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  192M    0 22220    0     0  32029      0  1:44:53 --:--:--  1:44:53 32017
  0  192M    0  576k    0     0   345k      0  0:09:29  0:00:01  0:09:28  345k
  1  192M    1 2865k    0     0  1074k      0  0:03:03  0:00:02  0:03:01 1073k
  3  192M    3 7326k    0     0  1997k      0  0:01:38  0:00:03  0:01:35 1997k
  7  192M    7 13.9M    0     0  3063k      0  0:01:04  0:00:04  0:01:00 3063k
 11  192M   11 21.7M    0     0  3933k      0  0:00:50  0:00:05  0:00:45 4477k
 15  192M   15 29.8M    0     0  4591k      0  0:00:42  0:00:06  0:00:36 6007k
 20  192M   20 38.4M    0     0  5141k      0  0:00:38  0:00:07  0:00:31 7310k
 24  192M   24 47.1M    0     0  5560k      0  0:00:35  0:00:08  0:00:27 8167k
 29  192M   29 55.8M    0     0  5917k      0  0:00:33  0:00:09  0:00:24 8581k
 33  192M   33 64.2M    0     0  6171k      0  0:00:31  0:00:10  0:00:21 8711k
 37  192M   37 72.9M    0     0  6405k      0  0:00:30  0:00:11  0:00:19 8823k
 42  192M   42 81.6M    0     0  6599k      0  0:00:29  0:00:12  0:00:17 8835k
 47  192M   47 90.3M    0     0  6770k      0  0:00:29  0:00:13  0:00:16 8874k
 51  192M   51 99.0M    0     0  6916k      0  0:00:28  0:00:14  0:00:14 8846k
 56  192M   56  108M    0     0  7061k      0  0:00:27  0:00:15  0:00:12 8957k
 60  192M   60  116M    0     0  7156k      0  0:00:27  0:00:16  0:00:11 8909k
 65  192M   65  125M    0     0  7295k      0  0:00:26  0:00:17  0:00:09 9056k
```

```
 70  192M   70  134M    0     0  7403k      0  0:00:26  0:00:18  0:00:08 9134k
 74  192M   74  143M    0     0  7483k      0  0:00:26  0:00:19  0:00:07 9147k
 79  192M   79  152M    0     0  7544k      0  0:00:26  0:00:20  0:00:06 9056k
 83  192M   83  159M    0     0  7557k      0  0:00:26  0:00:21  0:00:05 8895k
 86  192M   86  166M    0     0  7537k      0  0:00:26  0:00:22  0:00:04 8394k
 91  192M   91  175M    0     0  7605k      0  0:00:25  0:00:23  0:00:02 8359k
 95  192M   95  183M    0     0  7635k      0  0:00:25  0:00:24  0:00:01 8234k
100  192M  100  192M    0     0  7674k      0  0:00:25  0:00:25 --:--:-- 8218k
## 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     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
  0  192M    0  235k    0     0   197k      0  0:16:38  0:00:01  0:16:37  197k
  0  192M    0 1751k    0     0   808k      0  0:04:03  0:00:02  0:04:01  808k
  2  192M    2 4727k    0     0  1507k      0  0:02:10  0:00:03  0:02:07 1507k
  4  192M    4 9587k    0     0  2324k      0  0:01:24  0:00:04  0:01:20 2324k
  8  192M    8 15.9M    0     0  3185k      0  0:01:01  0:00:05  0:00:56 3263k
 12  192M   12 23.0M    0     0  3865k      0  0:00:50  0:00:06  0:00:44 4755k
 15  192M   15 30.4M    0     0  4378k      0  0:00:44  0:00:07  0:00:37 5939k
 19  192M   19 37.6M    0     0  4754k      0  0:00:41  0:00:08  0:00:33 6800k
 24  192M   24 47.2M    0     0  5306k      0  0:00:37  0:00:09  0:00:28 7768k
 29  192M   29 57.1M    0     0  5780k      0  0:00:34  0:00:10  0:00:24 8437k
 34  192M   34 66.1M    0     0  6089k      0  0:00:32  0:00:11  0:00:21 8808k
 39  192M   39 75.3M    0     0  6367k      0  0:00:30  0:00:12  0:00:18 9198k
 44  192M   44 85.0M    0     0  6637k      0  0:00:29  0:00:13  0:00:16 9695k
 49  192M   49 94.5M    0     0  6850k      0  0:00:28  0:00:14  0:00:14 9657k
 53  192M   53  102M    0     0  6973k      0  0:00:28  0:00:15  0:00:13 9387k
 58  192M   58  111M    0     0  7088k      0  0:00:27  0:00:16  0:00:11 9312k
 62  192M   62  119M    0     0  7121k      0  0:00:27  0:00:17  0:00:10 8944k
 65  192M   65  126M    0     0  7151k      0  0:00:27  0:00:18  0:00:09 8501k
 70  192M   70  135M    0     0  7258k      0  0:00:27  0:00:19  0:00:08 8416k
 74  192M   74  143M    0     0  7327k      0  0:00:26  0:00:20  0:00:06 8396k
 79  192M   79  152M    0     0  7386k      0  0:00:26  0:00:21  0:00:05 8348k
 83  192M   83  160M    0     0  7423k      0  0:00:26  0:00:22  0:00:04 8460k
 87  192M   87  167M    0     0  7429k      0  0:00:26  0:00:23  0:00:03 8436k
 91  192M   91  176M    0     0  7487k      0  0:00:26  0:00:24  0:00:02 8359k
 95  192M   95  183M    0     0  7490k      0  0:00:26  0:00:25  0:00:01 8149k
100  192M  100  192M    0     0  7545k      0  0:00:26  0:00:26 --:--:-- 8221k
## 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  189M    0 70148    0     0  77778      0  0:42:31 --:--:--  0:42:31 77769
  0  189M    0  990k    0     0   527k      0  0:06:07  0:00:01  0:06:06  527k
  1  189M    1 3654k    0     0  1268k      0  0:02:32  0:00:02  0:02:30 1268k
  4  189M    4 8138k    0     0  2097k      0  0:01:32  0:00:03  0:01:29 2097k
  7  189M    7 14.3M    0     0  3022k      0  0:01:04  0:00:04  0:01:00 3022k
 11  189M   11 22.4M    0     0  3914k      0  0:00:49  0:00:05  0:00:44 4609k
 16  189M   16 31.3M    0     0  4673k      0  0:00:41  0:00:06  0:00:35 6232k
 21  189M   21 40.3M    0     0  5239k      0  0:00:36  0:00:07  0:00:29 7525k
 25  189M   25 49.0M    0     0  5656k      0  0:00:34  0:00:08  0:00:26 8418k
```

```
 30  189M   30 57.7M    0     0  5987k      0  0:00:32  0:00:09  0:00:23 8881k
 34  189M   34 65.9M    0     0  6210k      0  0:00:31  0:00:10  0:00:21 8909k
 39  189M   39 73.8M    0     0  6367k      0  0:00:30  0:00:11  0:00:19 8698k
 43  189M   43 81.6M    0     0  6490k      0  0:00:29  0:00:12  0:00:17 8463k
 47  189M   47 89.6M    0     0  6613k      0  0:00:29  0:00:13  0:00:16 8313k
 51  189M   51 97.9M    0     0  6742k      0  0:00:28  0:00:14  0:00:14 8231k
 56  189M   56  106M    0     0  6895k      0  0:00:28  0:00:15  0:00:13 8386k
 60  189M   60  114M    0     0  6955k      0  0:00:27  0:00:16  0:00:11 8352k
 65  189M   65  123M    0     0  7058k      0  0:00:27  0:00:17  0:00:10 8520k
 69  189M   69  132M    0     0  7176k      0  0:00:27  0:00:18  0:00:09 8737k
 74  189M   74  140M    0     0  7262k      0  0:00:26  0:00:19  0:00:07 8812k
 78  189M   78  149M    0     0  7327k      0  0:00:26  0:00:20  0:00:06 8697k
 83  189M   83  157M    0     0  7353k      0  0:00:26  0:00:21  0:00:05 8697k
 87  189M   87  165M    0     0  7388k      0  0:00:26  0:00:22  0:00:04 8568k
 91  189M   91  173M    0     0  7420k      0  0:00:26  0:00:23  0:00:03 8343k
 96  189M   96  181M    0     0  7486k      0  0:00:25  0:00:24  0:00:01 8375k
100  189M  100  189M    0     0  7540k      0  0:00:25  0:00:25 --:--:-- 8468k
## 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     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
  0  109M    0  157k    0     0   122k      0  0:15:14  0:00:01  0:15:13  122k
  1  109M    1 1587k    0     0   687k      0  0:02:42  0:00:02  0:02:40  687k
  3  109M    3 4407k    0     0  1325k      0  0:01:24  0:00:03  0:01:21 1325k
  7  109M    7 8258k    0     0  1926k      0  0:00:57  0:00:04  0:00:53 1926k
 11  109M   11 12.9M    0     0  2494k      0  0:00:44  0:00:05  0:00:39 2689k
 18  109M   18 19.8M    0     0  3239k      0  0:00:34  0:00:06  0:00:28 4043k
 25  109M   25 27.5M    0     0  3874k      0  0:00:28  0:00:07  0:00:21 5352k
 31  109M   31 34.8M    0     0  4303k      0  0:00:25  0:00:08  0:00:17 6296k
 38  109M   38 42.4M    0     0  4681k      0  0:00:23  0:00:09  0:00:14 7042k
 46  109M   46 50.4M    0     0  5023k      0  0:00:22  0:00:10  0:00:12 7703k
 53  109M   53 58.6M    0     0  5316k      0  0:00:20  0:00:11  0:00:09 7924k
 60  109M   60 66.4M    0     0  5536k      0  0:00:20  0:00:12  0:00:08 7956k
 66  109M   66 72.5M    0     0  5591k      0  0:00:19  0:00:13  0:00:06 7731k
 72  109M   72 79.4M    0     0  5692k      0  0:00:19  0:00:14  0:00:05 7574k
 79  109M   79 86.3M    0     0  5785k      0  0:00:19  0:00:15  0:00:04 7352k
 85  109M   85 92.9M    0     0  5841k      0  0:00:19  0:00:16  0:00:03 7026k
 91  109M   91 99.2M    0     0  5872k      0  0:00:19  0:00:17  0:00:02 6697k
 97  109M   97  106M    0     0  5961k      0  0:00:18  0:00:18 --:--:-- 6941k
100  109M  100  109M    0     0  5997k      0  0:00:18  0:00:18 --:--:-- 7003k
## There are 20127059 observations in 2018
```

(b)

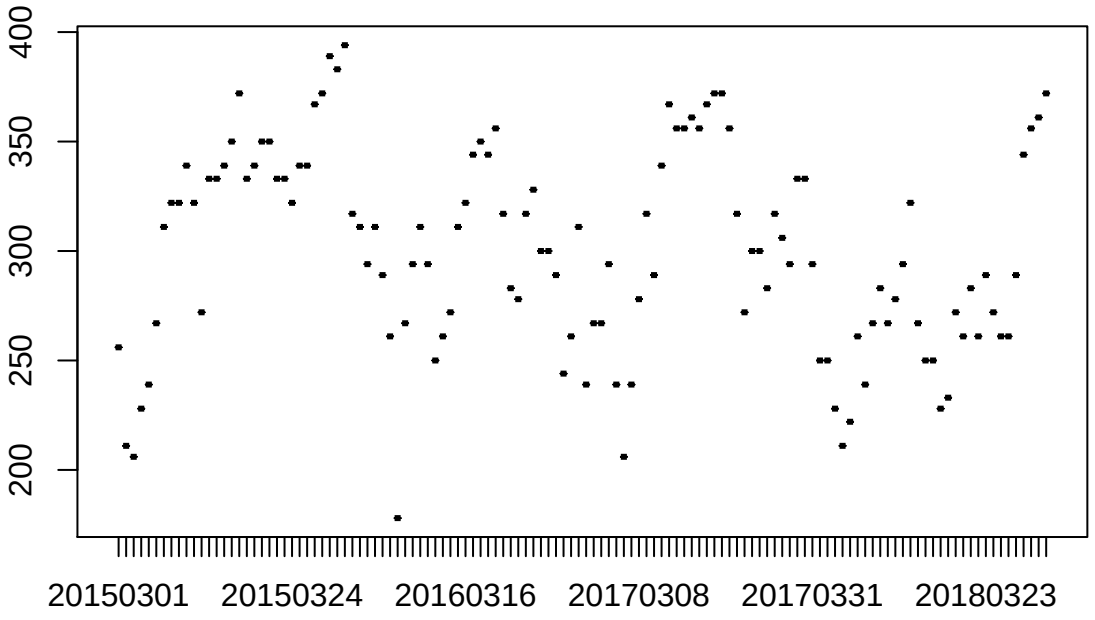
```
## subset to the station corresponding to Death Valley, to TMAX, and
## to March, and put all the data into a single file 'DVtmaxMarch'

## 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 8959k    0 14231    0     0  24207      0  0:06:19 --:--:--  0:06:19 24202
  6 8959k    6  584k    0     0   366k      0  0:00:24  0:00:01  0:00:23  366k
 29 8959k   29 2599k    0     0  1000k      0  0:00:08  0:00:02  0:00:06  999k
 72 8959k   72 6537k    0     0  1821k      0  0:00:04  0:00:03  0:00:01 1821k
100 8959k  100 8959k    0     0  2226k      0  0:00:04  0:00:04 --:--:-- 2226k
```

(c)

```
## make a single plot of side-by-side boxplots using 'DVtmaxMarch'

data <- read.csv('DVtmaxMarch', header = FALSE)
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
  4 8959k    4  404k    0     0   295k      0  0:00:30  0:00:01  0:00:29  295k
```

```
 24 8959k   24 2224k    0     0   943k      0  0:00:09  0:00:02  0:00:07  943k
 58 8959k   58 5248k    0     0  1574k      0  0:00:05  0:00:03  0:00:02 1574k
100 8959k  100 8959k    0     0  2158k      0  0:00:04  0:00:04 --:--:-- 2158k
##   % 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
 65 11885   65  7747    0     0   7171      0  0:00:01  0:00:01 --:--:--  7166
100 11885  100 11885    0     0  10984      0  0:00:01  0:00:01 --:--:-- 10984
##   % 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 12042  100 12042    0     0  25431      0 --:--:-- --:--:-- --:--:-- 25458
## 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
  0  6068    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100  6068  100  6068    0     0  12528      0 --:--:-- --:--:-- --:--:-- 12511
##   % 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  3670  100  3670    0     0   7730      0 --:--:-- --:--:-- --:--:--  7742
## 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     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
  1 26.6M    1  357k    0     0   258k      0  0:01:45  0:00:01  0:01:44  258k
  8 26.6M    8 2287k    0     0   959k      0  0:00:28  0:00:02  0:00:26  959k
 20 26.6M   20 5607k    0     0  1652k      0  0:00:16  0:00:03  0:00:13 1652k
 41 26.6M   41 10.9M    0     0  2570k      0  0:00:10  0:00:04  0:00:06 2569k
 70 26.6M   70 18.9M    0     0  3599k      0  0:00:07  0:00:05  0:00:02 3883k
100 26.6M  100 26.6M    0     0  4403k      0  0:00:06  0:00:06 --:--:-- 5593k
## 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   2295      0 --:--:-- --:--:-- --:--:--  2291
## 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 30159    0     0  43867      0  0:03:29 --:--:--  0:03:29 43835
  9 8959k    9  857k    0     0   514k      0  0:00:17  0:00:01  0:00:16  514k
 35 8959k   35 3216k    0     0  1205k      0  0:00:07  0:00:02  0:00:05 1205k
 80 8959k   80 7177k    0     0  1957k      0  0:00:04  0:00:03  0:00:01 1957k
100 8959k  100 8959k    0     0  2215k      0  0:00:04  0:00:04 --:--:-- 2215k
## 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    555      0 --:--:-- --:--:-- --:--:--   555
## 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
  2 3707k    2  107k    0     0  93937      0  0:00:40  0:00:01  0:00:39 93912
 23 3707k   23  865k    0     0   412k      0  0:00:08  0:00:02  0:00:06  412k
 82 3707k   82 3068k    0     0   991k      0  0:00:03  0:00:03 --:--:--  991k
100 3707k  100 3707k    0     0  1125k      0  0:00:03  0:00:03 --:--:-- 1125k
## 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  44014      0 --:--:-- --:--:-- --:--:-- 44089
```

```
## 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  41209      0 --:--:-- --:--:-- --:--:-- 41162
## 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")
```