Assignment-1

Md Kamrul Hasan Khan

January 27, 2020

Homework must be submitted as a single pdf file which is the output of the RMarkdown file. Please save the file as `Last-Name-assignment-1.pdf` and email it to jrtipton@uark.edu. For example, the student with the last name of Fox would submit the file `Fox-assignment-1.pdf`.

The objective of this assignment is to familiarize yourself with data. In the assignment, you will solve problems using `RMarkdown`.

Summary of Data

The data provided in *USHCN.RData* posted on the website consists of multiple objects from the United States Historical Climatology Network (http://cdiac.ornl.gov/epubs/ndp/ushcn/ushcn.html).

```
load(here::here("data", "USHCN.RData"))
```

There are n 41975 observations of daily precipitation measurements (in hundredths of inches) at n_s 1218 sites in the $n \times n_s$ matrix *precip*. The $n_s \times 2$ matrix *lon_lat* contains the longitude (first column) and latitude (second column) for each of the n_s stations. The n-vector *days* contains the day of the year (1-365) of the measurement (ignoring leap-days) and the n-vector *years* contains the year of the measurement. The n_s vector *elev* contains the elevation of each station and the n_s vector *station_name* contains the name of each station.

Problem 1

Create a single data.frame that is in long format (each row is an observation, each column is a variable – the data.frame should have 51125550 rows) that merges the objects *precip*, *lon_lat*, *days*, *years*, *elev*, and *station_name*. For examples of *tidy* data , see https://tidyr.tidyverse.org/articles/tidy-data.html and https://tidyr.tidyverse.org/

Solution of Problem 1

A single data.frame creation:

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.0.5

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.1     v stringr 1.4.0
## v tidyr   1.1.3     v forcats 0.5.1
## v readr   1.4.0

## Warning: package 'ggplot2' was built under R version 4.0.5

## Warning: package 'tibble' was built under R version 4.0.5

## Warning: package 'tidyr' was built under R version 4.0.5

## Warning: package 'readr' was built under R version 4.0.5

## Warning: package 'purrr' was built under R version 4.0.5

## Warning: package 'stringr' was built under R version 4.0.5

## Warning: package 'forcats' was built under R version 4.0.5

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(tidyr)

load(here::here("data", "USHCN.RData"))

d1 <- data.frame(precip, days, years)
n = nrow(precip)
d2 <- d1 %>%
    gather(precip, precipitation, X1:X1218) %>%
    select(-precip) %>%
    mutate(longitue = rep(lon_lat[ , 1], each = n) ) %>%
    mutate(latitude = rep(lon_lat[ , 2], each = n) ) %>%
    mutate(elevation = rep(elev, each = n) ) %>%
    mutate(station = rep(station_name, each = n) )
```

Problem 2

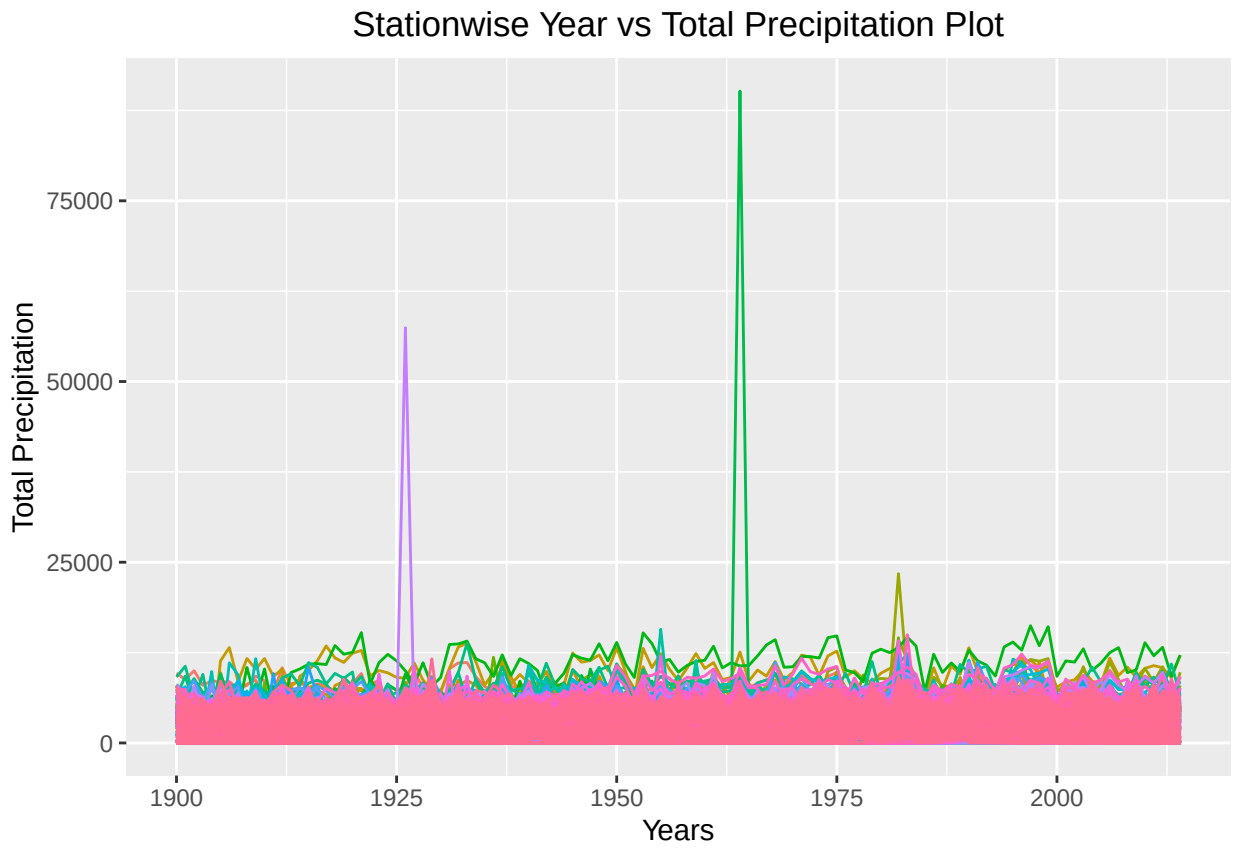
For each station, calculate the total annual rainfall at each site. Plot these annual summaries as a "spaghetti" plot where each station is given its own line.

Solution of Problem 2

```
d3 <- d2 %>%
    group_by(years, station) %>%
    summarize(total_precip = sum(precipitation,  na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'years'. You can override using the `.groups` argument.
```

```
d3 %>%
    ggplot(aes(x = years, y = total_precip, col = station)) +
    geom_line() +
    xlab("Years")+
    ylab("Total Precipitation") +
    labs(title = "Stationwise Year vs Total Precipitation Plot")+
    theme(legend.position = "none", plot.title = element_text(hjust = 0.5))
```



Problem 3

Choose 3 sites from the data. Calculate a linear regression of daily precipitation vs. year at each of the sites. Provide figures and write a few sentences describing the results. Make sure you write out the answers in **clear, English sentences. Communication matters!**

Solution of Problem 3

This data contains significant proportion (around 17%) of missing data. Hence below I choose three sites with least missing data (0%, 0.02%, 0.06%). Then are linear regression model is run for each station independently where the response constructed by taking average of all year's precipitations at each day and the covariate is those yearly precipitations on the corresponding day. Here the reponse is actully a function of covriates and for that reason all the regression coefficents are simmilar. The residual plots in next three figures shows that the residuals are significantly small of all stations. This is because the response can be written as a function of all covariates. The qqplots shows that the data follows approximately normal distribution.

```
aa = apply(precip, 2, function(x) sum(x >=0, na.rm = TRUE))
bb = sort(aa, decreasing = TRUE)

st_1 <- station_name[which(aa == bb[1])]
st_2 <- station_name[which(aa == bb[2])]
st_3 <- station_name[which(aa == bb[3])]

d4 <- d2 %>%
  group_by(days, station) %>%
  summarize(total_precip = mean(precipitation,  na.rm = TRUE))
```

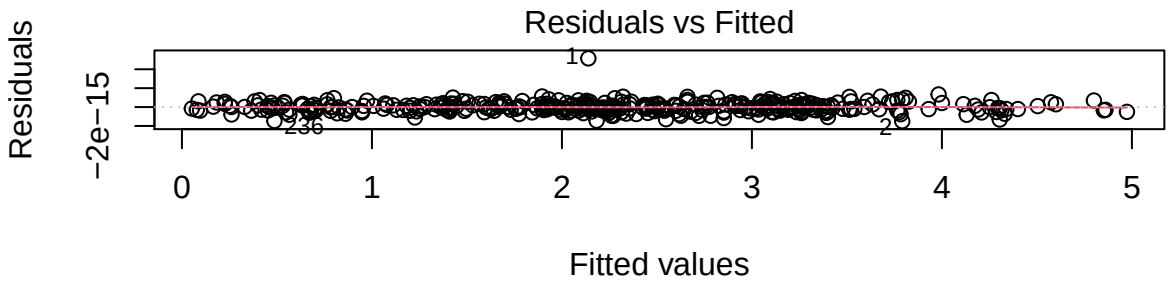
```
## `summarise()` has grouped output by 'days'. You can override using the `.groups` argument.
```

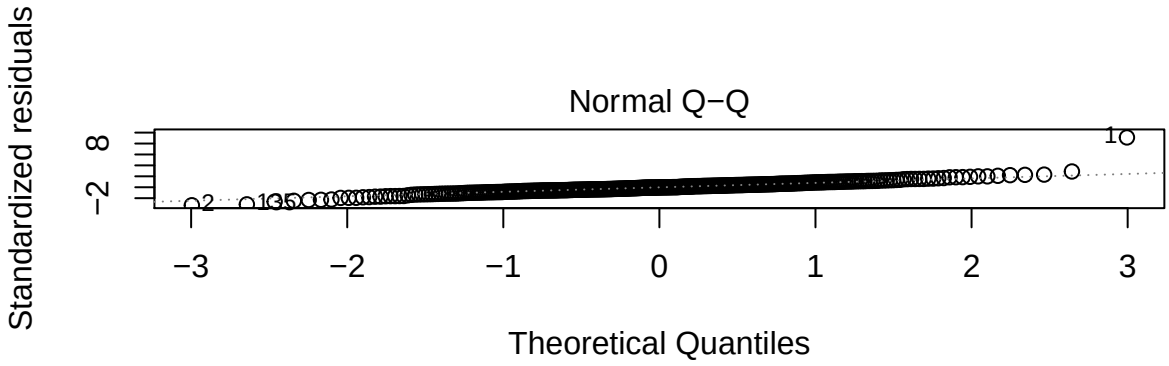
```
## For first station

yy_1 <- d4 %>%
  filter(station == st_1)%>%
  pull(total_precip)

d_st_1 <- d2 %>%
  pivot_wider(names_from = years, values_from = precipitation) %>%
  filter(station == st_1) %>%
  select(-(1:5)) %>%
  mutate(response = yy_1)

station_1_lm = lm( response ~. , data = d_st_1 )
par(mfrow = c(2, 1))
plot(station_1_lm, which = 1:2)
```



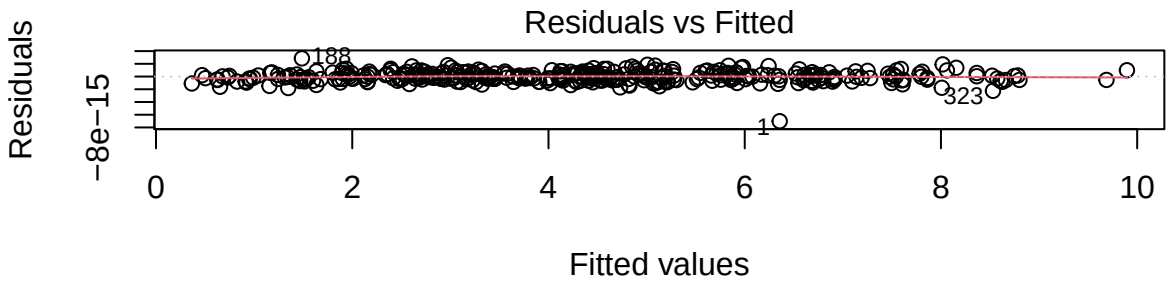


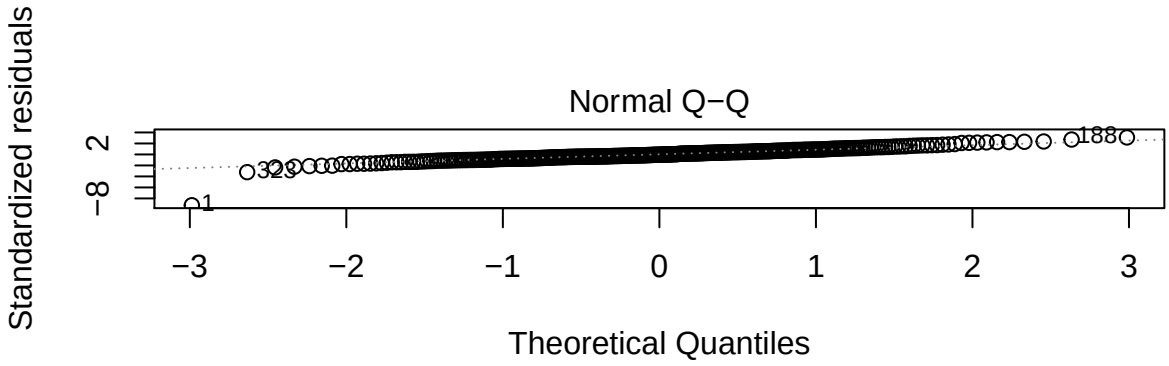

```
## For second station

yy_2 <- d4 %>%
  filter(station == st_2)%>%
  pull(total_precip)

d_st_2 <- d2 %>%
  pivot_wider(names_from = years, values_from = precipitation) %>%
  filter(station == st_2) %>%
  select(-(1:5)) %>%
  mutate(response = yy_2)

station_2_lm = lm( response ~. , data = d_st_2 )
par(mfrow = c(2, 1))
plot(station_2_lm, which = 1:2)
```



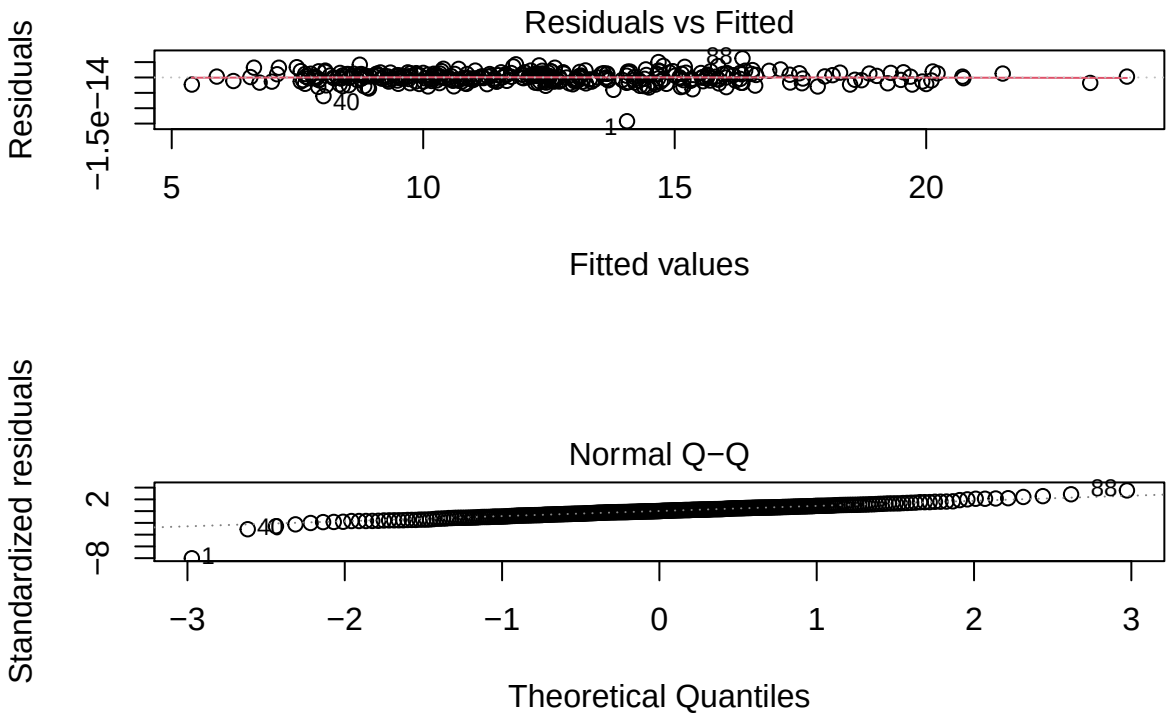



```r
## For third station

yy_3 <- d4 %>%
  filter(station == st_3)%>%
  pull(total_precip)

d_st_3 <- d2 %>%
  pivot_wider(names_from = years, values_from = precipitation) %>%
  filter(station == st_3) %>%
  select(-(1:5)) %>%
  mutate(response = yy_3)

station_3_lm = lm( response ~. , data = d_st_3 )
par(mfrow = c(2, 1))
plot(station_3_lm, which = 1:2)
```



Problem 4

Create a figure that displays the data. This can be a map of raw data, summarized data (i.e. like in problem 2), animation, time series plot, etc. that you think describes an interesting feature in the data. Write a few sentences that explain your plot and what makes it interesting. Make sure you write out the answers in **clear, English sentences. Communication matters!**

Solution of Problem 4

Since this data has a significant proportion of NAs, a summary data representation may be more informative than representing the whole raw data. Moreover, the temporal patern may explain the data more clearly. Therefore, the next figure dipicts the plot of station wise daily precipitation averaged over the years to show the patern of the precipitation within years. The figure represents that although in most of the station the averaged precipitation remains similar over the year, in some station there is significant jumps at the middle of the year.

```
d2 %>%
  group_by(days, station) %>%
  summarize(mean_precip = mean(precipitation,  na.rm = TRUE))%>%
    ggplot(aes(x = days, y = mean_precip, col = station)) +
    geom_line() +
    ylab("Average Precipitation") +
```

```
    xlab("Days") +
    labs(title = "Stationwise Days vs Average Precipitation Plot")+
    theme(legend.position = "none", plot.title = element_text(hjust = 0.5))
```

```
## `summarise()` has grouped output by 'days'. You can override using the `.groups` argument.
```