Problem 4

HW3

Hanyuan Chi(chixx105), Zhi Shen(shenx704)

February 22, 2017

```
suppressPackageStartupMessages({
  library(TSA)
  library(ggplot2)
  library(dplyr)
  library(forecast)
})
```

Time Series Decomposition

General Requirements

- Please do not change the path in `readRDS()`, your solutions will be automatically run by the bot and the bot will not have the folders that you have.
- Please review the resulting PDF and make sure that all code fits into the page. If you have lines of code that run outside of the page limits we will deduct points for incorrect formatting as it makes it unnecessarily hard to grade.
- Please avoid using esoteric R packages. We have already discovered some that generate arima models incorrectly. Stick to tried and true packages: base R, `forecast`, `TSA`, `zoo`, `xts`.

Problem Description

This problem is inspired by my own first encounter with time series decomposition (back at Samsung Electronics more than a decade ago).

A fellow engineer was working on GPS navigation devices and what was really curious to me was that the *Signal-to-Noise Ratio (SNR)* for GPS by design is *negative* meaning that the ambient radio noise is stronger than signal, in fact way stronger! Yet the device works!

As a human looking at this type of data, it is impossible to spot any patterns in it – the time series looks like a totally normal white noise. However, with the use of the math, the engineers are able to recover the faint signal from the satellite and remove all the noise.

For this problem, we will look at the version of that problem in the time domain. The key observations that you need to use here:

- the ambient radio noise is white noise
- the satellite sends the same (or similar) thing over and over again.

Given that, you can theoretically recover any signal no matter how faint from any levels of noise by repeating it enough times. In other words, high levels of noise impact the speed of data transfer rather than the possibility.

For this problem, please load the noisy signal data from file `problem4.Rds`

```r
problem4 <- readRDS("problem4.Rds") # Please do not change this line
```

This file contains a (simulated) noisy signal. Note that the data is generated such that:

- Var(Signal) = 1
- Var(Noise) = 100

$$\text{SNR} = 10 \cdot \log\left(\frac{\text{Var(Signal)}}{\text{Var(Noise)}}\right) \text{ decibel}$$

In other words, the noise is 100x more powerful than the signal and SNR is negative -20 decibel. Don't try to spot the signal in the raw time series - you will not be able to.

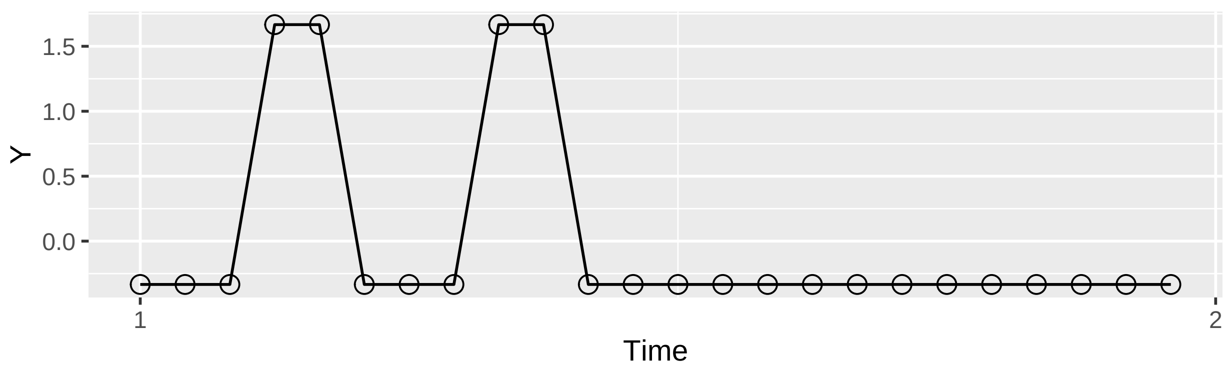
The signal's seasonality period is 24 (which means that the true signal is repeated by the satellite every 24 observations).

The true signal that is sent by the satellite is a sequence of pulses that look somewhat like a plot below. These pulses can be used to represent 0s and 1s. As an example, the plot contains 2 pulses.

```r
signal <- c(0,0,0,2,2,0,0,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0)
signal <- signal - mean(signal)

Y <- ts( signal, frequency = length(signal))

autoplot(Y) + geom_point(size=3, shape=1)
```



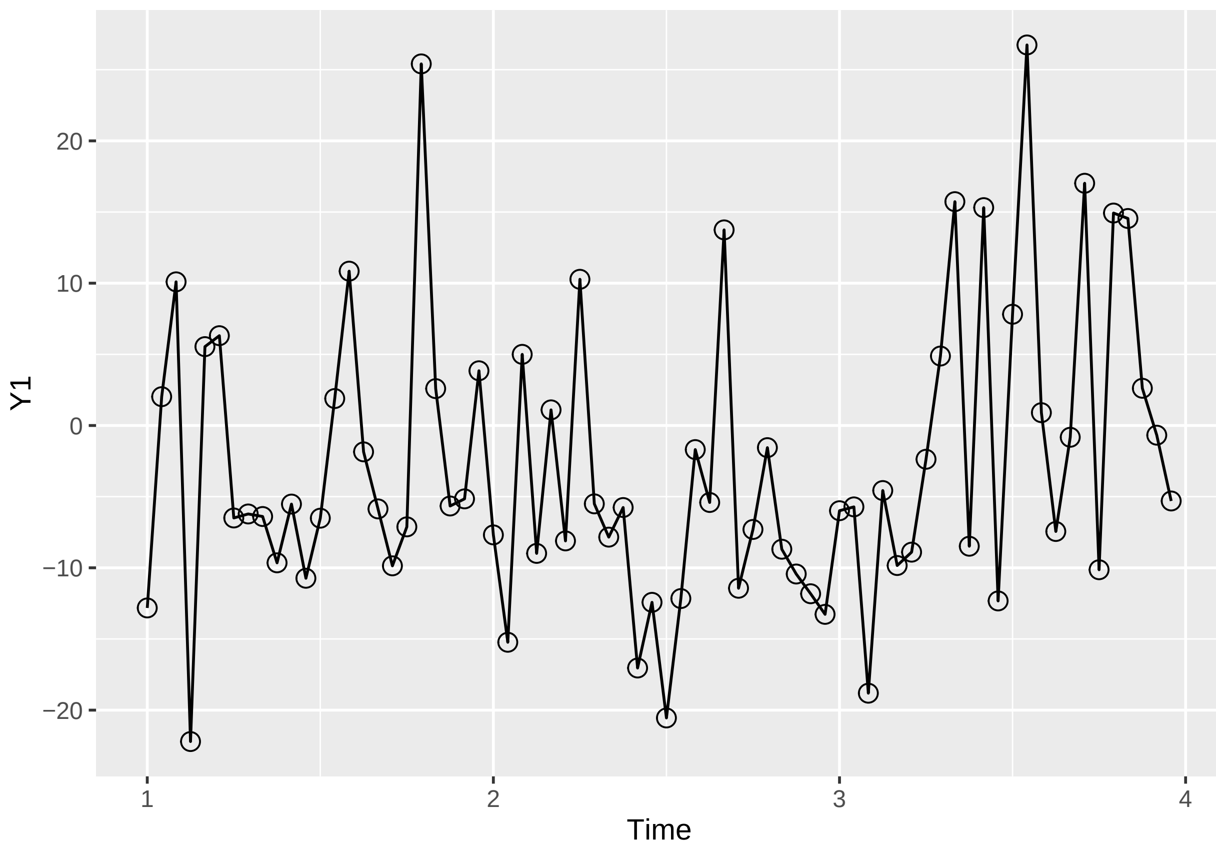
Question 1

Please plot the first 72 observations from the noisy signal data in `problem4`.

The true signal has been repeated 3 times by that time but you can't see it – it is completely overwhelmed by the background noise (the signal's scale is around ±1, while the noise scale is around ±10)

```r
# Plot here
Y1 <- ts( head(problem4,72), frequency = frequency(problem4))

autoplot(Y1) + geom_point(size=3, shape=1)
```



Q2

Please figure out how to remove all the (Gaussian) white noise and plot the signal (sequence of pulses) that you recovered. The true signal has been repeated so many times in `problem4` that it should be very easy to recover it.

For Q2, please do it by averaging "manually" and without using any time decomposition functions from the forecast package like `decompose`, `ma` or `stl`.

Note:

- your plot may not look as clean as the sample signal that I plotted above but the pulses will be very clear and visible nevertheless.

Hint:

- Use `cycle(x)` function to recover the season for each value
- In base R, you can use `aggregate` function with a formula to compute the means.
- In `dplyr`, you can use `group_by` and `summarise`.

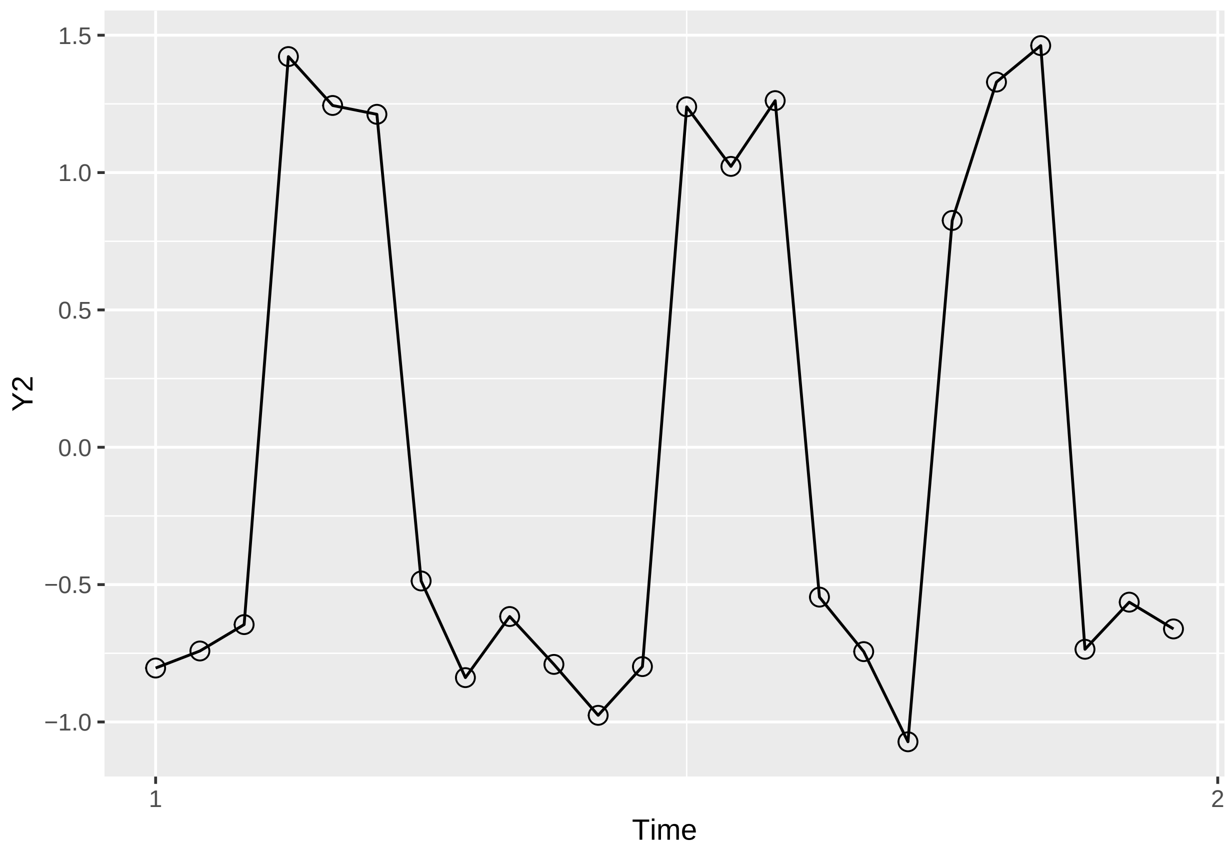
Output:

- please produce a vector `q2_means` of length 24 that contains the recovered signal (in other words, the seasonal means)

```
# Please write your code here
q2_means <- rep(NA, frequency(problem4))
```

```
p4_df <- data.frame(sig = problem4, pos = cycle(problem4))
p4_df <- aggregate(p4_df, by=list(p4_df$pos), FUN=mean)
q2_means <- p4_df$sig

# plot signal to check pulses
Y2 <- q2_means - mean(q2_means)
Y2 <- ts(Y2, frequency = frequency(problem4))
autoplot(Y2) + geom_point(size=3, shape=1)
```



Q3

As you recovered the true signal, how many pulses are there in one time window of length 24 observations?

Output:

- please a numeric value `q3_num_pulses` that contains the number of pulses that you saw on the plot.

```
# Please write down your answer here:
q3_num_pulses <- 3
```

Note:

- for the sample signal that I generated and showed above the correct answer would have been 2 as there are 2 pulses in the time window.
- please note that `forecast` package includes automated functions that would do time series decomposition for you such as `decompose`, `ma` or `stl`. For Q4, you shouldn't use them.

Take aways:

- Don't be afraid of white noise. In the world of randomness, white noise is a friend not an enemy.
- Be afraid of non-white noise. For GPS engineers, it is not the strong ambient noise that is the main issue, it is the faint correlated one – all these faint little reflections of the GPS signal from the nearby skyscrapers and buildings — this noise is strongly autocorrelated with the original signal and thus, cannot be cancelled out so easily. That's why your GPS tends to misbehave when you are in the middle of a downtown surrounded by the radio-reflective metal skyscrapers.