

PS1

Shubei Wang

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 substed 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	396k	0	0	317k	0	0:10:20 0:00:01 0:10:19 316k
2	192M	2	4974k	0	0	2215k	0	0:01:28 0:00:02 0:01:26 2214k
5	192M	5	11.1M	0	0	3539k	0	0:00:55 0:00:03 0:00:52 3539k
9	192M	9	17.8M	0	0	4301k	0	0:00:45 0:00:04 0:00:41 4300k
12	192M	12	24.4M	0	0	4777k	0	0:00:41 0:00:05 0:00:36 5027k
16	192M	16	31.2M	0	0	5129k	0	0:00:38 0:00:06 0:00:32 6334k
19	192M	19	38.1M	0	0	5398k	0	0:00:36 0:00:07 0:00:29 6828k
21	192M	21	40.9M	0	0	4857k	0	0:00:40 0:00:08 0:00:32 5650k
21	192M	21	40.9M	0	0	4351k	0	0:00:45 0:00:09 0:00:36 4391k
21	192M	21	40.9M	0	0	3940k	0	0:00:49 0:00:10 0:00:39 3127k
21	192M	21	40.9M	0	0	3600k	0	0:00:54 0:00:11 0:00:43 1834k
21	192M	21	40.9M	0	0	3314k	0	0:00:59 0:00:12 0:00:47 523k
21	192M	21	40.9M	0	0	3164k	0	0:01:02 0:00:13 0:00:49 5192
21	192M	21	41.4M	0	0	2975k	0	0:01:06 0:00:14 0:00:52 104k
22	192M	22	42.3M	0	0	2841k	0	0:01:09 0:00:15 0:00:54 306k
22	192M	22	43.9M	0	0	2767k	0	0:01:11 0:00:16 0:00:55 668k
24	192M	24	46.7M	0	0	2775k	0	0:01:10 0:00:17 0:00:53 1294k

26	192M	26	51.4M	0	0	2888k	0	0:01:08	0:00:18	0:00:50	2152k
30	192M	30	59.1M	0	0	3146k	0	0:01:02	0:00:19	0:00:43	3632k
35	192M	35	69.0M	0	0	3494k	0	0:00:56	0:00:20	0:00:36	5490k
38	192M	38	73.6M	0	0	3468k	0	0:00:56	0:00:21	0:00:35	5542k
38	192M	38	73.6M	0	0	3315k	0	0:00:59	0:00:22	0:00:37	5004k
38	192M	38	73.6M	0	0	3236k	0	0:01:00	0:00:23	0:00:37	4487k
38	192M	38	74.4M	0	0	3136k	0	0:01:02	0:00:24	0:00:38	3099k
42	192M	42	81.1M	0	0	3293k	0	0:00:59	0:00:25	0:00:34	2482k
47	192M	47	91.3M	0	0	3566k	0	0:00:55	0:00:26	0:00:29	4039k
53	192M	53	102M	0	0	3836k	0	0:00:51	0:00:27	0:00:24	6481k
58	192M	58	112M	0	0	4088k	0	0:00:48	0:00:28	0:00:20	8119k
64	192M	64	123M	0	0	4319k	0	0:00:45	0:00:29	0:00:16	9.8M
69	192M	69	134M	0	0	4539k	0	0:00:43	0:00:30	0:00:13	10.5M
75	192M	75	144M	0	0	4746k	0	0:00:41	0:00:31	0:00:10	10.6M
80	192M	80	155M	0	0	4944k	0	0:00:39	0:00:32	0:00:07	10.7M
86	192M	86	166M	0	0	5121k	0	0:00:38	0:00:33	0:00:05	10.7M
91	192M	91	176M	0	0	5274k	0	0:00:37	0:00:34	0:00:03	10.6M
97	192M	97	186M	0	0	5425k	0	0:00:36	0:00:35	0:00:01	10.5M
100	192M	100	192M	0	0	5503k	0	0:00:35	0:00:35	--:--:--	10.4M

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	300k	0	0	263k	0	0:12:28	0:00:01	0:12:27	262k
2	192M	2	4721k	0	0	2268k	0	0:01:26	0:00:02	0:01:24	2267k
5	192M	5	10.7M	0	0	3592k	0	0:00:54	0:00:03	0:00:51	3591k
9	192M	9	19.2M	0	0	4822k	0	0:00:40	0:00:04	0:00:36	4821k
14	192M	14	27.8M	0	0	5619k	0	0:00:35	0:00:05	0:00:30	5731k
19	192M	19	36.9M	0	0	6222k	0	0:00:31	0:00:06	0:00:25	7598k
23	192M	23	46.0M	0	0	6666k	0	0:00:29	0:00:07	0:00:22	8498k
28	192M	28	55.2M	0	0	6996k	0	0:00:28	0:00:08	0:00:20	9091k
33	192M	33	64.4M	0	0	7272k	0	0:00:27	0:00:09	0:00:18	9271k
38	192M	38	74.2M	0	0	7541k	0	0:00:26	0:00:10	0:00:16	9494k
41	192M	41	80.0M	0	0	7395k	0	0:00:26	0:00:11	0:00:15	8820k
45	192M	45	86.8M	0	0	7359k	0	0:00:26	0:00:12	0:00:14	8340k
48	192M	48	92.9M	0	0	7261k	0	0:00:27	0:00:13	0:00:14	7686k
51	192M	51	99.1M	0	0	7194k	0	0:00:27	0:00:14	0:00:13	7054k
54	192M	54	105M	0	0	7145k	0	0:00:27	0:00:15	0:00:12	6353k
58	192M	58	112M	0	0	7174k	0	0:00:27	0:00:16	0:00:11	6686k
62	192M	62	120M	0	0	7220k	0	0:00:27	0:00:17	0:00:10	6884k
67	192M	67	128M	0	0	7299k	0	0:00:26	0:00:18	0:00:08	7398k
71	192M	71	137M	0	0	7379k	0	0:00:26	0:00:19	0:00:07	7902k
76	192M	76	147M	0	0	7519k	0	0:00:26	0:00:20	0:00:06	8654k
82	192M	82	158M	0	0	7686k	0	0:00:25	0:00:21	0:00:04	9335k
87	192M	87	168M	0	0	7835k	0	0:00:25	0:00:22	0:00:03	9934k
93	192M	93	179M	0	0	7975k	0	0:00:24	0:00:23	0:00:01	10.1M
99	192M	99	190M	0	0	8108k	0	0:00:24	0:00:24	--:--:--	10.6M
100	192M	100	192M	0	0	8121k	0	0:00:24	0:00:24	--:--:--	10.7M

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      0      0      0      0      0  --:--:-- --:--:-- --:--:--      0
0 189M      0 378k      0      0 254k      0 0:12:42 0:00:01 0:12:41 253k
0 189M      0 1112k      0      0 464k      0 0:06:56 0:00:02 0:06:54 464k
1 189M      1 2565k      0      0 743k      0 0:04:20 0:00:03 0:04:17 743k
2 189M      2 4839k      0      0 1101k      0 0:02:55 0:00:04 0:02:51 1101k
4 189M      4 8760k      0      0 1624k      0 0:01:59 0:00:05 0:01:54 1765k
7 189M      7 14.9M      0      0 2387k      0 0:01:21 0:00:06 0:01:15 3033k
12 189M     12 23.7M      0      0 3284k      0 0:00:58 0:00:07 0:00:51 4632k
17 189M     17 32.9M      0      0 4022k      0 0:00:48 0:00:08 0:00:40 6311k
22 189M     22 42.2M      0      0 4599k      0 0:00:42 0:00:09 0:00:33 7675k
27 189M     27 51.5M      0      0 5076k      0 0:00:38 0:00:10 0:00:28 8801k
32 189M     32 61.1M      0      0 5492k      0 0:00:35 0:00:11 0:00:24 9472k
37 189M     37 70.4M      0      0 5817k      0 0:00:33 0:00:12 0:00:21 9573k
42 189M     42 79.7M      0      0 6100k      0 0:00:31 0:00:13 0:00:18 9590k
47 189M     47 90.0M      0      0 6406k      0 0:00:30 0:00:14 0:00:16 9801k
53 189M     53 100M      0      0 6676k      0 0:00:29 0:00:15 0:00:14 9.7M
58 189M     58 111M      0      0 6947k      0 0:00:27 0:00:16 0:00:11 10.0M
64 189M     64 121M      0      0 7169k      0 0:00:27 0:00:17 0:00:10 10.2M
70 189M     70 132M      0      0 7378k      0 0:00:26 0:00:18 0:00:08 10.5M
75 189M     75 143M      0      0 7562k      0 0:00:25 0:00:19 0:00:06 10.6M
79 189M     79 150M      0      0 7558k      0 0:00:25 0:00:20 0:00:05 10.0M
83 189M     83 157M      0      0 7523k      0 0:00:25 0:00:21 0:00:04 9414k
87 189M     87 165M      0      0 7549k      0 0:00:25 0:00:22 0:00:03 8871k
91 189M     91 173M      0      0 7587k      0 0:00:25 0:00:23 0:00:02 8354k
96 189M     96 181M      0      0 7628k      0 0:00:25 0:00:24 0:00:01 7882k
99 189M     99 187M      0      0 7572k      0 0:00:25 0:00:25 --:--:-- 7629k
100 189M    100 189M      0      0 7586k      0 0:00:25 0:00:25 --:--:-- 7908k
## 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 101k      0      0 116k      0 0:16:08 --:--:-- 0:16:08 116k
3 109M      3 4413k      0      0 2373k      0 0:00:47 0:00:01 0:00:46 2373k
13 109M     13 14.3M      0      0 5204k      0 0:00:21 0:00:02 0:00:19 5203k
22 109M     22 24.8M      0      0 6663k      0 0:00:16 0:00:03 0:00:13 6661k
32 109M     32 35.5M      0      0 7549k      0 0:00:14 0:00:04 0:00:10 7549k
40 109M     40 44.6M      0      0 7842k      0 0:00:14 0:00:05 0:00:09 9199k
50 109M     50 55.3M      0      0 8308k      0 0:00:13 0:00:06 0:00:07 10.2M
60 109M     60 66.0M      0      0 8641k      0 0:00:13 0:00:07 0:00:06 10.3M
68 109M     68 75.5M      0      0 8764k      0 0:00:12 0:00:08 0:00:04 10.1M
78 109M     78 85.6M      0      0 8934k      0 0:00:12 0:00:09 0:00:03 10.0M
87 109M     87 96.6M      0      0 9144k      0 0:00:12 0:00:10 0:00:02 10.4M
97 109M     97 107M      0      0 9318k      0 0:00:12 0:00:11 0:00:01 10.4M
100 109M    100 109M      0      0 9356k      0 0:00:12 0:00:12 --:--:-- 10.4M
## There are 20283714 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

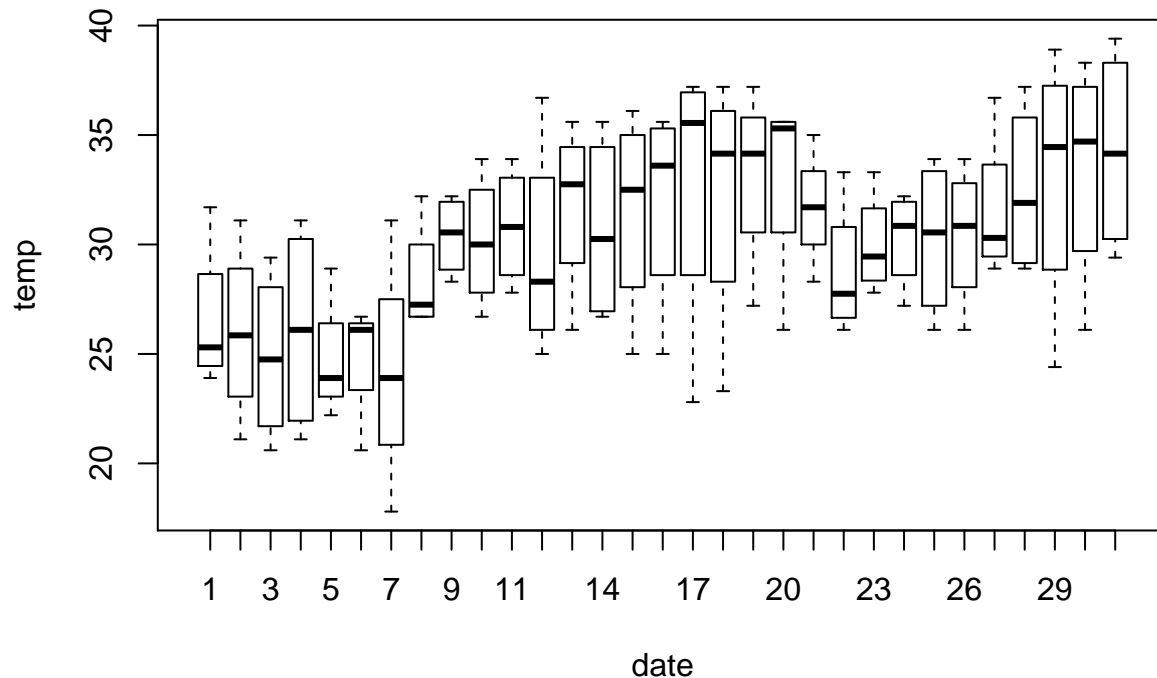
## subset the data and put it into a file
for ((i=5;i<=8;i++))
do
grep $dv 201${i}.csv | grep TMAX | grep 201${i}03 >> DVtmaxMarch
rm 201${i}.csv
done
```

##	% Total	% Received	% Xferd	Average Speed	Time	Time	Time	Current
##				Dload Upload	Total	Spent	Left	Speed
##								
0	0	0	0	0	0	--:--:--	--:--:--	0
0	8959k	0	23751	0	0	0:04:00	--:--:--	0:04:00 38062
23	8959k	23	2101k	0	0	0:00:06	0:00:01	0:00:05 1317k
100	8959k	100	8959k	0	0	4029k	0:00:02	0:00:02 --:--:-- 4030k

(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(V4/10~V2, data = data) # divide the temp by 10 according to
# ftp://ftp.ncdc.noaa.gov/pub/data/ghcn/daily/readme.txt
title(xlab = "date", ylab = "temp")
```



(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
}
```

```

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 17127    0      0 11150    0 0:13:42 0:00:01 0:13:41 11150
0 8959k    0 25127    0      0 11957    0 0:12:47 0:00:02 0:12:45 11953
1 8959k    1 110k     0      0 36848    0 0:04:08 0:00:03 0:04:05 36849
3 8959k    3 282k     0      0 69933    0 0:02:11 0:00:04 0:02:07 69921
5 8959k    5 462k     0      0 92658    0 0:01:39 0:00:05 0:01:34 94834
7 8959k    7 688k     0      0 113k     0 0:01:19 0:00:06 0:01:13 147k
11 8959k   11 1008k    0      0 143k     0 0:01:02 0:00:07 0:00:55 198k
17 8959k   17 1610k    0      0 198k     0 0:00:45 0:00:08 0:00:37 297k
28 8959k   28 2516k    0      0 277k     0 0:00:32 0:00:09 0:00:23 453k
45 8959k   45 4094k    0      0 407k     0 0:00:21 0:00:10 0:00:11 734k
72 8959k   72 6493k    0      0 587k     0 0:00:15 0:00:11 0:00:04 1166k
100 8959k  100 8959k    0      0 757k     0 0:00:11 0:00:11 --:--:-- 1666k
## % 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 11885  100 11885    0      0 20934    0  --:~:~:~  --:~:~:~  --:~:~:~ 20924
## % 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 12042    0      0      0      0      0      0  --:~:~:~  --:~:~:~  --:~:~:~      0
100 12042  100 12042    0      0 25572    0  --:~:~:~  --:~:~:~  --:~:~:~ 25566
## 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 -n $i txt_name | tail -n 1)
curl https://www1.ncdc.noaa.gov/pub/data/ghcn/daily/$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 13638 0 --:--:-- --:--:-- --:--:-- 13666
## % 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 8678 0 --:--:-- --:--:-- --:--:-- 8696
## 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
0 26.6M 0 224k 0 0 217k 0 0:02:05 0:00:01 0:02:04 217k
24 26.6M 24 6557k 0 0 3254k 0 0:00:08 0:00:02 0:00:06 3254k
58 26.6M 58 15.6M 0 0 5301k 0 0:00:05 0:00:03 0:00:02 5300k
98 26.6M 98 26.3M 0 0 6700k 0 0:00:04 0:00:04 --:~:~ 6699k
100 26.6M 100 26.6M 0 0 6705k 0 0:00:04 0:00:04 --:~:~ 6862k
## 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 2558 0 --:--:-- --:--:-- --:--:-- 2561
## 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 7751 0 0 15371 0 0:09:56 --:--:-- 0:09:56 15348
18 8959k 18 1620k 0 0 1098k 0 0:00:08 0:00:01 0:00:07 1097k
100 8959k 100 8959k 0 0 4093k 0 0:00:02 0:00:02 --:--:-- 4094k
## 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
0 0 0 0 0 0 0 0 --:--:-- --:~:~:~ --:~:~:~ 0
100 270 100 270 0 0 648 0 --:~:~:~ --:~:~:~ --:~:~:~ 647
## 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
2 3707k 2 97127 0 0 110k 0 0:00:33 --:~:~:~ 0:00:33 110k
100 3707k 100 3707k 0 0 2102k 0 0:00:01 0:00:01 --:~:~:~ 2102k
## 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
0 0 0 0 0 0 0 0 --:~:~:~ --:~:~:~ --:~:~:~ 0
100 26498 100 26498 0 0 44112 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
24 31860 24 7756 0 0 15383 0 0:00:02 --:~:~:~ 0:00:02 15358
100 31860 100 31860 0 0 53439 0 --:~:~:~ --:~:~:~ --:~:~:~ 53366
## 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.

```

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

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

## send data back to R

```



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

