STA141Assignment 2

Weitong(Jessie) Lin

ID: #913513998

Step 1.

```
#set direction
setwd("~/Desktop/UC Davis/141/STA141 Assignment 2/NASA")
#summarise the data files
table(gsub("[0-9].*", "", list.files(pattern="*.txt")))
```

```
##
##   cloudhigh    cloudlow    cloudmid       ozone    pressure    surft
emp
##          72          72          72          72          72
 72
## temperature
##          72
```

When seeing the data file, we can find that there are 72 files for each one of 7 variables. So we need to combine them for each variable. First, I will write a function called "Cleaning_Data" which is used to deal with a single txt file:

```
#For this function, the input is a single data file that we want to run,
 the output is the cleanning data.frame
Cleaning_Data<-function(x){
  #read a ".txt"
  raw_data=readLines(x)
  ## deal with the main table
  #get rid of the first 5 lines
  data_content = raw_data[-(1:5)]
  #exact information about longtitude
  longtitude=unlist(strsplit(data_content[1], ' '))
  #remove all " "
  longtitude=longtitude[nchar(longtitude)>0]
  #convert "E","W" into +/-
  #exact all "W"/"E"
  longtitude_dir=substring(longtitude,nchar(longtitude))
  #exact all values
  longtitude_num=as.numeric(substring(longtitude,1,nchar(longtitude)-1))
  #if it's "**W", then turn into "-**", else turn into "**"
  longtitude=ifelse(longtitude_dir=='W', -longtitude_num, longtitude_nu
m)
  #split latitude and grid points
  temp1=strsplit(data_content[-(1:2)], "/")
```

```
  #exact information about latitude
  latitude=sapply(temp1, '[', 1)
  latitude_temp=unlist(strsplit(latitude, " "))
  #remove all " "
  latitude=latitude_temp[nchar(latitude_temp)>0]
  #convert N","S" into +/-
  #exact all "N"/"S"
  latitude_dir=substring(latitude,nchar(latitude))
  #exact all values
  latitude_num=as.numeric(substring(latitude,1,nchar(latitude)-1))
  #if it's "**N", then turn into "**", else turn into "-**"
  latitude=ifelse(latitude_dir=='N', latitude_num, -latitude_num)
  #exact grid points
  temp2=sapply(temp1,'[', -1)
  #get grid of useless variables
  temp3=strsplit(unlist(temp2), ":")
  data_content_temp=sapply(temp3, '[', -1)
  data_content_temp1=strsplit(data_content_temp, " ")
  #remove all " "
  data_content_temp1=lapply(data_content_temp1, function(x) x[nchar(x)>
0])
  data_content_clean=unlist(data_content_temp1)
  #repeat latitude
  reptimes_latitude=length(data_content_temp1)
  latitude_rep=rep(latitude,each=reptimes_latitude)
  #repeat longtitude
  reptimes_longtitude=length(data_content_temp1[[1]])
  longtitude_rep=rep(longtitude,reptimes_longtitude)
  ##deal with time
  #exact time
  time=unlist(strsplit(raw_data[5], ' '))
  #exact Date
  Date=time[nchar(time)>0][3]
  #change data format
  Date=as.Date(Date, format="%d-%b-%Y")
  #repeat Date
  Date_rep=rep(Date,length(data_content_clean))
  #combine these 4 variables
  data_clean=data.frame(Date_rep, latitude_rep, longtitude_rep, as.nume
ric(data_content_clean))
  #get the name for what you read. e.g. the name for "cloudhigh[0-9].tx
t" is "cloudhigh"
  Name_for_content=gsub("[0-9].*", "", x)
  colnames(data_clean)=c('Date','Latitude', 'Longtitude', Name_for_cont
ent)
  return(data_clean)
}
```

Now, let's list all possible patterns of data files.

```
Var_Name=c("cloudhigh", "cloudmid", "cloudlow", "ozone", "pressure", "s
urftemp", "temperature")
```

Here is a function to combine all 72 files for a single variable.

```
#For this function, the input is the index for "Var_Name", the output i
s the combining data.frame for 72 files for one variable.
combine_data=function(i){
  #find all files names related the specific pattern.
  All_files=unlist(lapply(Var_Name[i], function(patterns) list.files(ge
twd(),pattern=patterns)))
  #read all files
  read_All=lapply(All_files,Cleaning_Data)
  #combine those files by row into a big dataframe
  Combine_Data = do.call(rbind,read_All)
  #show the result
  Combine_Data
}
```

Now let's dell with the data files for all 7 variables. Also we will check some rows to make sure it's correct.

```
#get 7 data.frames for 7 variables
Data_List_Diffvar=lapply(1:length(Var_Name),combine_data)
names(Data_List_Diffvar)=Var_Name
#list the 1st, 601st,1201st, 240001st, 410001st rows data for each vari
able
lapply(1:length(Var_Name),function(i) Data_List_Diffvar[[i]][c(1,601,12
01,24001,41001),])
```

```
## [[1]]
##             Date Latitude Longtitude cloudhigh
## 1     1995-01-16     36.2     -113.8        26
## 601   1995-10-16     33.8     -113.8         2
## 1201  1995-11-16     31.2     -113.8         6
## 24001 1998-11-16     -3.8     -113.8         0
## 41001 1995-09-16     26.2      -93.8         7
##
## [[2]]
##             Date Latitude Longtitude cloudmid
## 1     1995-01-16     36.2     -113.8     34.5
## 601   1995-10-16     33.8     -113.8      9.0
## 1201  1995-11-16     31.2     -113.8      9.0
## 24001 1998-11-16     -3.8     -113.8      0.0
## 41001 1995-09-16     26.2      -93.8      7.5
##
## [[3]]
##             Date Latitude Longtitude cloudlow
## 1     1995-01-16     36.2     -113.8      7.5
## 601   1995-10-16     33.8     -113.8     17.0
## 1201  1995-11-16     31.2     -113.8     13.0
```

```
## 24001 1998-11-16     -3.8     -113.8     15.5
## 41001 1995-09-16     26.2      -93.8     30.0
##
## [[4]]
##             Date Latitude Longtitude ozone
## 1     1995-01-16     36.2     -113.8   304
## 601   1995-10-16     33.8     -113.8   274
## 1201  1995-11-16     31.2     -113.8   282
## 24001 1998-11-16     -3.8     -113.8   262
## 41001 1995-09-16     26.2      -93.8   272
##
## [[5]]
##             Date Latitude Longtitude pressure
## 1     1995-01-16     36.2     -113.8      835
## 601   1995-10-16     33.8     -113.8      915
## 1201  1995-11-16     31.2     -113.8      970
## 24001 1998-11-16     -3.8     -113.8     1000
## 41001 1995-09-16     26.2      -93.8     1000
##
## [[6]]
##             Date Latitude Longtitude surftemp
## 1     1995-01-16     36.2     -113.8    272.7
## 601   1995-10-16     33.8     -113.8    296.9
## 1201  1995-11-16     31.2     -113.8    293.2
## 24001 1998-11-16     -3.8     -113.8    296.0
## 41001 1995-09-16     26.2      -93.8    304.0
##
## [[7]]
##             Date Latitude Longtitude temperature
## 1     1995-01-16     36.2     -113.8       272.1
## 601   1995-10-16     33.8     -113.8       297.8
## 1201  1995-11-16     31.2     -113.8       297.4
## 24001 1998-11-16     -3.8     -113.8       297.4
## 41001 1995-09-16     26.2      -93.8       302.3
```

Here we can get a list of 7 dataframes for 7 variables called "Data_List_Diffvar"

Step 2.

First, let's see whether the values for "Date", "Longtitude" and "Latitude" are the same and also in the same order for all 7 data.frames. My strategy is to compare values of "Date", "Longtitude" and "Latitude" between data.frame(1,2), (2,3),(3,4), (4,5), (5,6) and (6,7). If we get 6*3 "True", then the observations for each the 7 variables and for each date correspond to the same collection of points on the grid, and in the same order.

```
#This funtion will pass different data.frame into sapply function
Check_Equal_two=function(vars){
  #To see whether it's 'equal' for each 3 column between two given data.
```

```
frame
  sapply(1:3, function(i) all.equal(Data_List_Diffvar[[vars]][i],Data_L
ist_Diffvar[[vars+1]][i]))
}
#passing different data.frames into "Check_Equal_two" to see whether th
ay are the same
sapply(1:6, Check_Equal_two)
```

```
##      [,1] [,2] [,3] [,4] [,5] [,6]
## [1,] TRUE TRUE TRUE TRUE TRUE TRUE
## [2,] TRUE TRUE TRUE TRUE TRUE TRUE
## [3,] TRUE TRUE TRUE TRUE TRUE TRUE
```

From the result, which is all "TRUE", we can conclude that the observations for each the 7 variables and for each date correspond to the same collection of points on the grid, and in the same order.

Since the first three columns are the same, we can easily combine the dataset by adding 7 variables together.

```
#exact all variables from 7 data.frames
Contents=lapply(1:7, function(i) Data_List_Diffvar[[i]][4])
#combine them
Contents_combine=do.call(cbind, Contents)
#add "Date", "Longtitude" and "Latitude" into 7 variables.
Final_Clean_Data=cbind(Data_List_Diffvar[[1]][1:3],Contents_combine)
#show some results
head(Final_Clean_Data)
```

```
##         Date Latitude Longtitude cloudhigh cloudmid cloudlow ozone
## 1 1995-01-16     36.2     -113.8      26.0     34.5      7.5   304
## 2 1995-01-16     36.2     -111.2      23.0     32.0      7.0   306
## 3 1995-01-16     36.2     -108.8      23.0     32.0      7.0   306
## 4 1995-01-16     36.2     -106.2      17.0     29.5      7.0   294
## 5 1995-01-16     36.2     -103.8      19.5     33.0     11.0   308
## 6 1995-01-16     36.2     -101.2      17.0     34.0     14.5   310
##   pressure surftemp temperature
## 1      835    272.7       272.1
## 2      810    270.9       270.3
## 3      810    270.9       270.3
## 4      775    269.7       270.9
## 5      795    273.2       271.5
## 6      915    275.6       275.6
```

Step 3.

I write a function for adding data

```
#The input for this function is "x: the name of the data file" & "old_d
ata: the original dataset that you want to add new variables in" & "Col
```

```
umn_name: the column name that you want to assign for this new variable
".
Adding_var<-function(x,old_data,Column_name){
  #read data file
  data_content=readLines(x)
  #split latitude and grid points
  temp1=strsplit(data_content[-1]," ")
  #exact grid points
  data_content_clean=unlist(sapply(temp1,'[',-1))
  #repeat data for 72 times
  data_rep=as.numeric(rep(data_content_clean,72))
  #add this new variable
  data_clean=data.frame(old_data,data_rep)
  #change the column name that you want
  names(data_clean)[names(data_clean) == 'data_rep']=Column_name
  return(data_clean)
}
```

This function can be applied to all new variables that you want to add.

Now let's add the new "elevation" variable into original files

```
New_Data=Adding_var("intlvtn.dat", Final_Clean_Data,"elevation")
head(New_Data)
```

```
##         Date Latitude Longtitude cloudhigh cloudmid cloudlow ozone
## 1 1995-01-16     36.2     -113.8      26.0     34.5      7.5   304
## 2 1995-01-16     36.2     -111.2      23.0     32.0      7.0   306
## 3 1995-01-16     36.2     -108.8      23.0     32.0      7.0   306
## 4 1995-01-16     36.2     -106.2      17.0     29.5      7.0   294
## 5 1995-01-16     36.2     -103.8      19.5     33.0     11.0   308
## 6 1995-01-16     36.2     -101.2      17.0     34.0     14.5   310
##   pressure surftemp temperature elevation
## 1      835    272.7       272.1   1526.25
## 2      810    270.9       270.3   1759.56
## 3      810    270.9       270.3   1948.38
## 4      775    269.7       270.9   2241.31
## 5      795    273.2       271.5   1692.75
## 6      915    275.6       275.6    865.19
```

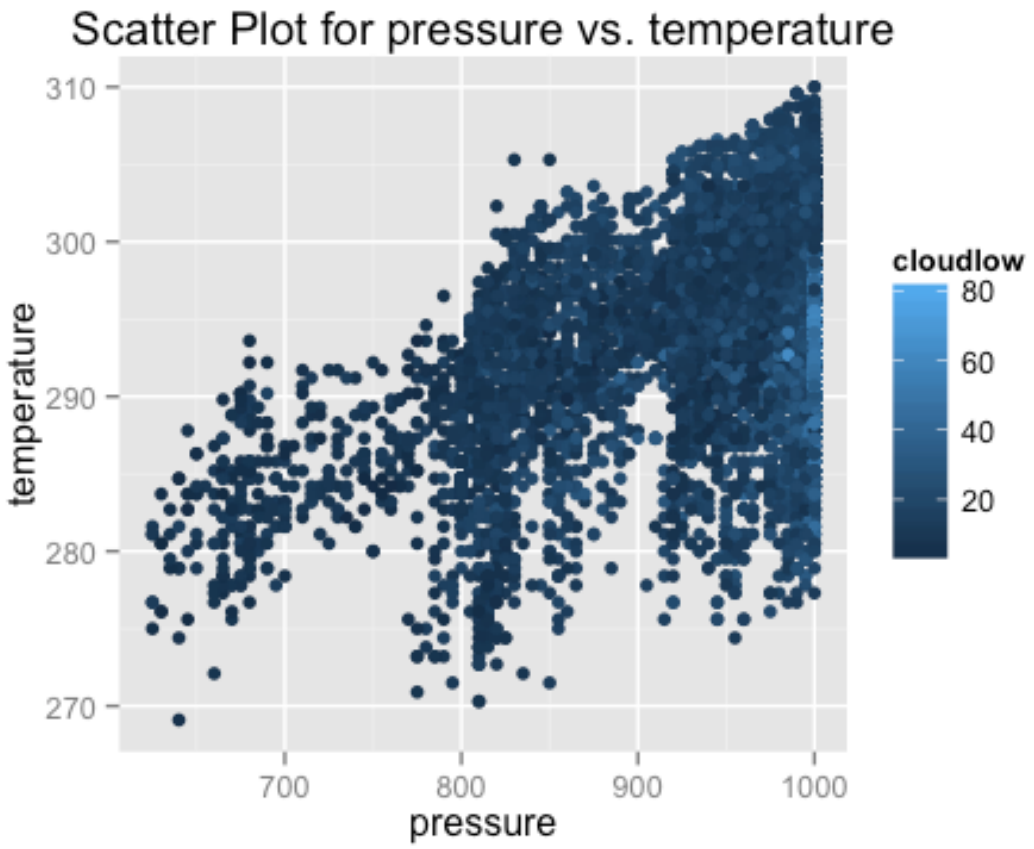
Step 4.

1.

I use ggplot to draw this plot

```
library(ggplot2)
#remove all "NA" data
New_Data_clean=subset(New_Data, !is.na(New_Data$cloudlow))
#show ggplot
```

```r
ggplot(New_Data_clean,aes(x=pressure, y=temperature,color=cloudlow,5))
+
  geom_point()+
  labs(list(title = "Scatter Plot for pressure vs. temperature"))
```



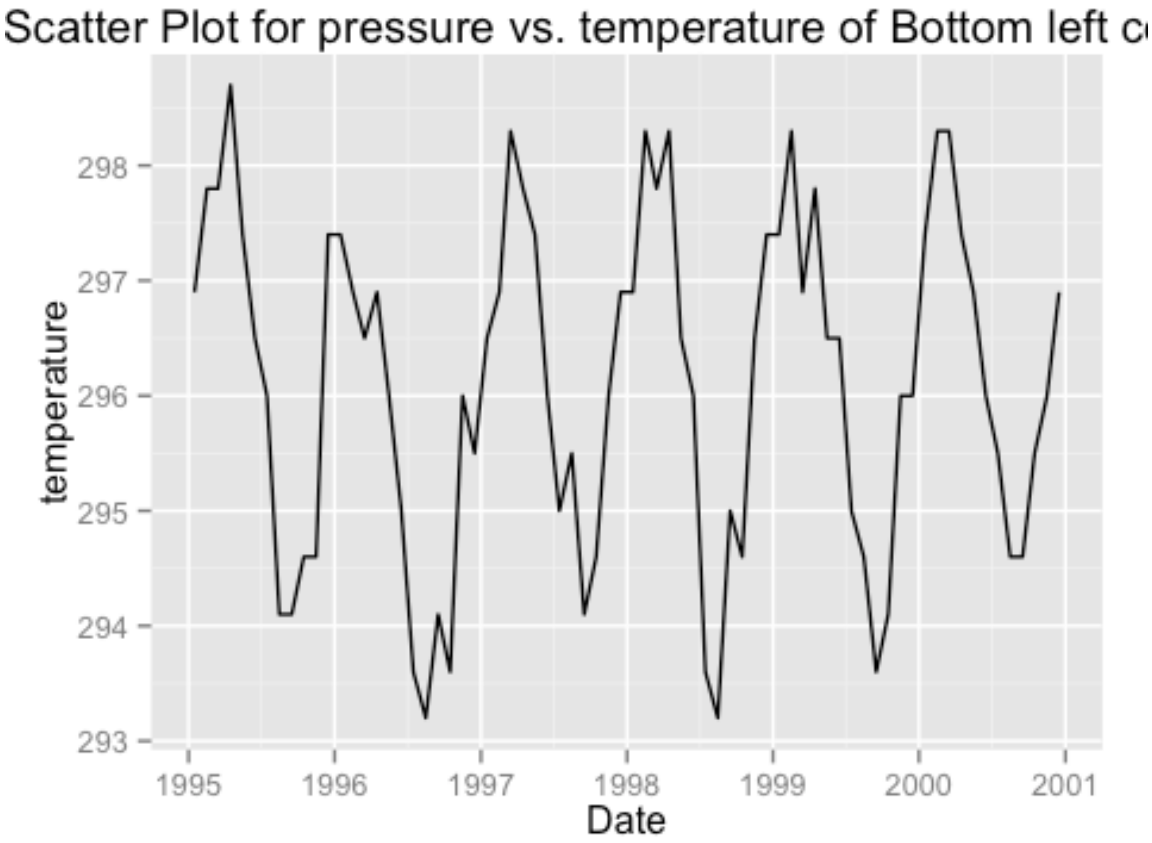
2.

I will grab the four corners' latitudes and longtitudes.

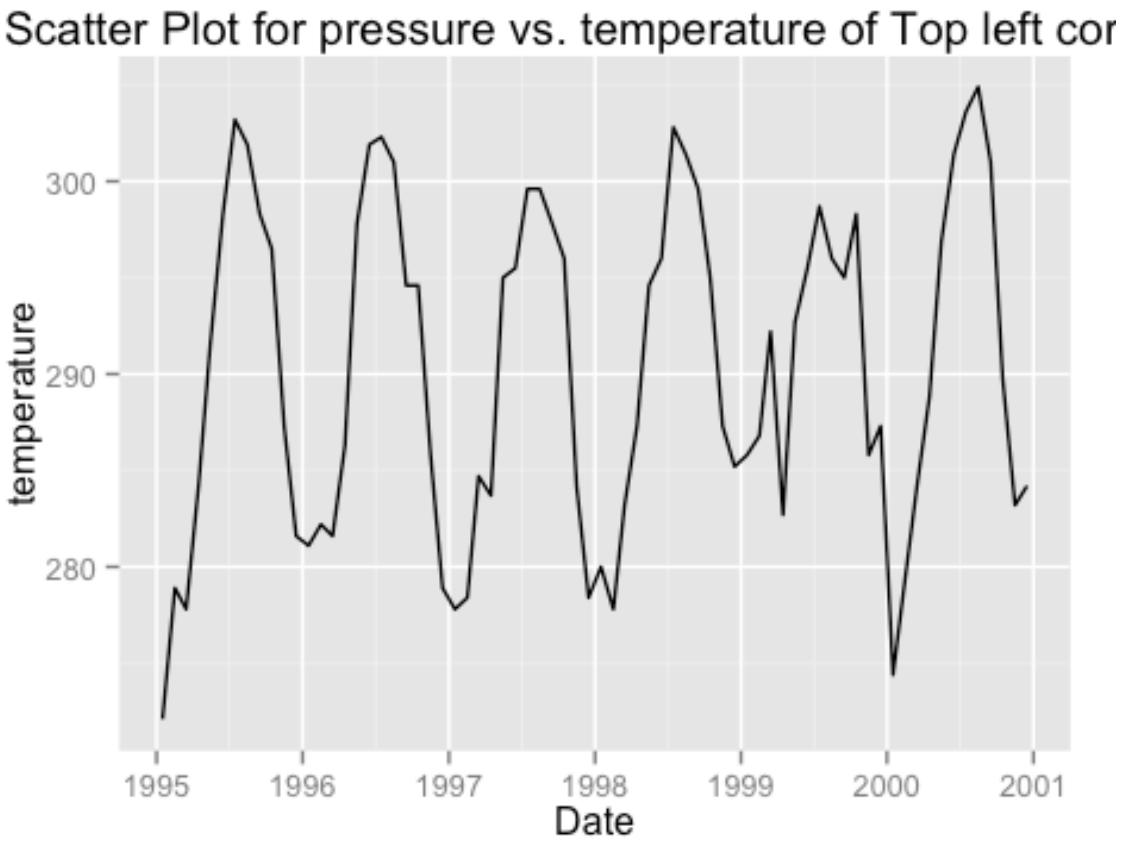
```r
# min and max values for Latitude and Longtitude
Long_exteme=c(min(New_Data$Longtitude),max(New_Data$Longtitude))
Lat_exteme=c(min(New_Data$Latitude), max(New_Data$Latitude))
#This function is to return "Date" and "temperature" under min & max of
 Latitude, given a Longtitude.
Corner_temp=function(j){
results=lapply(1:2,function(i) subset(New_Data[,c('Date','temperature
')], New_Data$Longtitude==Long_exteme[j] & New_Data$Latitude==Lat_extem
e[i]))
return(results)
}

#passing longtitude value to function "Corner_temp"
Corner_temp_four=sapply(1:2,Corner_temp)
```

Now Let's draw pictures for 4 corners.

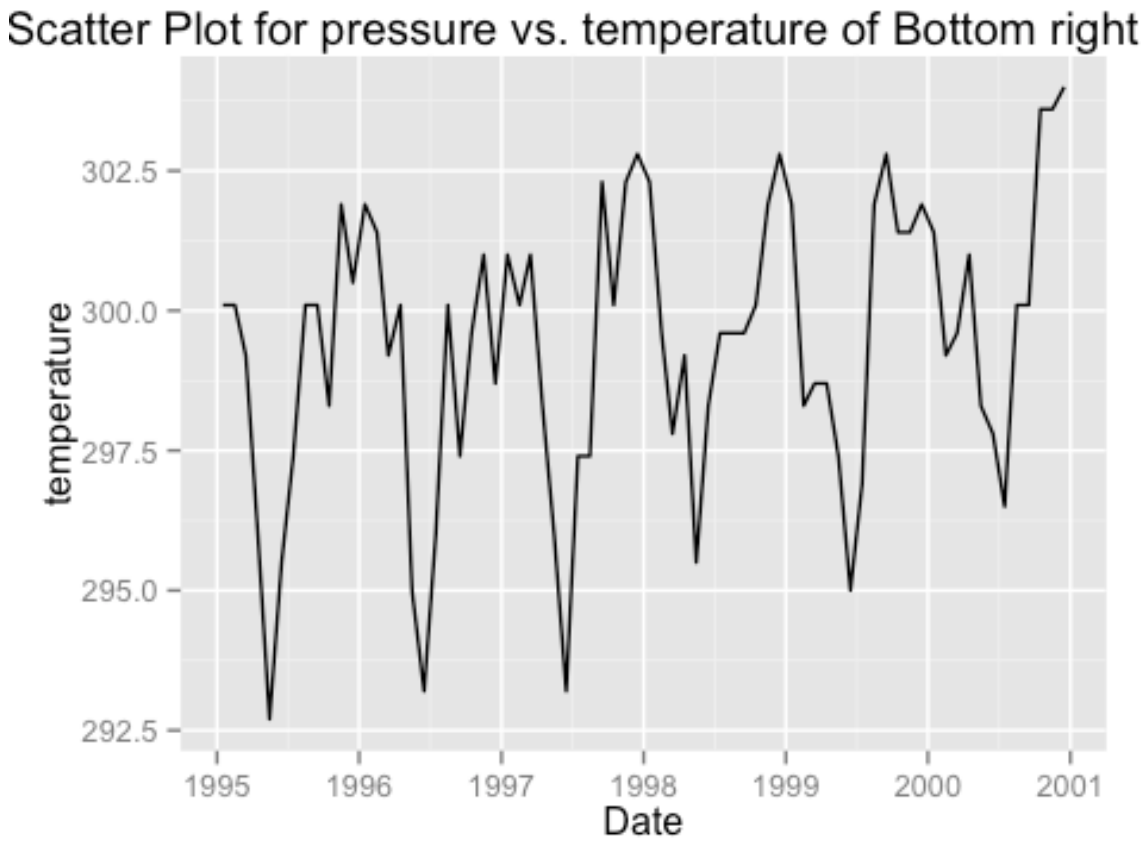
```
ggplot(Corner_temp_four[[1]], aes(x=Date, y=temperature))+
  geom_line()+
  labs(list(title = "Scatter Plot for pressure vs. temperature of Botto
m left corner"))
```



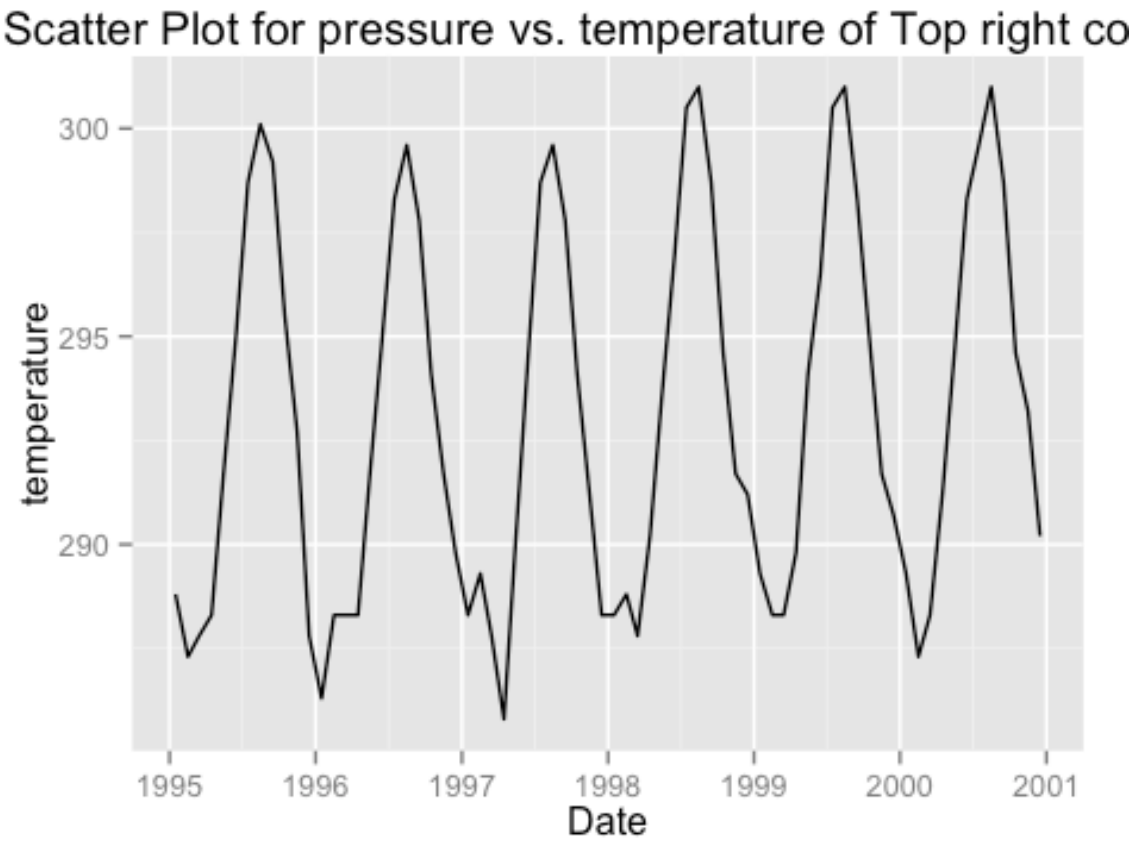

```
ggplot(Corner_temp_four[[2]], aes(x=Date, y=temperature))+
  geom_line()+
  labs(list(title = "Scatter Plot for pressure vs. temperature of Top l
eft corner"))
```




```
ggplot(Corner_temp_four[[3]], aes(x=Date, y=temperature))+
  geom_line()+
  labs(list(title = "Scatter Plot for pressure vs. temperature of Botto
m right corner"))
```




```
ggplot(Corner_temp_four[[4]], aes(x=Date, y=temperature))+
  geom_line()+
  labs(list(title = "Scatter Plot for pressure vs. temperature of Top r
ight corner"))
```



3.

Assign the original data into a new dataset

```
Data_for4=New_Data
```

Change "Latitude"" and "Longtitude" for this new dataset into factor. Thus we can use the levels to find the data for a specific position

```
Data_for4$Latitude=as.factor(Data_for4$Latitude)
Latitude_level=levels(Data_for4$Latitude)
Data_for4$Longtitude=as.factor(Data_for4$Longtitude)
Longtitude_level=levels(Data_for4$Longtitude)
```

This funtion aims to get all values under different 24 Longtitude_levels when a Latitude_level has already been assigned.

```
fix_latitude=function(Lat_index){
#Since 'Data_for4' and 'New_Data' are basicly the same data except the
type of "Latitude" and 'Longtitude', so we can get the data index(numbe
rs of row) from 'Data_for4'and use them to 'New_Data'.Now we can find t
hose values for different levels of 'Longtitude', given a "Latitude".
result=lapply(1:24, function(j) subset(New_Data, Data_for4$Latitude==La
```

```
titude_level[Lat_index] & Data_for4$Longtitude==Longtitude_level[j]))

  return(result)
}
```

passing all possible "Lat_index" to function "fix_latitude".

```
Data_for_all_lat=sapply(1:24,function(Lat_index) fix_latitude(Lat_inde
x))
```

What we get here is a big list that contains of 24 x 24=576 data.frames.

Now, let see get the mean and sd for each variables under different positions

```
#get rid of the "Date" column, to calculate means for each data.frame.
Since "Latitude" and "Longtitude" in each data.frame are the same, so t
he mean of them will be the same as their really values.
Mean_All=as.data.frame(t(sapply(1:576, function(j) round(apply(Data_for
_all_lat[[j]][,-1],2,mean),2))))
#show some results for Mean_All
head(Mean_All)
```

```
##   Latitude Longtitude cloudhigh cloudmid cloudlow  ozone pressure su
rftemp
## 1    -21.2     -113.8      1.99     5.78    37.17 268.25     1000
296.24
## 2    -21.2     -111.2      1.63     5.24    38.83 268.94     1000
295.84
## 3    -21.2     -108.8      1.31     5.15    40.56 269.14     1000
295.46
## 4    -21.2     -106.2      1.17     5.71    42.92 269.47     1000
295.07
## 5    -21.2     -103.8      1.08     6.55    43.69 269.75     1000
294.47
## 6    -21.2     -101.2      1.05     6.91    44.77 269.44     1000
294.02
##   temperature elevation
## 1      296.11         0
## 2      295.78         0
## 3      295.41         0
## 4      294.98         0
## 5      294.71         0
## 6      294.36         0
```

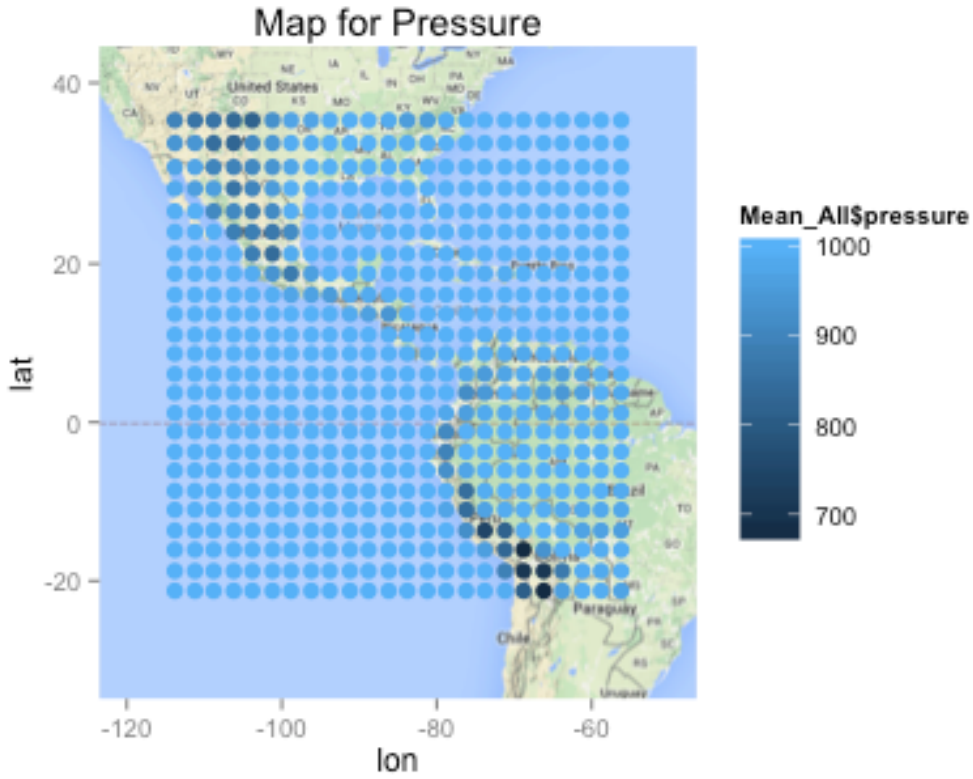
```
#get rid of "Date", "Latitude" and "Longtitude" columns, to calculate s
ds for each data.frame.
SD_All=as.data.frame(t(sapply(1:576, function(j) round(apply(Data_for_a
ll_lat[[j]][,-(1:3)],2,sd),2))))
#add "Latitude" and "Longtitude" back.
SD_All=cbind(Mean_All[1:2],SD_All)
# show some results
head(SD_All)
```

```
##   Latitude Longtitude cloudhigh cloudmid cloudlow ozone pressure sur
ftemp
## 1    -21.2     -113.8      2.77     3.82     5.78 12.26        0
 1.50
## 2    -21.2     -111.2      2.29     3.35     5.69 12.51        0
 1.51
## 3    -21.2     -108.8      1.96     3.26     5.99 12.56        0
 1.56
## 4    -21.2     -106.2      1.62     3.65     6.34 12.73        0
 1.57
## 5    -21.2     -103.8      1.72     4.23     6.73 12.48        0
 1.55
## 6    -21.2     -101.2      1.81     4.70     6.84 12.45        0
 1.62
##   temperature elevation
## 1        1.48         0
## 2        1.51         0
## 3        1.46         0
## 4        1.50         0
## 5        1.50         0
## 6        1.53         0
```

4.

Now let's draw map.

```
library(ggmap)
#we take the mean of Longtitude and Latitue to be as our center positio
n of map.
nasa_center <- c(lon=mean(Mean_All$Longtitude), lat=mean(Mean_All$Latit
ude))
#get the map
nasamap=get_googlemap(center = nasa_center, zoom=3)

## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=
7.5,-85&zoom=3&size=640x640&maptype=terrain&sensor=false

ggmap(nasamap)+
  geom_point(aes(x=Mean_All$Longtitude,y=Mean_All$Latitude,col=Mean_All
$pressure),size=2.5)+
  labs(list(title = "Map for Pressure"))
```

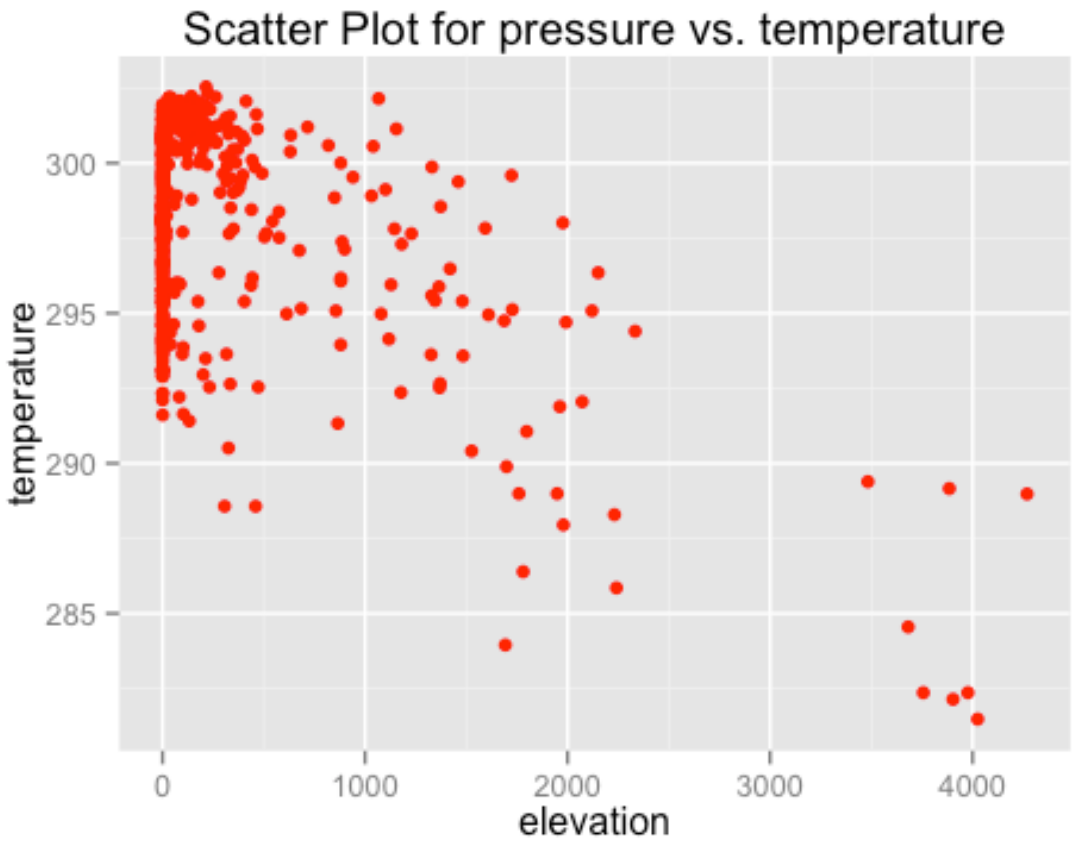


From the plot, we can see that the pressure is lower in the south-west of North America, (e.g.New Mexico in USA and Mexico) and left-center of South America, (e.g. Peru and Bolivia).

5.

From (3), the values of "Latitude", "Longtitude" and "elevation" are the same in each data.frame. Thus, the mean for them are equal to them selves. So we can draw the plot directly.

```
ggplot(Mean_All,aes(x=elevation,y=temperature)) +
  geom_point(color="red")+
  labs(list(title = "Scatter Plot for pressure vs. temperature"))
```



From the plot, we can conclude that temperature and elevation have a roughly negative relationship.