Problem Set 1 - STAT 243

Malvika Rajeev

9/6/2018

The solutions to problem set 1.

Question 1 part (b) and (c)

First two steps:

```
cd ~/Desktop/statistical_computing/repository  #this is my main directory
mkdir temp
cd temp
```

To actually understand and visualise the dataset, I looked through the data for 2018, and then to create the boxplot I grouped the data for the last five years.

```
curl -O https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/2018.csv.gz
gunzip 2018.csv.gz
head 2018.csv
```

The first column clearly indicates some kind of an area code, and the third column has specific temperature indications.

Finding the area code using the text file on the website

Again, using curl, I downlaoded the stations text file.

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

```
##   % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
##                                  Dload  Upload   Total   Spent    Left  Speed
##
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
  0 8994k    0 14231    0     0  23003      0  0:06:40 --:--:--  0:06:40 22990
  6 8994k    6  550k    0     0   340k      0  0:00:26  0:00:01  0:00:25  339k
 16 8994k   16 1518k    0     0   580k      0  0:00:15  0:00:02  0:00:13  580k
 30 8994k   30 2737k    0     0   756k      0  0:00:11  0:00:03  0:00:08  756k
 45 8994k   45 4057k    0     0   875k      0  0:00:10  0:00:04  0:00:06  875k
 61 8994k   61 5495k    0     0   978k      0  0:00:09  0:00:05  0:00:04 1096k
 77 8994k   77 7003k    0     0  1058k      0  0:00:08  0:00:06  0:00:02 1290k
 94 8994k   94 8503k    0     0  1116k      0  0:00:08  0:00:07  0:00:01 1396k
100 8994k  100 8994k    0     0  1134k      0  0:00:07  0:00:07 --:--:-- 1453k
```

After this, I just looked for the line with the word "DEATH" and stored the first field (the area code) as a variable.

```
grep Death stations.txt
code=$(grep DEATH stations.txt | cut -d" " -f1)
echo "$code"
```

```
## USC00042319
```

Then, looking for the code in the 2018 file and then using pipes to look for "TMAX" and the month March. (and saving the segmented part as a new file)

```
grep $code 2018.csv | grep TMAX | grep ,201803> 2018marchtmax.csv
head 2018marchtmax.csv
```

Question 1 part (d)

The main function follows the same logic as above, except for function syntax, storing new arguments, etc. As specificed in the question, the first, second, third, fourth and fifth arguments are the location, weather variable, start year, end year and month respectively.

```
function get_weather(){
  curl -O https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/ghcnd-stations.txt
  mv ghcnd-stations.txt stations.txt
  code=$(grep $1 stations.txt | cut -d" " -f1)
  if ["$1"=="-h"]; then; echo "Please enter a valid argument. Enter the location as specified in
  if [ -z "$code" ]; then;
    echo "Weather station not entered correctly. Please enter in ALL CAPS."; return 1; fi
  if [ "$#" -ne 5 ]; then; echo "Enter the correct number of arguments, in the correct order"; return 1

  for i in $(seq $3 $4)
    do
    curl -O https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/by_year/$i.csv.gz
    gunzip $i.csv.gz
  done

  for i in $(seq $3 $4); do
    grep "$code" "$i".csv | grep ,"$i""$5"" | grep "$2">"$1"and"$2"and"$5".csv;
    rm "$i".csv
    done
}
```

PLOTTING THE BOXPLOT

I narrowed the data for March, 2014-2018,(tmax), basically exactly like I did for 2018, except that I created a for loop:

```
for i in $(seq 2014 2018)
do
curl -O https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/by_year/"$i".csv.gz
gunzip $i.csv.gz
done
```

Then, saving all the data according to the specifications, and concatenating into a single CSV file.

```
for i in $(seq 2014 2018)
do
grep "$code" "$i".csv | grep TMAX | grep "$i"03 > "$i"marchtmax.csv
rm $i.csv
done
cat 201*marchtmax.csv > plotdata.csv
for i in $(seq 2014 2018)
do
```

```
rm "$i"marchtmax.csv
done
```

```
head plotdata.csv
```

```
## USC00042319,20140301,TMAX,189,,,7,0800
## USC00042319,20140302,TMAX,222,,,7,0800
## USC00042319,20140303,TMAX,217,,,7,0800
## USC00042319,20140304,TMAX,244,,,7,0800
## USC00042319,20140305,TMAX,289,,,7,0800
## USC00042319,20140306,TMAX,289,,,7,0800
## USC00042319,20140307,TMAX,311,,,7,0800
## USC00042319,20140308,TMAX,283,,,7,0800
## USC00042319,20140309,TMAX,300,,,7,0800
## USC00042319,20140310,TMAX,311,,,7,0800
```

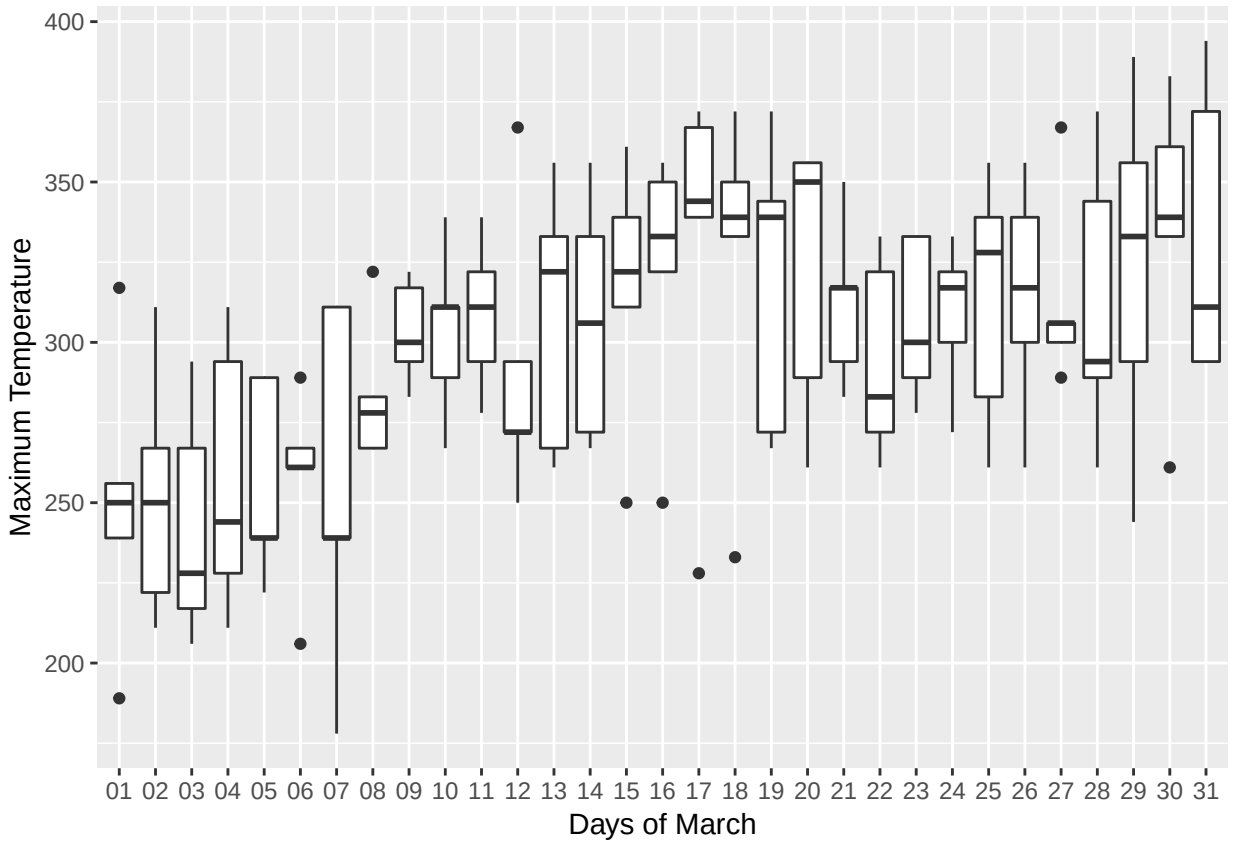
Now, creating the boxplot using R:

```
library(readr)
marchdata <- read_csv("plotdata.csv",col_names = FALSE)
```

```
## Parsed with column specification:
## cols(
##   X1 = col_character(),
##   X2 = col_integer(),
##   X3 = col_character(),
##   X4 = col_integer(),
##   X5 = col_character(),
##   X6 = col_character(),
##   X7 = col_integer(),
##   X8 = col_character()
## )
```

```
date <- marchdata$X2
maxtemp <- marchdata$X4


library(ggplot2)
ggplot() + geom_boxplot(aes(x=factor(substr(date, 7, 8)), y=maxtemp))+labs(x="Days of March",y="Maximum
```



PART 4 - Finding the text files.

For this part, I downloaded the source code of the page and have it read as a standard text file. After that, we loop through all the lines after segmenting it on the basis of .txt and using the sed command to replace the surrounding html chunks.

```
page=$(curl -s https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/)

for t in $(echo "$page" | grep ".txt" | sed 's/^.*\(href.*\)/\1/g' | sed 's/.txt.*//' | cut -c 7-)
do
curl -O https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/${t}.txt
echo "Downloading ${t}.txt"
done
```

```
##   % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
##                                  Dload  Upload   Total   Spent    Left  Speed
##
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100  3670  100  3670    0     0   8935      0 --:--:-- --:--:-- --:--:--  8929
## Downloading ghcnd-countries.txt
##   % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
##                                  Dload  Upload   Total   Spent    Left  Speed
##
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
  0 26.7M    0  118k    0     0   152k      0  0:02:59 --:--:--  0:02:59  152k
```

```
  4 26.7M    4 1297k    0     0   761k      0  0:00:35  0:00:01  0:00:34  761k
 12 26.7M   12 3516k    0     0  1300k      0  0:00:21  0:00:02  0:00:19 1300k
 26 26.7M   26 7282k    0     0  1965k      0  0:00:13  0:00:03  0:00:10 1965k
 46 26.7M   46 12.4M    0     0  2719k      0  0:00:10  0:00:04  0:00:06 2719k
 66 26.7M   66 17.8M    0     0  3200k      0  0:00:08  0:00:05  0:00:03 3678k
 78 26.7M   78 21.0M    0     0  3209k      0  0:00:08  0:00:06  0:00:02 4041k
 89 26.7M   89 23.9M    0     0  3168k      0  0:00:08  0:00:07  0:00:01 4170k
 99 26.7M   99 26.6M    0     0  3130k      0  0:00:08  0:00:08 --:--:-- 3994k
100 26.7M  100 26.7M    0     0  3125k      0  0:00:08  0:00:08 --:--:-- 3598k
## Downloading ghcnd-inventory.txt
##   % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
##                                 Dload  Upload   Total   Spent    Left  Speed
##
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100  1086  100  1086    0     0   2647      0 --:--:-- --:--:-- --:--:--  2648
## Downloading ghcnd-states.txt
##   % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
##                                 Dload  Upload   Total   Spent    Left  Speed
##
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
  0 8994k    0 30159    0     0  51388      0  0:02:59 --:--:--  0:02:59 51378
  9 8994k    9  896k    0     0   589k      0  0:00:15  0:00:01  0:00:14  589k
 23 8994k   23 2084k    0     0   827k      0  0:00:10  0:00:02  0:00:08  827k
 38 8994k   38 3490k    0     0   992k      0  0:00:09  0:00:03  0:00:06  992k
 52 8994k   52 4732k    0     0  1048k      0  0:00:08  0:00:04  0:00:04 1048k
 65 8994k   65 5896k    0     0  1062k      0  0:00:08  0:00:05  0:00:03 1181k
 77 8994k   77 6959k    0     0  1068k      0  0:00:08  0:00:06  0:00:02 1213k
 86 8994k   86 7810k    0     0  1032k      0  0:00:08  0:00:07  0:00:01 1135k
 96 8994k   96 8662k    0     0  1014k      0  0:00:08  0:00:08 --:--:-- 1030k
100 8994k  100 8994k    0     0   998k      0  0:00:09  0:00:09 --:--:--  948k
## Downloading ghcnd-stations.txt
##   % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
##                                 Dload  Upload   Total   Spent    Left  Speed
##
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100   270  100   270    0     0    644      0 --:--:-- --:--:-- --:--:--   644
## Downloading ghcnd-version.txt
##   % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
##                                 Dload  Upload   Total   Spent    Left  Speed
##
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
  9 3707k    9  351k    0     0   316k      0  0:00:11  0:00:01  0:00:10  316k
 85 3707k   85 3179k    0     0  1548k      0  0:00:02  0:00:02 --:--:-- 1547k
100 3707k  100 3707k    0     0  1664k      0  0:00:02  0:00:02 --:--:-- 1664k
## Downloading mingle-list.txt
##   % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
##                                 Dload  Upload   Total   Spent    Left  Speed
##
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100 26498  100 26498    0     0  52399      0 --:--:-- --:--:-- --:--:-- 52471
## Downloading readme.txt
##   % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
##                                 Dload  Upload   Total   Spent    Left  Speed
```

```
##
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100 31860  100 31860    0     0  65867      0 --:--:-- --:--:-- --:--:-- 65826
## Downloading status.txt
```

PART 5(B)

Basically used the existing R object "data", converted it into a numpy array, then summed it along the column.

```
library(reticulate)
np <- import("numpy", convert = FALSE)
arr <- np$array(c(date))
sum <- arr$cumsum()
print(sum)
```

```
## [  20140301   40280603   60420906   80561210  100701515  120841821
##   140982128  161122436  181262745  201403055  221543366  241683678
##   261823991  281964305  302104620  322244936  342385253  362525571
##   382665890  402806210  422946531  443086853  463227176  483367500
##   503507825  523648151  543788478  563928806  584069135  604209465
##   624349796  644500097  664650399  684800702  704951006  725101311
##   745251617  765401924  785552232  805702541  825852851  846003162
##   866153474  886303787  906454101  926604416  946754732  966905049
##   987055367 1007205686 1027356006 1047506327 1067656649 1087806972
##  1107957296 1128107621 1148257947 1168408274 1188558602 1208708931
##  1228859261 1249009592 1269169893 1289330195 1309490498 1329650802
##  1349811107 1369971413 1390131720 1410292028 1430452337 1450612647
##  1470772958 1490933270 1511093583 1531253897 1551414212 1571574528
##  1591734845 1611895163 1632055482 1652215802 1672376123 1692536445
##  1712696768 1732857092 1753017417 1773177743 1793338070 1813498398
##  1833658727 1853819057 1873979388 1894149689 1914319991 1934490294
##  1954660598 1974830903 1995001209 2015171516 2035341824 2055512133
##  2075682443 2095852754 2116023066 2136193379 2156363693 2176534008
##  2196704324 2216874641 2237044959 2257215278 2277385598 2297555919
##  2317726241 2337896564 2358066888 2378237213 2398407539 2418577866
##  2438748194 2458918523 2479088853 2499259184 2519439485 2539619787
##  2559800090 2579980394 2600160699 2620341005 2640521312 2660701620
##  2680881929 2701062239 2721242550 2741422862 2761603175 2781783489
##  2801963804 2822144120 2842324437 2862504755 2882685074 2902865394
##  2923045715 2943226037 2963406360 2983586684 3003767009 3023947335
##  3044127662 3064307990 3084488319 3104668649 3124848980]
```

```
py_to_r(sum)
```

```
##   [1]   20140301   40280603   60420906   80561210  100701515  120841821
##   [7]  140982128  161122436  181262745  201403055  221543366  241683678
##  [13]  261823991  281964305  302104620  322244936  342385253  362525571
##  [19]  382665890  402806210  422946531  443086853  463227176  483367500
##  [25]  503507825  523648151  543788478  563928806  584069135  604209465
##  [31]  624349796  644500097  664650399  684800702  704951006  725101311
##  [37]  745251617  765401924  785552232  805702541  825852851  846003162
##  [43]  866153474  886303787  906454101  926604416  946754732  966905049
##  [49]  987055367 1007205686 1027356006 1047506327 1067656649 1087806972
##  [55] 1107957296 1128107621 1148257947 1168408274 1188558602 1208708931
##  [61] 1228859261 1249009592 1269169893 1289330195 1309490498 1329650802
##  [67] 1349811107 1369971413 1390131720 1410292028 1430452337 1450612647
##  [73] 1470772958 1490933270 1511093583 1531253897 1551414212 1571574528
##  [79] 1591734845 1611895163 1632055482 1652215802 1672376123 1692536445
##  [85] 1712696768 1732857092 1753017417 1773177743 1793338070 1813498398
```

```
##  [91] 1833658727 1853819057 1873979388 1894149689 1914319991 1934490294
##  [97] 1954660598 1974830903 1995001209 2015171516 2035341824 2055512133
## [103] 2075682443 2095852754 2116023066 2136193379 2156363693 2176534008
## [109] 2196704324 2216874641 2237044959 2257215278 2277385598 2297555919
## [115] 2317726241 2337896564 2358066888 2378237213 2398407539 2418577866
## [121] 2438748194 2458918523 2479088853 2499259184 2519439485 2539619787
## [127] 2559800090 2579980394 2600160699 2620341005 2640521312 2660701620
## [133] 2680881929 2701062239 2721242550 2741422862 2761603175 2781783489
## [139] 2801963804 2822144120 2842324437 2862504755 2882685074 2902865394
## [145] 2923045715 2943226037 2963406360 2983586684 3003767009 3023947335
## [151] 3044127662 3064307990 3084488319 3104668649 3124848980
```

******END**