n 5
# First, we use sed command to remove the double quote on each field. Then, we use sort to compare the number on the sixth filed(-k option means column, -t means delimiter)
# Finally, use tail command to show the five countries with greatest values
# The result includes Turkey, Pakistan, Uzbekistan, Algeria and Spain
for ((i=1965;i<=2005;i+=10));do
grep $i country.csv | grep Area | sed 's/\"//g' | sort -n -t ',' -k 6 | tail -n 5 >> file.txt
done
# We automate the aggregate process and loop it from the year 1965 to 2005, then put the stdout into file.txt
less file.txt
# Check the content in the file
# We can observe that there is a little difference of countries among different years
```

(b)

```
function agri(){
# Define a function called agri
if [ "$1" == "-h" ]
# Print the introduction of this function if user types "-h"
then
echo "This function asks you to enter a valid item code as input. The output will be the data in csv format otherwise a warning signal will appear."
exit 0
# Exit the function
fi
curl "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" -o "data.zip"
# Download the zip file by passing the number of item code as the first argument ($1)
unzip -p data.zip > data.csv
# download the zip file from the url and unzip it as data.csv
byte=$(ls -la data.csv | cut -d' ' -f8)
# variable "byte" records the size of the file in bytes.
# If the url is invalid, the downloaded file will be a nearly empty file with approximate 244 bytes
if [ $byte -lt 1000 ]
# If the file is less than 1000 bytes, the url is invalid
then
echo "You've passed an invalid number. Please Try again."
fi
less data.csv
# Check the content in the csv file
}
```

3.

```
curl https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/ > data.html
# Download the whole HTML file as "data.html"
less data.html
# Check the HTML file, we observe that the critical words like "ghcnd-countries.txt" are dispersed among the file.
data=$( grep .txt data.html | cut -d"" -f8)
# We try to filter the txt files, the result will look like "ghcnd-countries.txt","readme.txt" and so on
for i in $data;do
# Loop each item (ex."readme.txt") in $data
echo $i;
# Print the name of the item
curl -o "$i" https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/$i ; done
# Download each item by adding its name on the tail of the url link
```

4.

\documentclass{article}

\begin{document}

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]}.

 # Below is the R chunk

<<>>=

hist(LakeHuron)

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

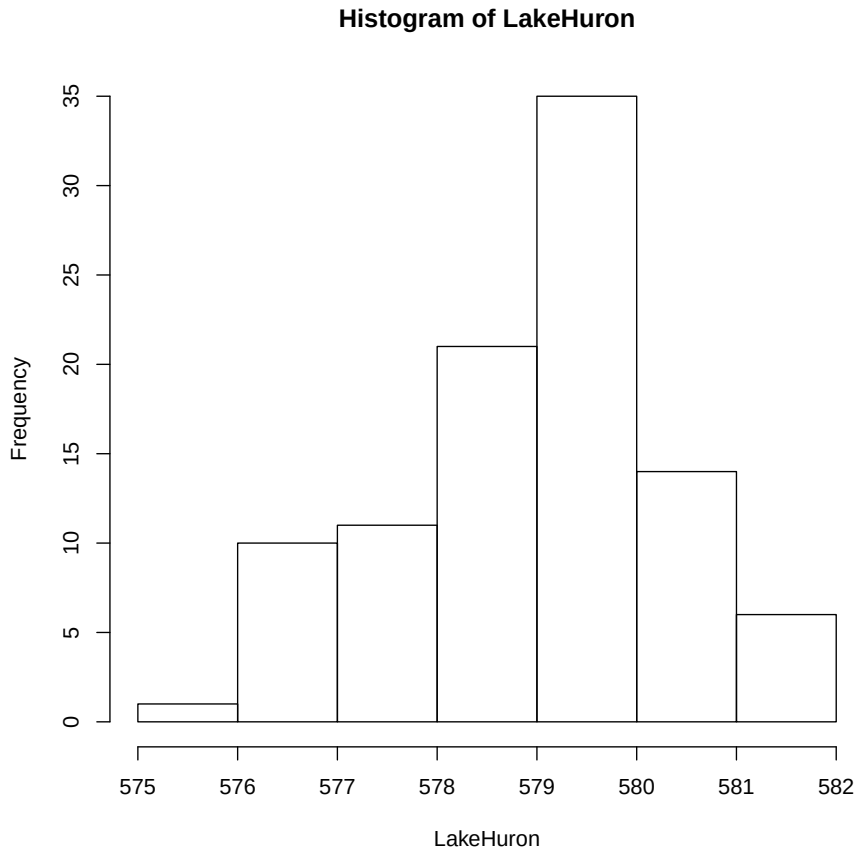
yearExtrema <- attributes(LakeHuron)$tsp[1]-1 + lowHi

@

\end{document}

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
```