PS1

ShubeiWang

8/30/2018

3

For question(a)-(c), I used the weather data in 2015-2018. Firstly I used 'curl' command and a for loop to download the files I needed. Then I subseted to the station corresponding to Death Valley, to TMAX and to March and put them into a single file named 'DVtmaxMarch'. At last I created an R chunk to read the file and make a single plot of side-by-side boxplots.

For question(d), I wrote a shell function that takes four arguments: a string for identifying the location, the weather variable of interest, the years of interest and the month of interest, and put the data into a file named weather_data

(a)

```
## download yearly climate data from 2015 to 2018 and report the
## number of observations in each year

for ((i=5;i<=8;i++))
do
curl -o 201$i.csv.gz https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/by_year/201$i.csv.gz
gzip -d 201$i.csv.gz
count=$(cat 201$i.csv | wc -l)
echo "There are$count observations in 201$i"
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
  0  192M    0 78148    0     0  73452      0  0:45:43  0:00:01  0:45:42 73447
  0  192M    0  552k    0     0   268k      0  0:12:12  0:00:02  0:12:10  268k
  1  192M    1 2287k    0     0   750k      0  0:04:22  0:00:03  0:04:19  750k
  2  192M    2 5412k    0     0  1335k      0  0:02:27  0:00:04  0:02:23 1334k
  5  192M    5  9.9M    0     0  2022k      0  0:01:37  0:00:05  0:01:32 2068k
  8  192M    8 17.0M    0     0  2889k      0  0:01:08  0:00:06  0:01:02 3490k
 13  192M   13 25.4M    0     0  3706k      0  0:00:53  0:00:07  0:00:46 5124k
 17  192M   17 34.1M    0     0  4350k      0  0:00:45  0:00:08  0:00:37 6547k
 22  192M   22 43.1M    0     0  4890k      0  0:00:40  0:00:09  0:00:31 7779k
 27  192M   27 52.4M    0     0  5344k      0  0:00:36  0:00:10  0:00:26 8696k
 32  192M   32 61.5M    0     0  5709k      0  0:00:34  0:00:11  0:00:23 9119k
 36  192M   36 70.9M    0     0  6030k      0  0:00:32  0:00:12  0:00:20 9304k
 41  192M   41 80.3M    0     0  6309k      0  0:00:31  0:00:13  0:00:18 9457k
 46  192M   46 89.8M    0     0  6553k      0  0:00:30  0:00:14  0:00:16 9561k
 51  192M   51 99.3M    0     0  6764k      0  0:00:29  0:00:15  0:00:14 9615k
 56  192M   56  107M    0     0  6864k      0  0:00:28  0:00:16  0:00:12 9410k
 60  192M   60  116M    0     0  7015k      0  0:00:28  0:00:17  0:00:11 9388k
 65  192M   65  126M    0     0  7158k      0  0:00:27  0:00:18  0:00:09 9375k
 70  192M   70  135M    0     0  7307k      0  0:00:26  0:00:19  0:00:07 9422k
 75  192M   75  145M    0     0  7420k      0  0:00:26  0:00:20  0:00:06 9393k
 80  192M   80  154M    0     0  7526k      0  0:00:26  0:00:21  0:00:05 9656k
 85  192M   85  164M    0     0  7630k      0  0:00:25  0:00:22  0:00:03 9726k
 90  192M   90  173M    0     0  7705k      0  0:00:25  0:00:23  0:00:02 9681k
 94  192M   94  181M    0     0  7748k      0  0:00:25  0:00:24  0:00:01 9430k
 99  192M   99  190M    0     0  7807k      0  0:00:25  0:00:25 --:--:-- 9362k
100  192M  100  192M    0     0  7807k      0  0:00:25  0:00:25 --:--:-- 9228k
## There are 35233244 observations in 2015
##   % 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  192M    0 51252    0     0  64895      0  0:51:44 --:--:--  0:51:44 64875
  0  192M    0  510k    0     0   298k      0  0:10:58  0:00:01  0:10:57  298k
  1  192M    1 2042k    0     0   749k      0  0:04:22  0:00:02  0:04:20  749k
  2  192M    2 4729k    0     0  1271k      0  0:02:34  0:00:03  0:02:31 1271k
  4  192M    4 8635k    0     0  1828k      0  0:01:47  0:00:04  0:01:43 1828k
  7  192M    7 13.5M    0     0  2439k      0  0:01:20  0:00:05  0:01:15 2820k
 10  192M   10 20.1M    0     0  3071k      0  0:01:04  0:00:06  0:00:58 4019k
 15  192M   15 28.9M    0     0  3847k      0  0:00:51  0:00:07  0:00:44 5540k
 20  192M   20 38.6M    0     0  4551k      0  0:00:43  0:00:08  0:00:35 6996k
 25  192M   25 48.2M    0     0  5086k      0  0:00:38  0:00:09  0:00:29 8171k
 29  192M   29 57.3M    0     0  5481k      0  0:00:35  0:00:10  0:00:25 8954k
 34  192M   34 66.6M    0     0  5830k      0  0:00:33  0:00:11  0:00:22 9529k
 39  192M   39 75.8M    0     0  6113k      0  0:00:32  0:00:12  0:00:20 9604k
 43  192M   43 83.8M    0     0  6248k      0  0:00:31  0:00:13  0:00:18 9186k
 48  192M   48 92.3M    0     0  6432k      0  0:00:30  0:00:14  0:00:16 9044k
 52  192M   52  101M    0     0  6588k      0  0:00:29  0:00:15  0:00:14 8958k
 56  192M   56  109M    0     0  6691k      0  0:00:29  0:00:16  0:00:13 8709k
 61  192M   61  118M    0     0  6848k      0  0:00:28  0:00:17  0:00:11 8716k
 66  192M   66  127M    0     0  7003k      0  0:00:28  0:00:18  0:00:10 9087k
 71  192M   71  137M    0     0  7147k      0  0:00:27  0:00:19  0:00:08 9253k
 76  192M   76  146M    0     0  7265k      0  0:00:27  0:00:20  0:00:07 9391k
 81  192M   81  156M    0     0  7388k      0  0:00:26  0:00:21  0:00:05 9716k
 86  192M   86  166M    0     0  7495k      0  0:00:26  0:00:22  0:00:04 9788k
 91  192M   91  175M    0     0  7583k      0  0:00:25  0:00:23  0:00:02 9752k
 96  192M   96  185M    0     0  7670k      0  0:00:25  0:00:24  0:00:01 9729k
100  192M  100  192M    0     0  7729k      0  0:00:25  0:00:25 --:--:-- 9752k
## There are 35384539 observations in 2016
##   % 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
  0  189M    0  274k    0     0   217k      0  0:14:52  0:00:01  0:14:51  217k
  1  189M    1 2055k    0     0   902k      0  0:03:34  0:00:02  0:03:32  902k
  2  189M    2 5290k    0     0  1620k      0  0:01:59  0:00:03  0:01:56 1620k
  5  189M    5  9.8M    0     0  2366k      0  0:01:21  0:00:04  0:01:17 2366k
  8  189M    8 16.1M    0     0  3149k      0  0:01:01  0:00:05  0:00:56 3321k
 13  189M   13 24.6M    0     0  4032k      0  0:00:48  0:00:06  0:00:42 4997k
 17  189M   17 33.6M    0     0  4704k      0  0:00:41  0:00:07  0:00:34 6417k
 21  189M   21 40.9M    0     0  5067k      0  0:00:38  0:00:08  0:00:30 7311k
 26  189M   26 49.3M    0     0  5459k      0  0:00:35  0:00:09  0:00:26 8105k
 30  189M   30 58.6M    0     0  5847k      0  0:00:33  0:00:10  0:00:23 8687k
 36  189M   36 68.1M    0     0  6197k      0  0:00:31  0:00:11  0:00:20 8909k
 40  189M   40 77.1M    0     0  6442k      0  0:00:30  0:00:12  0:00:18 9021k
 45  189M   45 86.5M    0     0  6678k      0  0:00:29  0:00:13  0:00:16 9348k
 50  189M   50 95.7M    0     0  6871k      0  0:00:28  0:00:14  0:00:14 9486k
 55  189M   55  104M    0     0  7036k      0  0:00:27  0:00:15  0:00:12 9476k
 60  189M   60  114M    0     0  7196k      0  0:00:26  0:00:16  0:00:10 9446k
 65  189M   65  123M    0     0  7320k      0  0:00:26  0:00:17  0:00:09 9475k
 70  189M   70  132M    0     0  7427k      0  0:00:26  0:00:18  0:00:08 9413k
 75  189M   75  142M    0     0  7551k      0  0:00:25  0:00:19  0:00:06 9493k
 80  189M   80  151M    0     0  7659k      0  0:00:25  0:00:20  0:00:05 9560k
 84  189M   84  160M    0     0  7744k      0  0:00:25  0:00:21  0:00:04 9526k
 90  189M   90  170M    0     0  7840k      0  0:00:24  0:00:22  0:00:02 9636k
 95  189M   95  180M    0     0  7933k      0  0:00:24  0:00:23  0:00:01 9781k
100  189M  100  189M    0     0  8007k      0  0:00:24  0:00:24 --:--:-- 9787k
## There are 34748555 observations in 2017
##   % 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  109M    0 27253    0     0  38862      0  0:49:15 --:--:--  0:49:15 38821
  0  109M    0  815k    0     0   483k      0  0:03:52  0:00:01  0:03:51  483k
  2  109M    2 3073k    0     0  1143k      0  0:01:38  0:00:02  0:01:36 1143k
  5  109M    5 6518k    0     0  1772k      0  0:01:03  0:00:03  0:01:00 1772k
 10  109M   10 11.3M    0     0  2478k      0  0:00:45  0:00:04  0:00:41 2478k
 16  109M   16 18.3M    0     0  3319k      0  0:00:33  0:00:05  0:00:28 3781k
 24  109M   24 27.0M    0     0  4152k      0  0:00:27  0:00:06  0:00:21 5394k
 33  109M   33 36.2M    0     0  4835k      0  0:00:23  0:00:07  0:00:16 6821k
 41  109M   41 45.7M    0     0  5399k      0  0:00:20  0:00:08  0:00:12 8068k
 50  109M   50 54.8M    0     0  5808k      0  0:00:19  0:00:09  0:00:10 8941k
 58  109M   58 64.3M    0     0  6175k      0  0:00:18  0:00:10  0:00:08 9415k
 67  109M   67 73.4M    0     0  6446k      0  0:00:17  0:00:11  0:00:06 9507k
 75  109M   75 82.4M    0     0  6663k      0  0:00:16  0:00:12  0:00:04 9474k
 84  109M   84 92.0M    0     0  6897k      0  0:00:16  0:00:13  0:00:03 9496k
 92  109M   92  101M    0     0  7106k      0  0:00:15  0:00:14  0:00:01 9618k
 99  109M   99  109M    0     0  7149k      0  0:00:15  0:00:15 --:--:-- 9226k
100  109M  100  109M    0     0  7153k      0  0:00:15  0:00:15 --:--:-- 9211k
## There are 20229121 observations in 2018
```

(b)

```
## subset to the station corresponding to Death Valley, to TMAX, and
## to March, and put all the data into a single file 'DVtmaxMarch'

## find the station ID for Death Valley
curl -o stations.txt https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/ghcnd-stations.txt
dv=$(grep "DEATH VALLEY" stations.txt | head -1 | cut -d' ' -f1)
rm stations.txt

## subset the data and put it into a file
for ((i=5;i<=8;i++))
```

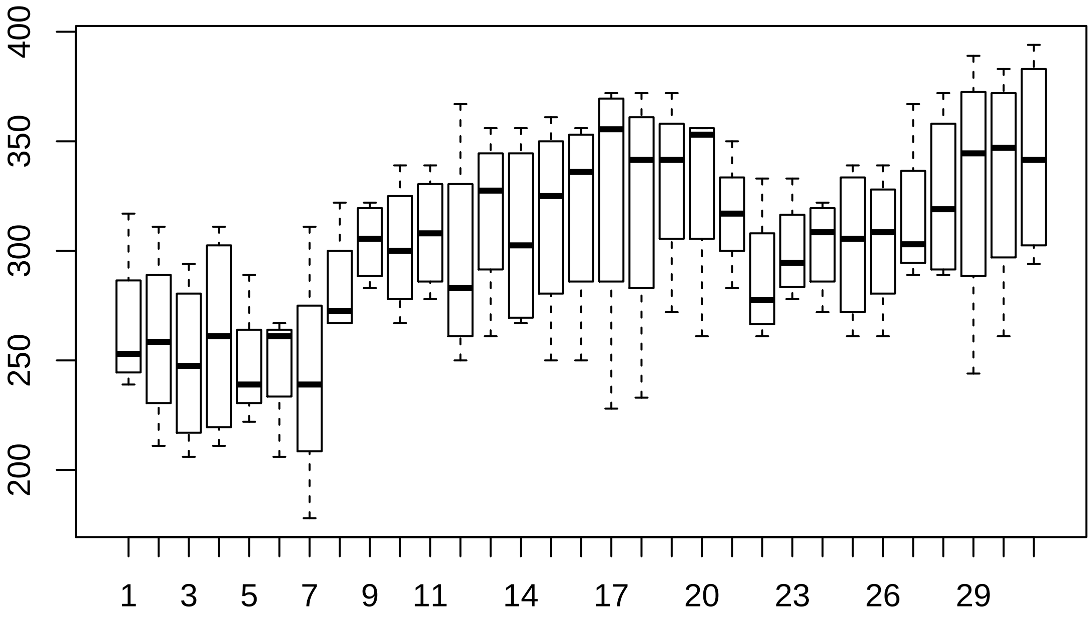
```
do
grep $dv 201${i}.csv | grep TMAX | grep 201${i}03 >> DVtmaxMarch
rm 201$i.csv
done
```

```
##   % 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
  1 8959k    1  162k    0     0   137k      0  0:01:05  0:00:01  0:01:04  137k
 14 8959k   14 1334k    0     0   614k      0  0:00:14  0:00:02  0:00:12  614k
 43 8959k   43 3865k    0     0  1221k      0  0:00:07  0:00:03  0:00:04 1221k
 88 8959k   88 7888k    0     0  1901k      0  0:00:04  0:00:04 --:--:-- 1900k
100 8959k  100 8959k    0     0  2056k      0  0:00:04  0:00:04 --:--:-- 2158k
```

(c)

```
## make a single plot of side-by-side boxplots containing TMAX on
## each day using 'DVtmaxMarch'

data <- read.csv('DVtmaxMarch', header = FALSE)
for (j in 5:8){
  for (i in 1:31){
  data$V2 <- data$V2 - (data$V2 == 20100300+j*10000+i)*(20100300+j*10000)
  }
} # categorize the data by each day in March
boxplot(V4~V2, data = data)
```



(d)

```
## generate a file including the weather data of interest.
## usage: get_weather "location" "weather variable" "year1 year2..." "month"
## use get_weather "-h" to get more help information

function get_weather(){
if [ ${1}  == "-h" ]; then # give help information
  echo -e "This function will generate a file including the weather data of interest.\n
It includes four arguments: location, weather variable, years and month of interest.\n
if location matches zero or more than one stations ID, you'll get a warning.\n
usage: get_weather \"location\" \"weather variable\" \"year1 year2...\" \"month\"\n
example: get_weather \"VALLEYVIEW AGDM\" \"TMAX\" \"2017 2018\" \"05\"\n"

elif [ $# != "4" ]; then # give a warning when the number of arguments is wrong
  echo "Warning: wrong number of arguments!"
else
  curl -o stations.txt https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/ghcnd-stations.txt
  ID=$(grep ${1} stations.txt | cut -d' ' -f1)
  exist=$(grep ${1} stations.txt | uniq | wc -l)
  rm stations.txt
  if [ $exist != '1' ]; then
  echo "Warning: can't find a single station!" # give a warning when there are no or one more matches
  else
    for i in $3
    do
    curl -o $i.csv.gz https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/by_year/$i.csv.gz
    gzip -d $i.csv.gz
    grep $ID ${i}.csv | grep $2 | grep ${i}${4} >> weather_data
    rm $i.csv # remove the raw downloaded data files
    done
  fi
fi
}

## some test examples
get_weather -h
get_weather "PRAHA-KLEMENTINUM" "TMAX" "1817 1815"
get_weather "PRAHA-KLEMENTINUM" "TMAX" "1817 1815" "05"
head -n 10 weather_data
```

```
## This function will generate a file including the weather data of interest.
##
## It includes four arguments: location, weather variable, years and month of interest.
##
## if location matches zero or more than one stations ID, you'll get a warning.
##
## usage: get_weather "location" "weather variable" "year1 year2..." "month"
##
## example: get_weather "VALLEYVIEW AGDM" "TMAX" "2017 2018" "05"
##
## Warning: wrong number of arguments!
##   % 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
  0 8959k    0 91263    0     0  84198      0  0:01:48  0:00:01  0:01:47 84190
  7 8959k    7  675k    0     0   324k      0  0:00:27  0:00:02  0:00:25  324k
 26 8959k   26 2346k    0     0   759k      0  0:00:11  0:00:03  0:00:08  759k
 56 8959k   56 5026k    0     0  1233k      0  0:00:07  0:00:04  0:00:03 1232k
 98 8959k   98 8831k    0     0  1732k      0  0:00:05  0:00:05 --:--:-- 1770k
100 8959k  100 8959k    0     0  1745k      0  0:00:05  0:00:05 --:--:-- 2191k
##   % 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 11885  100 11885    0     0  20336      0 --:--:-- --:--:-- --:--:-- 20351
##   % 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 12042  100 12042    0     0  25858      0 --:--:-- --:--:-- --:--:-- 25841
## EZE00100082,18170501,TMAX,148,,,E,
## EZE00100082,18170502,TMAX,172,,,E,
## EZE00100082,18170503,TMAX,186,,,E,
## EZE00100082,18170504,TMAX,132,,,E,
## EZE00100082,18170505,TMAX,132,,,E,
## EZE00100082,18170506,TMAX,167,,,E,
## EZE00100082,18170507,TMAX,157,,,E,
## EZE00100082,18170508,TMAX,186,,,E,
## EZE00100082,18170509,TMAX,214,,,E,
## EZE00100082,18170510,TMAX,181,,,E,
```

4

For this question, I used bash to download all the files ending in .txt from the National Climate Data Center website.

```
## automatically download all the files ending in .txt from
## https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/.

curl https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/ > html
cat html | grep txt | cut -d'"' -f8 > txt_name # extract the names of all .txt files in 'txt_name'
rm html

count=$(cat txt_name | wc -l)
for ((i=1;i<=count;i++)) # use a for loop to download the .txt files
do
name=$(head -$i txt_name | tail -1)
curl https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/$name > $name
echo "downloading $name" #provide a status message telling the name of the file when downloading
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
100  6068  100  6068    0     0  12341      0 --:--:-- --:--:-- --:--:-- 12358
##   % 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   6941      0 --:--:-- --:--:-- --:--:--  6937
## 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 26.6M    0 14229    0     0  23610      0  0:19:44 --:--:--  0:19:44 23597
  1 26.6M    1  477k    0     0   296k      0  0:01:32  0:00:01  0:01:31  295k
  7 26.6M    7 2008k    0     0   766k      0  0:00:35  0:00:02  0:00:33  766k
 16 26.6M   16 4415k    0     0  1222k      0  0:00:22  0:00:03  0:00:19 1222k
 29 26.6M   29 8032k    0     0  1740k      0  0:00:15  0:00:04  0:00:11 1740k
 47 26.6M   47 12.6M    0     0  2318k      0  0:00:11  0:00:05  0:00:06 2595k
 70 26.6M   70 18.7M    0     0  2908k      0  0:00:09  0:00:06  0:00:03 3755k
100 26.6M  100 26.6M    0     0  3631k      0  0:00:07  0:00:07 --:--:-- 5163k
## 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
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100  1086  100  1086    0     0   2264      0 --:--:-- --:--:-- --:--:--  2262
## 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 8959k    0 14231    0     0  23925      0  0:06:23 --:--:--  0:06:23 23917
  3 8959k    3  348k    0     0   220k      0  0:00:40  0:00:01  0:00:39  220k
 19 8959k   19 1770k    0     0   684k      0  0:00:13  0:00:02  0:00:11  684k
 45 8959k   45 4113k    0     0  1153k      0  0:00:07  0:00:03  0:00:04 1153k
 84 8959k   84 7574k    0     0  1665k      0  0:00:05  0:00:04  0:00:01 1664k
100 8959k  100 8959k    0     0  1840k      0  0:00:04  0:00:04 --:--:-- 2094k
## 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    578      0 --:--:-- --:--:-- --:--:--   579
## 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
  3 3707k    3  118k    0     0    98k      0  0:00:37  0:00:01  0:00:36   98k
 27 3707k   27 1016k    0     0   461k      0  0:00:08  0:00:02  0:00:06  461k
 76 3707k   76 2822k    0     0   878k      0  0:00:04  0:00:03  0:00:01  878k
100 3707k  100 3707k    0     0  1045k      0  0:00:03  0:00:03 --:--:-- 1045k
## 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
 84 26498   84 22235    0     0  33256      0 --:--:-- --:--:-- --:--:-- 33236
100 26498  100 26498    0     0  39595      0 --:--:-- --:--:-- --:--:-- 39549
## 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
100 31860  100 31860    0     0  46356      0 --:--:-- --:--:-- --:--:-- 46375
## downloading status.txt
```

5(b)

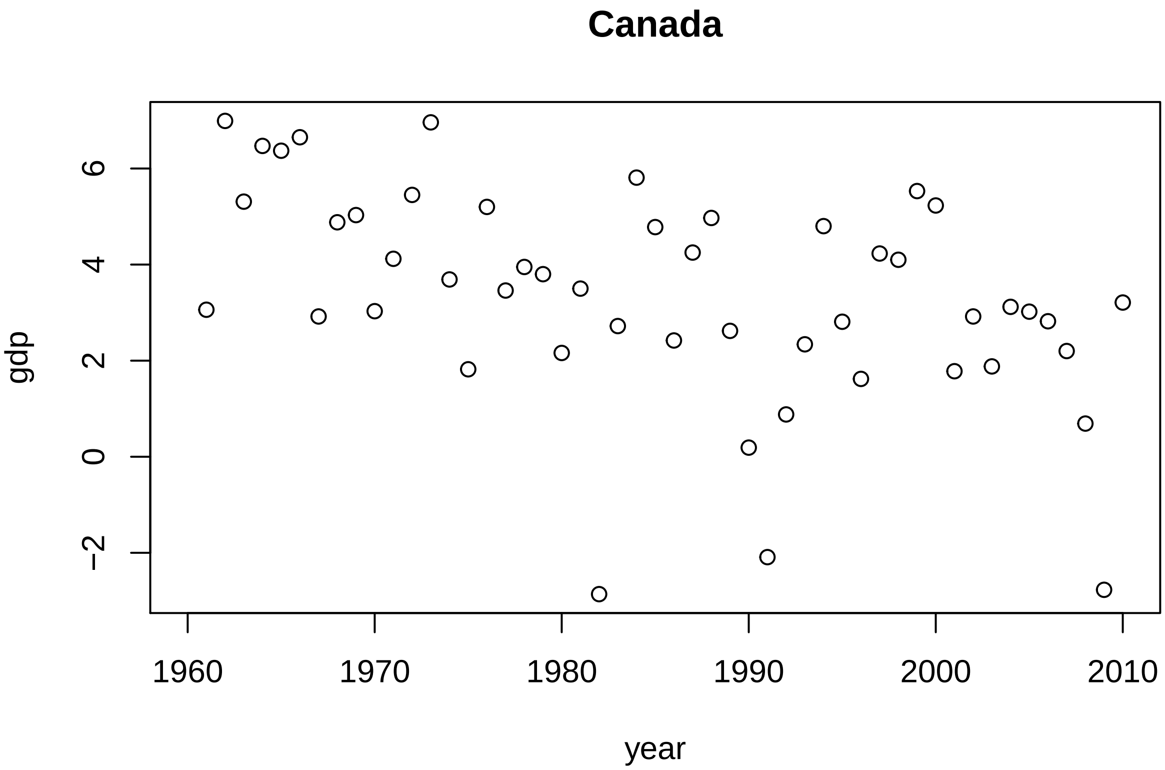
This package makes it possible to call Python from R and vice versa, and translate between R and Python objects.

```r
## read cpds.csv into R
dataR <- read.csv("cpds.csv", stringsAsFactors = FALSE)
```

```python
## manipulate the data in Python
import pandas
dataPy = r.dataR
newdata = dataPy[dataPy['country'] == "Canada"]
```

```r
## send data back to R

newdata <- py$newdata
year <- newdata[,"year"]
gdp <- newdata[,"realgdpgr"]
plot(gdp~year)
title("Canada")
```


Canada
6
4
2
0
−2
gdp
1960
1970
1980
1990
2000
2010
year