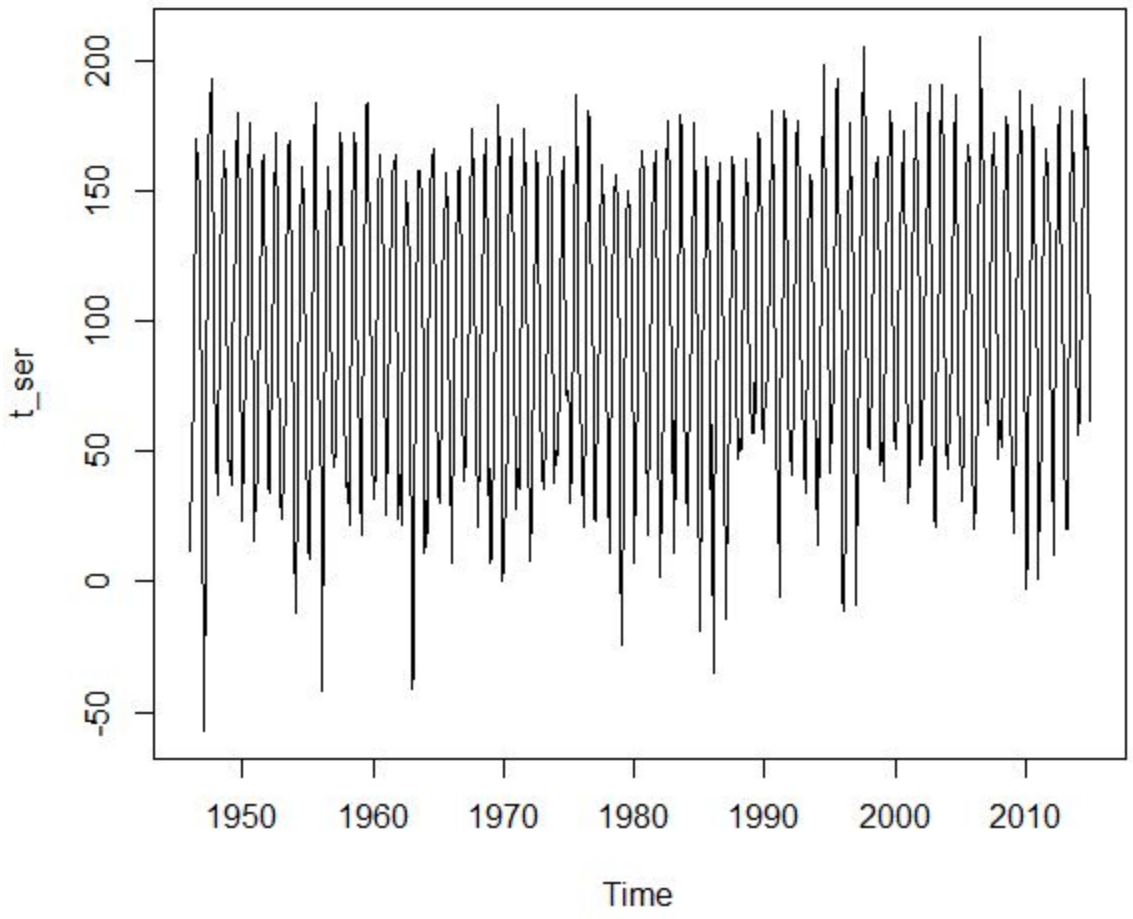
Data Analysis and Statistical Methods
Frans Simanjuntak - S3038971
Problem set 6

1. ***Make a vector t of the times corresponding to each data point combine the times and temperature values in a data frame. Plot the time series vs. t.***

The R-Code

```
library(tidyverse)
library(forecast)

setwd("X:/My Desktop/Statistic/assignment6/assignment6")

df <- read.csv(file = "Pr_20May1(1).csv", header = TRUE)

#rename the column
names(df)[1]<-"temperature"

#create a squence month from 1946 to december 2014
df$t = seq(from = as.Date("1946-01-01"), to = as.Date("2014-12-31"), by="1 month")

#plot the time series
t_ser <- ts(df$temperature, start = c(1946, 1), freq = 12)
plot(t_ser)
```

The output



Can you see a seasonality in the data? Which period is it?

Yes. From the above figure we can see that the the gradual increase of temperature in March. Then, it dramatically rises up in April until it reaches its highest peak in August. Starting from september, the temperature gradually decreases and then it dramatically decreases in November until it reaches its lowest peak in February in the following year. Then the pattern is repeated over and over again each year.

Can you see a trend in the data due to global warming?

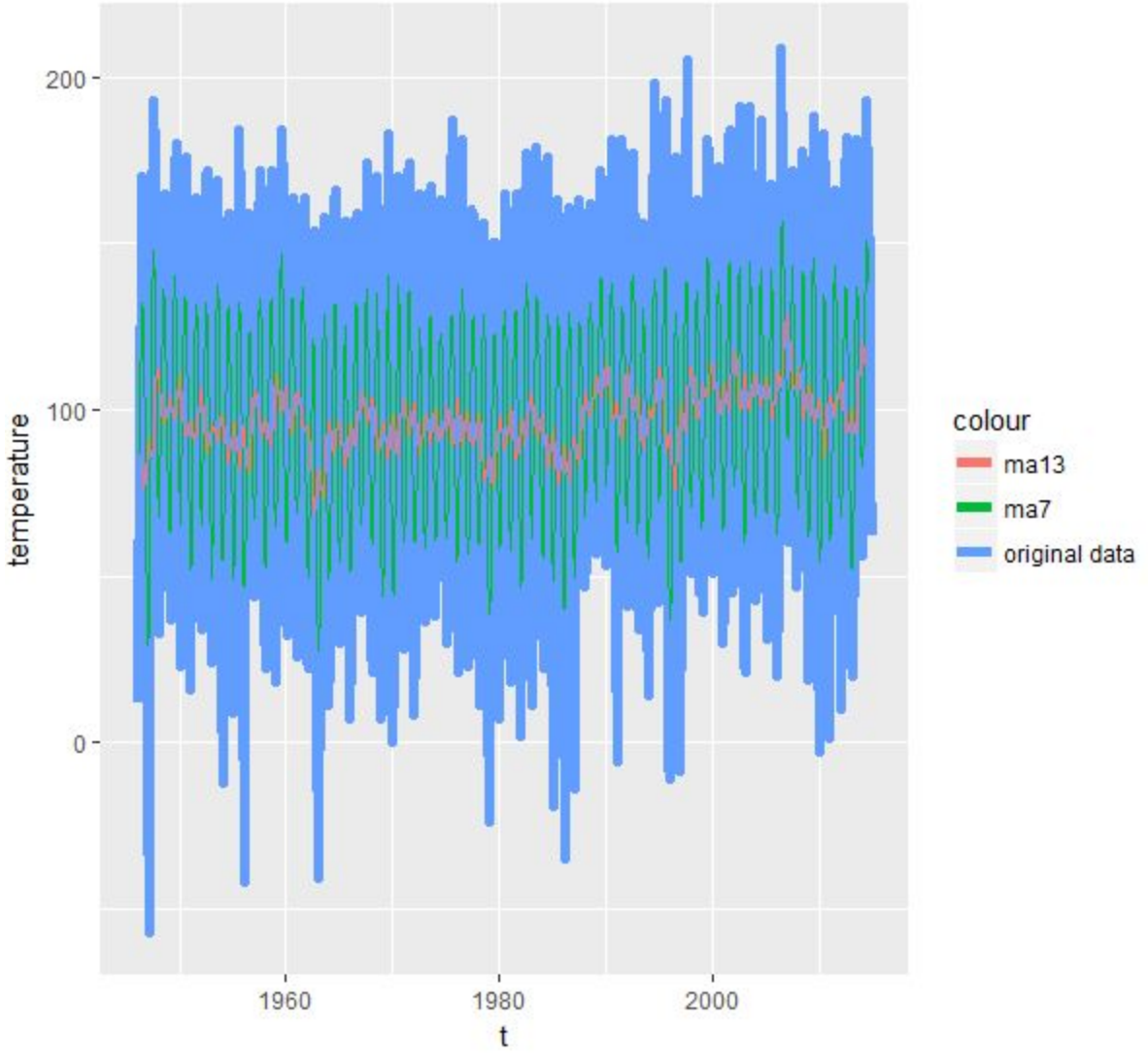
Yes, from the above figure we can see the trend due to global warming. Since 1990, the average of temperature rises up compared to previous years and it reaches its highest peak in July 2006. Even though in March 1991 the temperature dramatically decreases to -57 (if we convert to actual value should be -5.7), however it keeps increasing ever since. This is the fact due to global warming.

2. ***Calculate a moving average with order m = 7 and 13 (see lecture) and add it to the plot.***

The R-Code

```
library(tidyverse)
library(forecast)

setwd("X:/My Desktop/Statistic/assignment6/assignment6")

df <- read.csv(file = "Pr_20May1(1).csv", header = TRUE)

#rename the column
names(df)[1]<-"temperature"

#create a squence month from 1946 to december 2014
df$t = seq(from = as.Date("1946-01-01"), to = as.Date("2014-12-31"), by="1 month")

#moving average with m=7 and m=13
df_with_ma = df %>%
  mutate(ma7 = ma(temperature, order = 7),
         ma13 = ma(temperature, order = 13))

#plot df_with_ma
ggplot(df_with_ma, aes(x=t)) +
  geom_line(aes(y = temperature,
                color="original data"),
            size=2) +
  geom_line(aes(y = ma(temperature,
                       order = 7),
                color="ma7")) +
  geom_line(aes(y = ma(temperature,
                       order = 13),
                color="ma13")) +
  xlim(c(as.Date("1946-01-01"),
         as.Date("2014-12-31")))
```

The Output



Explain the order m means

The order m=7 means that 7 data points are averaged (three to the left and the rest is to the right of the data point)

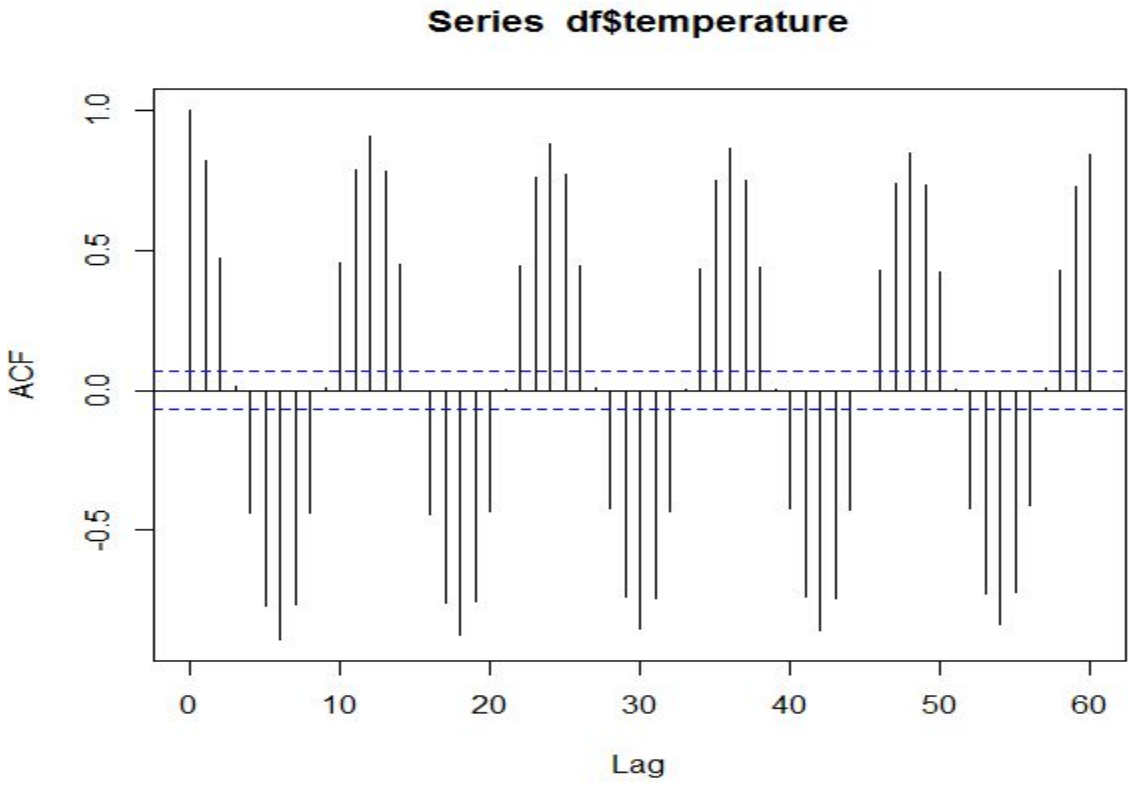
The order m=13 means that 13 data points are averaged (six to the left and the rest is to the right of the data point)

3. ***Calculate the autocorrelation function (ACF) of the time series and plot it up to lag 60.***

The R-Code

```
library(tidyverse)
library(forecast)

setwd("X:/My Desktop/Statistic/assignment6/assignment6")

df <- read.csv(file = "Pr_20May1(1).csv", header = TRUE)

#rename the column
names(df)[1]<-"temperature"

#create a squence month from 1946 to december 2014
df$t = seq(from = as.Date("1946-01-01"), to = as.Date("2014-12-31"), by="1 month")

#Calculate the autocorrelation function (ACF) of the time series and plot it up to lag 60
acf_T = acf(df$temperature, type = "correlation", lag.max = 60)
```

The output



What variability is mainly reflected in the ACF?

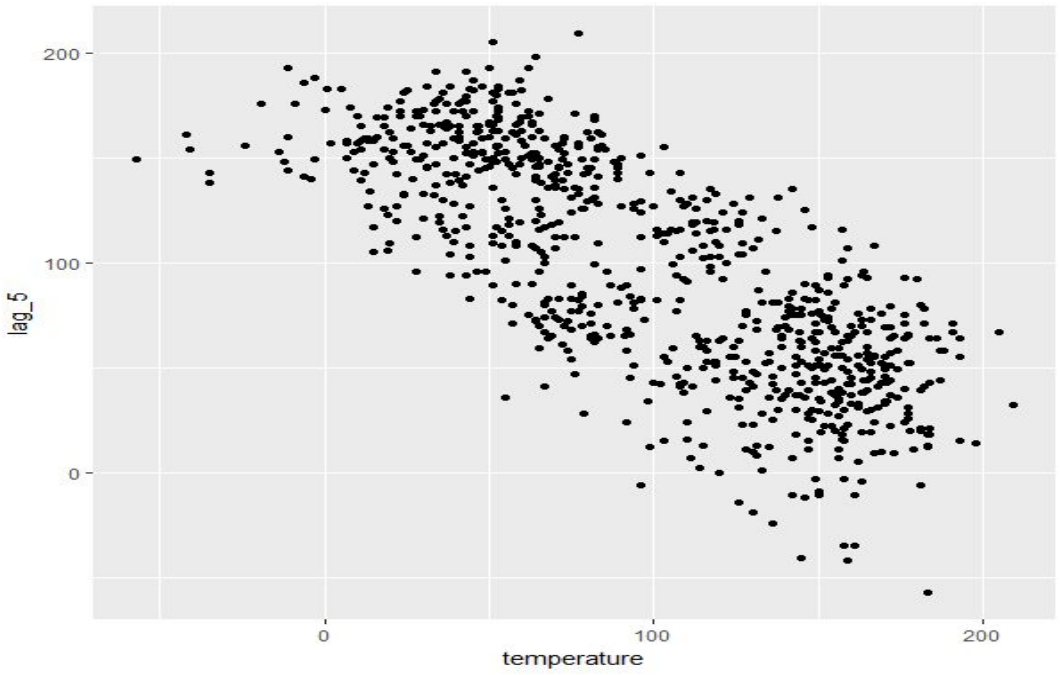
This shows that the time series data is highly correlated on a yearly basis.

4. ***Show the actual correlation (scatter) plot that is used to calculate the ACF at lag 5.***

The R-Code

```
library(tidyverse)
library(forecast)

setwd("X:/My Desktop/Statistic/assignment6/assignment6")

df <- read.csv(file = "Pr_20May1(1).csv", header = TRUE)

#rename the column
names(df)[1]<-"temperature"

#create a squence month from 1946 to december 2014
df$t = seq(from = as.Date("1946-01-01"), to = as.Date("2014-12-31"), by="1 month")

## data shifted 5
ggplot(data = df %>% mutate(lag_5 = lag(temperature, 5)),
     aes(x=temperature, y=lag_5)) +
 geom_point()
```

The Output



5. ***Decompose the time series using classical decomposition.***
Hint: you need to construct an appropriate time series object first.

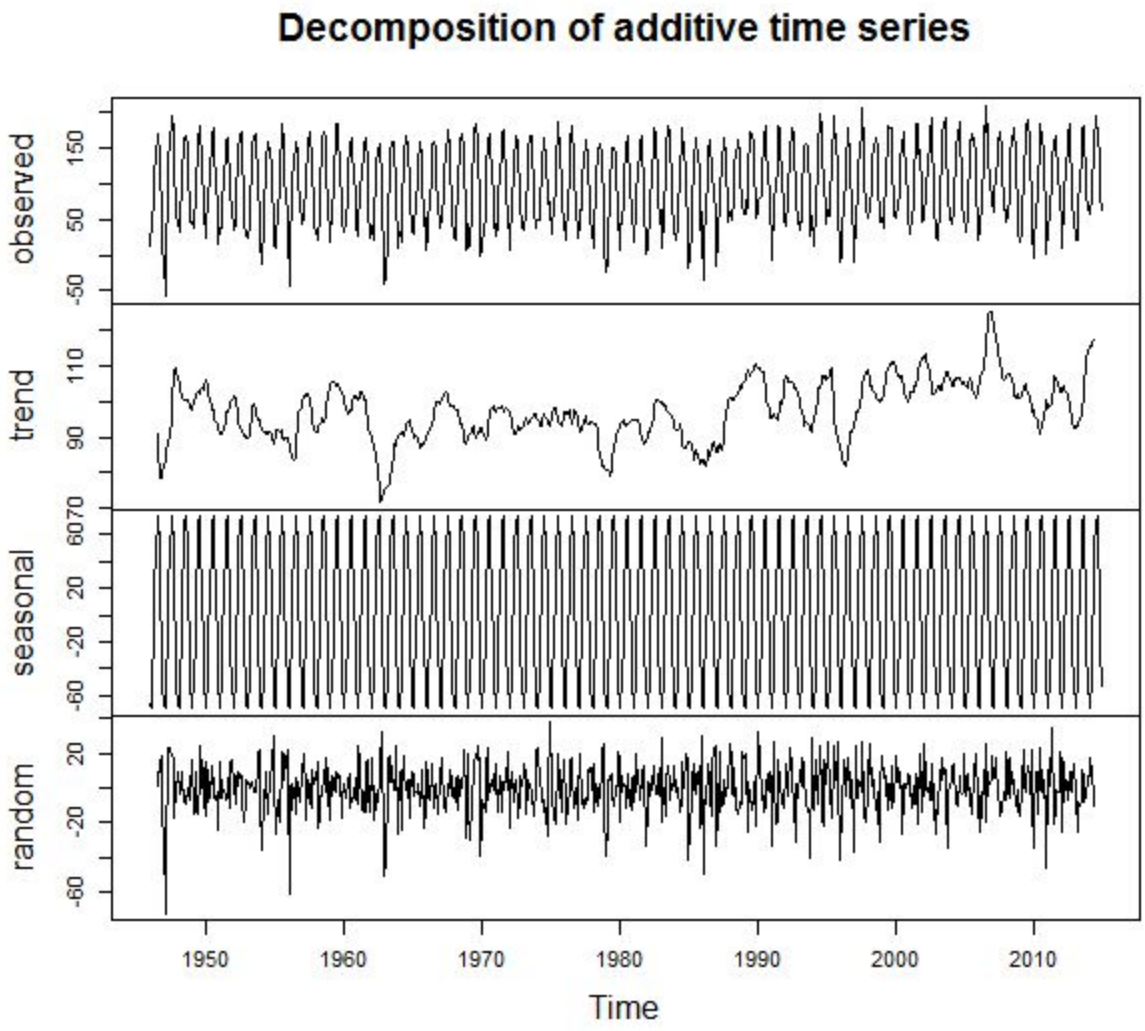
The R-Code

```
library(tidyverse)
library(forecast)

setwd("X:/My Desktop/Statistic/assignment6/assignment6")

df <- read.csv(file = "Pr_20May1(1).csv", header = TRUE)

#rename the column
names(df)[1]<-"temperature"

#create a squence month from 1946 to december 2014
df$t = seq(from = as.Date("1946-01-01"), to = as.Date("2014-12-31"), by="1
month")

#time series
t_ser <- ts(df$temperature, start = c(1946, 1),end = c(2014, 12), freq = 12)

#decompose the time series
output = decompose(t_ser)

#plot the time series decompostion
plot(output)
```

What frequency do you choose for the time series object?

I use the frequency at 12 since we want to decompose the data point into monthly data with a yearly cycle.

Show the plot and give an interpretation of all elements of the plot.



Interpretation of results:

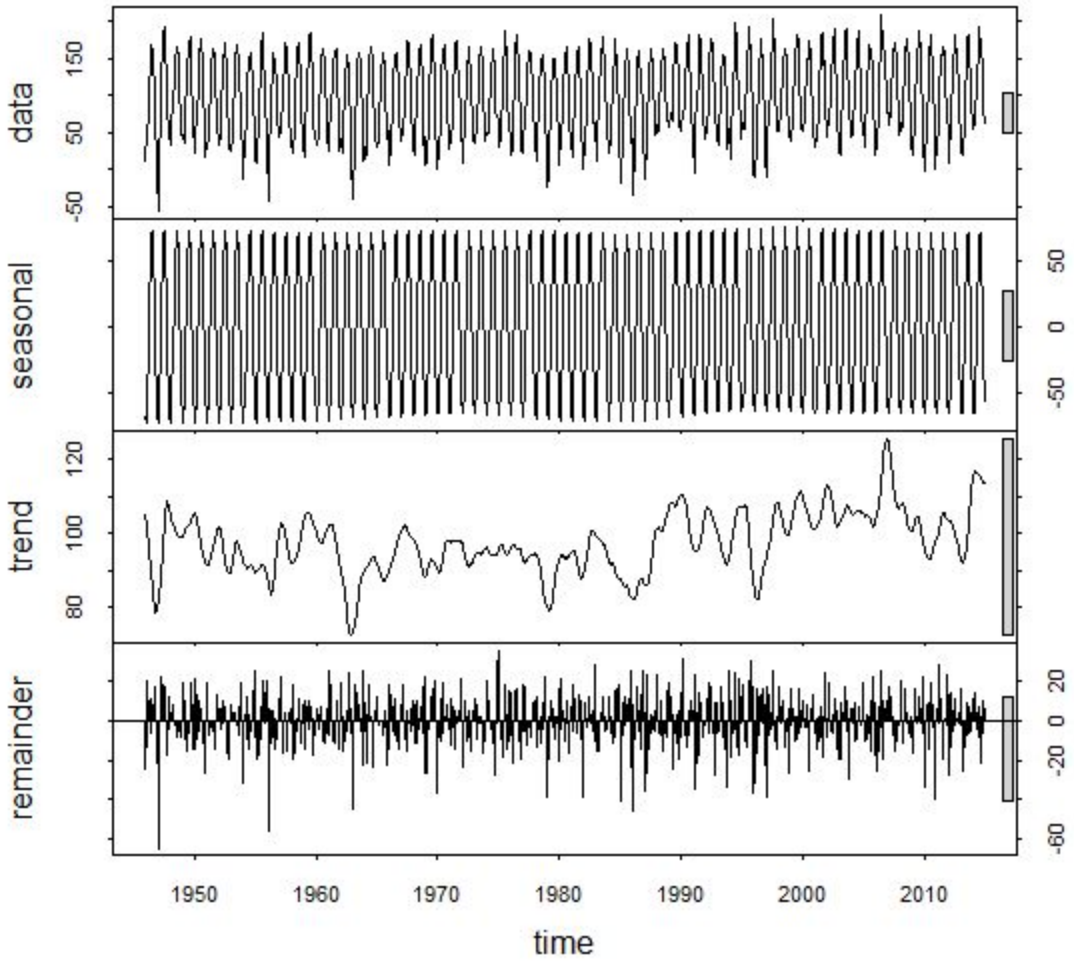
- **Observed** shows the observed value (the actual value)
- **Trend** shows the overall trend or the underlying trend of the metrics.
- **Seasonal** shows the seasonal variation or pattern that repeats with fixed period of time.
- **Random** shows the random noise or the residuals of the time series after allocation into the seasonal and trend time series.

6. **Decompose the time series using stl() decomposition. Set the parameters "t.window" so that you get a smoother trend.**

The R-Code

```
library(tidyverse)
library(forecast)

setwd("X:/My Desktop/Statistic/assignment6/assignment6")

df <- read.csv(file = "Pr_20May1(1).csv", header = TRUE)

#rename the column
names(df)[1]<-"temperature"

#create a squence month from 1946 to december 2014
df$t = seq(from = as.Date("1946-01-01"), to = as.Date("2014-12-31"), by="1 month")

#time series
t_ser <- ts(df$temperature, start = c(1946, 1),end = c(2014, 12), freq = 12)

#decompose the time series using stl
output = stl(t_ser, s.window = 25, t.window = (1.5 * 12)/(1 - 1/12))

#plot the time series decompostion using stl
plot(output)
```

The output



Can you detect a "warming" trend?

Yes. We can easily detect the warming trend by looking at the gray bar on the right side of the trend which divides the window into 6 parts. The average temperature is 100, therefore the warming trend can be defined as the temperature above 100.

Why do you not see this so clearly in the classical decomposition?

It's because in the classical decomposition there is not any feature which is able to determine how smooth the trend should be as stl does. If the trend is smoother then we get more variability in the remainder. Therefore, analysing warming trend using classical decomposition will be quite difficult because it does not have the smoothing feature.

What do “s.window” and “t.window” represent?

- **S.window** is the span (in lags) of the loess window for seasonal extraction, which should be odd and at least 7. We can set a window over how many cycles the seasonal cycle can vary.
- **T.window is** the span (in lags) of the loess window for trend extraction. It determines determines how smooth the trend should be. If the trend is smoother then you get more variability in the remainder.

Explain the main difference to the classical decomposition

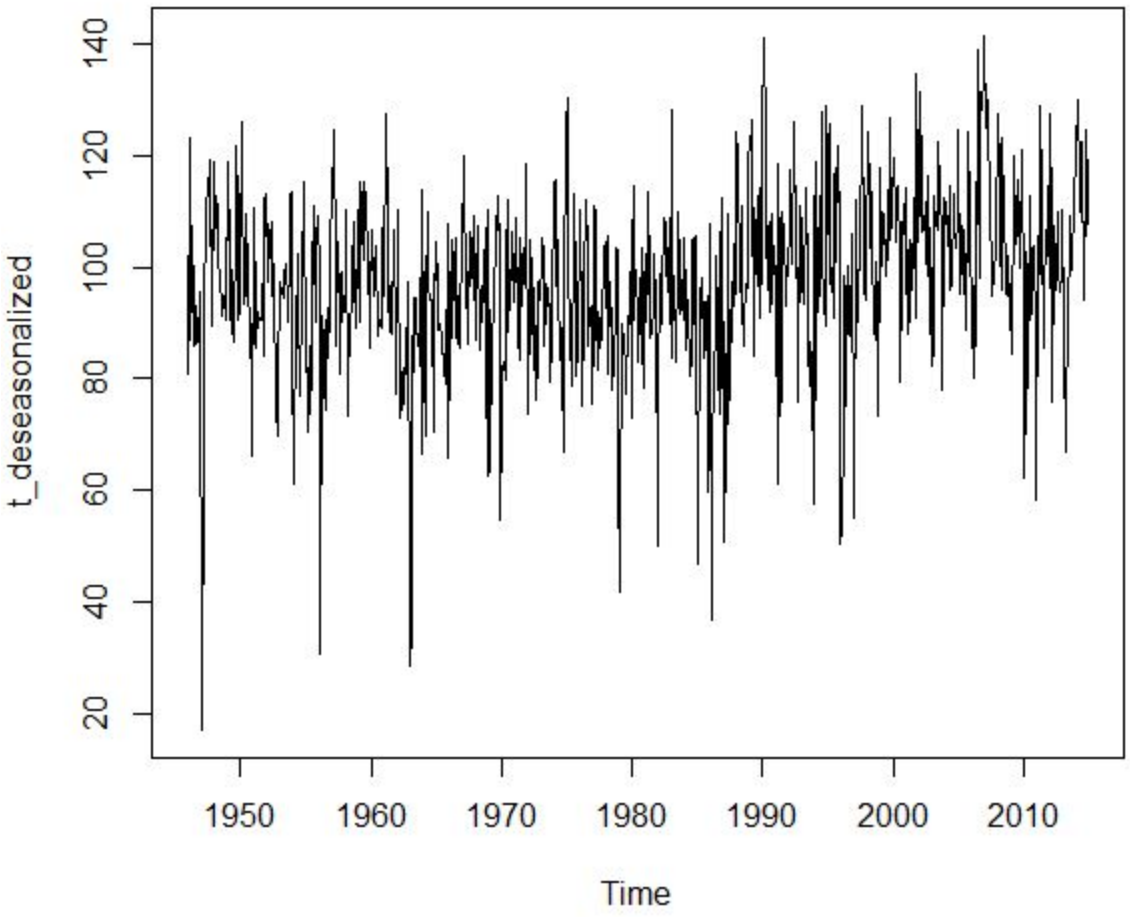
- The classical decomposition methods are unable to capture these seasonal changes over time while in STL the seasonal component is allowed to change over time and the rate of change can be controlled by the user.
- In STL the smoothness of the trend-cycle can also be controlled by the user while in classical decomposition it is not possible.
- The value of the time series in a small number of periods may be particularly unusual. The classical method is not robust to these kinds of unusual values. However, we this can be achieved in STL by allowing user to specify a robust decomposition.

7. Calculate and plot the de-seasonalized time series for the stl() decomposition.

The R-Code

```
library(tidyverse)
library(forecast)

setwd("X:/My Desktop/Statistic/assignment6/assignment6")
df <- read.csv(file = "Pr_20May1(1).csv", header = TRUE)

#rename the column
names(df)[1]<-"temperature"

#create a squence month from 1946 to december 2014
df$t = seq(from = as.Date("1946-01-01"), to = as.Date("2014-12-31"), by="1 month")

#time series
t_ser <- ts(df$temperature, start = c(1946, 1),end = c(2014, 12), freq = 12)

#decompose the time series using stl
output = stl(t_ser, s.window = 25, t.window = (1.5 * 12)/(1 - 1/12))

#Calculate and plot the de-seasonalized time series.
t_deseasonalized <- df$temperature - output$time.series[,"seasonal"]
plot(t_deseasonalized)
```

The output



What is shown in this plot?

This plot shows the monthly changes of the temperature per year. We can see from the plot the gradual increase of temperature in March. Then, it dramatically rises up in April until it reaches its highest peak in August. Starting from september, the temperature gradually decreases and then it dramatically decreases in November until it reaches its lowest peak in February in the following year.

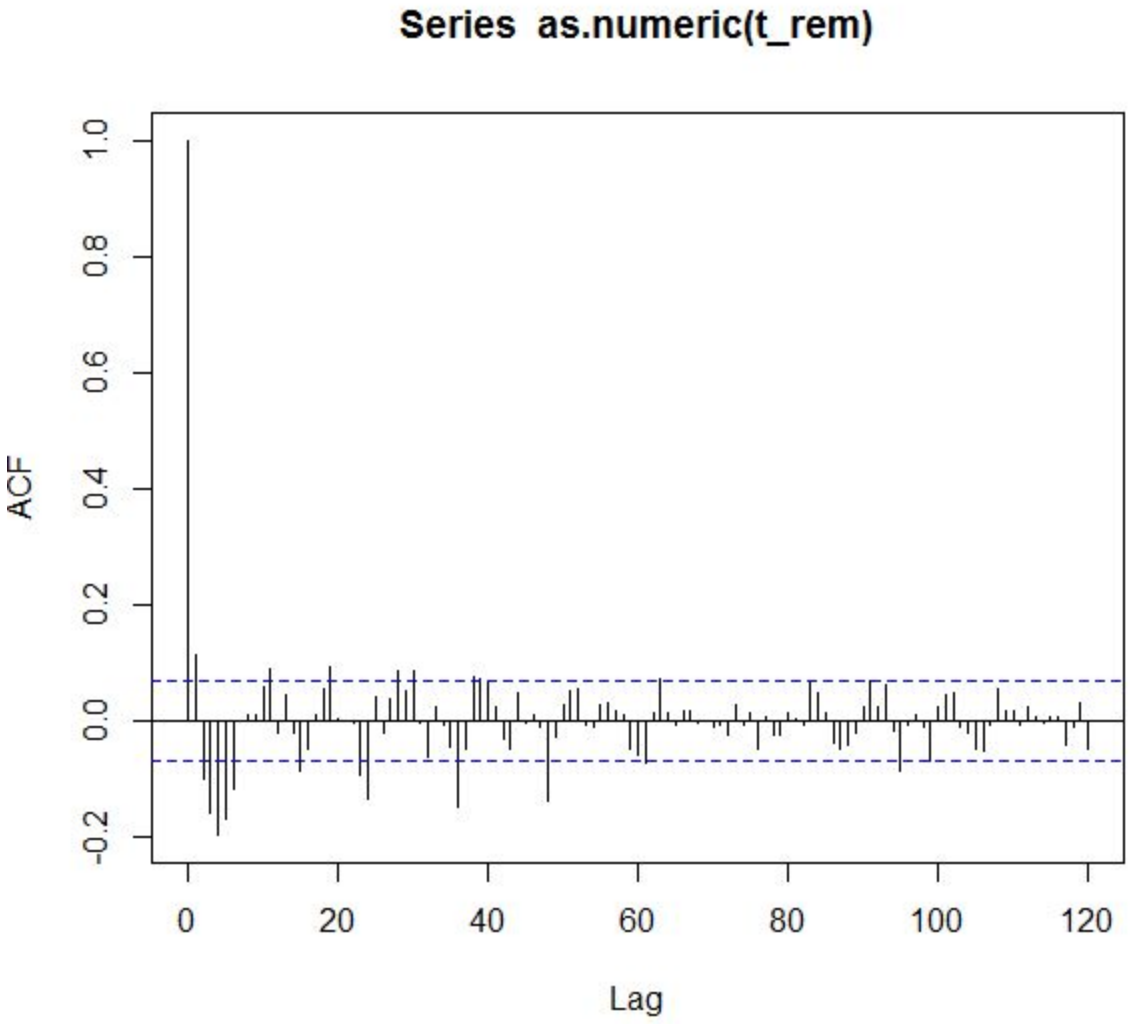
Apart from the different values of temperature per year, the pattern is still the same and it is repeated over and over again each year.

8. **Calculate the ACF of the remainder of the stl() decomposition up to a lag.max of 120.**

The R-Code

```
library(tidyverse)
library(forecast)

setwd("X:/My Desktop/Statistic/assignment6/assignment6")
df <- read.csv(file = "Pr_20May1(1).csv", header = TRUE)

#rename the column
names(df)[1]<-"temperature"

#create a squence month from 1946 to december 2014
df$t = seq(from = as.Date("1946-01-01"), to = as.Date("2014-12-31"), by="1 month")

#time series
t_ser <- ts(df$temperature, start = c(1946, 1),end = c(2014, 12), freq = 12)

#decompose the time series using stl
output = stl(t_ser, s.window = 25, t.window = (1.5 * 12)/(1 - 1/12))

#Calculate the ACF of the remainder of the stl() decomposition up to a lag.max  of 120.
t_rem <- output$time.series[,"remainder"]
acf_rem <-acf(as.numeric(t_rem), type = "correlation", lag.max = 120)
```

The output



Is it completely random, or does it still contain information? If so, how would you interpret this?

The plot shows that there is an indication of weak correlation at lag 1. The ACF value of lag 1 is very high and this sharply reduces afterwards. In my opinion, the value at leg 1 is completely random since the information it's not carried on another lag. This would be considered as outlier.

However, even though the rest of the ACF values in between lag 2 and 7 are much higher than the values in other lags, in my opinion they are not completely random because the values still correlates with another lag.