

Problem Set #1

Todd Faulkenberry

9/6/2018

Question 3

Part A

In order to process the data and return the number of observations in a quick way for multiple years, I wrote a function that takes in one argument(the year), unzips the data, stores it in a .csv, and returns the number of lines in the new .csv.

```
death_valley_year () {  
  curl https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/by_year/$1.csv.gz | gunzip > $1.csv  
  wc -l $1.csv  
}  
  
death_valley_year 2016  
death_valley_year 2017  
death_valley_year 2018
```

Part B

First, I combined the datasets

```
join -1 1 -2 1 -t , 2016.csv 2017.csv > six_seven.csv  
join -1 1 -2 1 -t , six_seven.csv 2018.csv > data.csv
```

Then I called the station id for the station located in Greenland, CA (which is the the station for Death Valley in the dataset.) Though I spent about an hour thinking about it, I couldn't figure out a way to call awk in a function while simultaneously calling parameters, so I decided to not use one.

```
curl -o https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/ghcnd-stations.txt > stations.txt  
  
station_id = $(uniq stations.txt | sort | awk '$5 ~ /GREENLAND/' | awk '{ print $1 }')
```

Part C and Part D

Ran out of time and didn't have the chance to finish Part C and Part D. Sorry, this is completely my fault! I'll budget more time for future problem sets.

Question 4

The below code uses wget to grab all files of a certain extension (-A) to get all .txt files. The -r makes the code recursive, so it runs until it grabs all files. It displays information for the files as it downloads them, though the downloads go quickly so the information can be tough to follow.

```
wget -r -A -# '*.txt' https://www1.ncdc.noaa.gov/pub/data/ghcn/daily
```