```
---
title: "hw1"
author: "Ramon Daniel Crespo Chanis"
date: "9/6/2018"
SID: 3033083874
output: html_document
---
```

Problem 3
The objective of this proble is to programatically download and interpret weather data for a specific location and obtained in last 10 years from the website https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/by_year/. To achieve the objective, a working directory is generated in the Desktop of the user. All operations are run inside this folder. The code consists on four parts.
PART A -> downloading the data from the past 10 years
PART B -> processsing the data for the station DEATH VALLEY and for the data value TMAX
PART C -> Generating a boxplot plot by using R
PART D -> Introduce a variable that promps the user for input and re-do part A,B,C progrmatically for the input provided by the user

The most relevant shell functions used in this code are:
curl
gunzip
grep
echo
sed

To loop through the data for loops were used.

PART 3.A Generate directory and store the data in the new working directory

~~~
```{bash}
#PART A
#a.1 generate working directory and download data to working directory
cd ~
mkdir -p ~/Desktop/Stat243_ps1/data
cd ~/Desktop/Stat243_ps1/data
for e in "2009" "2010" "2011" "2012" "2013" "2014" "2015" "2016" "2017"
do
  curl -O
https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/by_year/$e.csv.gz
done

#a.2 unzip folders by using gunzip
for e in "2009" "2010" "2011" "2012" "2013" "2014" "2015" "2016" "2017"
do
  gunzip $e.csv
~~~

~~~
done

#a.3 return number of lines for each year
for e in "2009" "2010" "2011" "2012" "2013" "2014" "2015" "2016" "2017"
do
  echo "The number of data points for year $e is"
  grep -c "," $e.csv
done
```

PART 3.B Generate directory and store the data in the new working directory
```{bash}
#part 3.b.1 -> download station information into the working directory
cd ~/Desktop/Stat243_ps1/data
curl -O https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/ghcnd-stations.txt

#part 3.b.2 -> generate station id for DEATH VALLEY and for march (03) and store all relevant data into a new file called DV_compiled.csv
station=$(grep "DEATH VALLEY" ~/Desktop/Stat243_ps1/data/ghcnd-stations.txt | cut -d " " -f1)
echo "The station ID for Death Valley is"
echo $station

for e in "2009" "2010" "2011" "2012" "2013" "2014" "2015" "2016" "2017" "2018"
do
  grep $station ~/Desktop/Stat243_ps1/data/$e.csv | grep ${e}03 | grep "TMAX" >> DV_compiled.csv
done

#part 3.b.3 -> return the user the number of datapoints detected for DEATH VALLEY for the month of march
echo "The number of data points found for DEATH VALLEY for the month of march is "
entries=$(grep -c $station DV_compiled.csv)
echo $entries

#part 3.b.4 -> Separate the date information from yyyymmdd to yyyy,mm,dd and store it in a new file called clean.csv
sed 's/./&,/16;s/./&,/19' DV_compiled.csv > clean.csv
```


PART 3.C Graph results by using r
```{r}
library("dplyr")
mydata <- read.table("~/Desktop/Stat243_ps1/data/clean.csv", header=FALSE,sep=",")
mydata
~~~

~~~
boxplot(mydata$V6 ~ mydata$V4, xlab = "day in march", ylab = "Maximum temperature (tenths of degrees C)", main = "Max temp vs Day")
```
~~~

PART 3.D automate tasks 3.A, 3.B, 3.C. This section of the code introduces a function that prompts the user for some information. The correct format to input the information is:
NAME OF STATION,VALUE TO BE EXTRACTED,MONTH. All in caps. If the information is not correct, the code will return an error

~~~
```{bash}
#plis input the string in the function bellow in the format "STATION NAME,VALUE OF INTEREST,MONTH". Ex "DEATH VALLEY,TMAX,03"

string="DEATH VALLEY,TMIN,03"

get_weather(){
#THIS FUNCTION SEEKS THE SPECIFIED WEATHER DATA FROM THE PAST 10 YEARS AND RETURNS A BOX PLOT
#CONTAINING HOW THE TREND IN THE SPECIFIED DATA STRING
#THERE ARE THREE PARTS OF THE FUNCTION:
#PART1 - USER INPUT
#PART2 - DOWNLOAD AND PROCESSING THE DATA
#PART3 - GRAPH THE DATA USING R

#PART1
#part 0 -> will generate a working directory in you desktop where everything will be stored
cd ~
mkdir -p ~/Desktop/Stat243_ps1/data
cd ~/Desktop/Stat243_ps1/data

#1.1-Seek user input and store it into a .csv file
echo $string > test.csv

curl -O https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/ghcnd-stations.txt

#1.2-Decompose the input and store it in variables : input_station, input_variable, station_id,
input_station=$(cut -d "," -f1 ~/Desktop/Stat243_ps1/data/test.csv)
echo "The input station specified is -> "$input_station

input_variable=$(cut -d "," -f2 ~/Desktop/Stat243_ps1/data/test.csv)
echo "The input variable to be extracted from the data is -> "$input_variable

input_period=$(cut -d "," -f3 ~/Desktop/Stat243_ps1/data/test.csv)
echo "The month selected to generate the data is -> "$input_period
~~~

```
station_id=$(grep "$input_station" ~/Desktop/Stat243_ps1/data/ghcnd-
stations.txt | cut -d " " -f1)

#1.3-Test if input_station produces a desired result
echo "Review of input"

if [ -n "$station_id" ]; then
    echo "1 - Succesfully extracted the station specified from the data
set"
else
    echo "1 - Sorry, that station name does not exist. Tray again and
use caps."
fi

#1.4 - Test if input variable produces a desired result
#1.4.1 - Remove previous generated data
# rm ~/Desktop/Stat243_ps1/data/available_data_type.csv

#1.4.2-Generate a list of all possible variable. I generated it and
uploaded to the cloud for
#computational speed
fileid="1355ZLAhKqCpI4PPKDDd0BBK8okyKF9QJ"
filename="available_data_type.csv"
curl -c ./cookie -s -L "https://drive.google.com/uc?
export=download&id=${fileid}" > /dev/null
curl -Lb ./cookie "https://drive.google.com/uc?
export=download&confirm=`awk '/download/ {print $NF}'
./cookie`&id=${fileid}" -o ${filename}


#1.4.3-Test if variable exists within the list of possible variables
var_available=$(grep ""
~/Desktop/Stat243_ps1/data/available_data_type.csv)
bool=false
for item in $var_available
do
    if [ "$input_variable" == "$item" ]; then
        echo "2 - Succesfully extracted the Input Variable from the data
set"
        bool=true;
    fi
done

if [ "$bool" == false ]; then
        echo "2 - Sorry, input variable not found in the available
options. Tray again and use caps."
fi

bool=false
for item in "01" "02" "03" "04" "05" "06" "07" "08" "09" "10" "11" "12"
do
```

~~~
    if [ "$input_period" == "$item" ]; then
        echo "3 - Succesfully extracted the Input Period from the data
set"
        bool=true;
    fi
done

if [ "$bool" == false ]; then
        echo "3 - Sorry, Input period not found in the available
options. Tray again and mm format. Ex: march = 03"
fi

#PART2
#Part2.1 - Get data for the past 10 years from the website provided
for e in "2009" "2010" "2011" "2012" "2013" "2014" "2015" "2016" "2017"
do
  curl -O
https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/by_year/$e.csv.gz
done

for e in "2009" "2010" "2011" "2012" "2013" "2014" "2015" "2016" "2017"
do
  gunzip $e.csv
done

#Part2.2 - Get station ids
curl -O https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/ghcnd-
stations.txt

#Part2.3 - Extract desired data from databases and temporarily store it
in a file named
#DV_compiled_p3d.csv
for e in "2009" "2010" "2011" "2012" "2013" "2014" "2015" "2016" "2017"
"2018"
do
  grep $station_id ~/Desktop/Stat243_ps1/data/$e.csv | grep
${e}${input_period} | grep ${input_variable} >> DV_compiled_p3d.csv
  rm $e.csv
done

#Part2.4 - Clean data. Noticing the constant format of the data
downloaded the data is parsed
#to aid reading the data file
sed 's/./&,/16;s/./&,/19' DV_compiled_p3d.csv > clean_p3d.csv

}


get_weather
```
~~~

PART 3.D. graph results

~~~
```{r}
library("dplyr")
mydata <- read.table("~/Desktop/Stat243_ps1/data/clean_p3d.csv",
header=FALSE,sep=",")
mydata
boxplot(mydata$V6 ~ mydata$V4, xlab = "day in march", ylab = "Maximum
temperature (tenths of degrees C)", main = "Max temp vs Day")
```
~~~

PROBLEM 4

The objective of this problem is to automatically download all the files ending in .txt from the website

~~~
```{bash}

mkdir -p ~/Desktop/problem4_rc
cd ~/Desktop/problem4_rc

curl -L https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/ > index
grep -o 'href.*txt</a>' index > file_list
grep -o '".*"' file_list > file_list2
sed 's/^"\(.*\)".*/\1/' file_list2 > file_list3.csv

var_available=$(grep "" file_list3.csv)
for item in $var_available
do
    echo "Downloading the file $item"
    curl -O https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/$item.txt
done

```
~~~