STAT 443: Lab 1

Derek Situ (62222807)

17 January, 2022

Question 1

(a)

```
dat <- read.csv("LakeLevels.csv", header = TRUE)
head(dat, 10)
```

```
##         Date LakeLevel
## 1   1/1/2007   3732.65
## 2   1/2/2007   3732.65
## 3   1/3/2007   3732.65
## 4   1/4/2007   3732.64
## 5   1/5/2007   3732.64
## 6   1/6/2007   3732.64
## 7   1/7/2007   3732.64
## 8   1/8/2007   3732.64
## 9   1/9/2007   3732.64
## 10 1/10/2007   3732.64
```

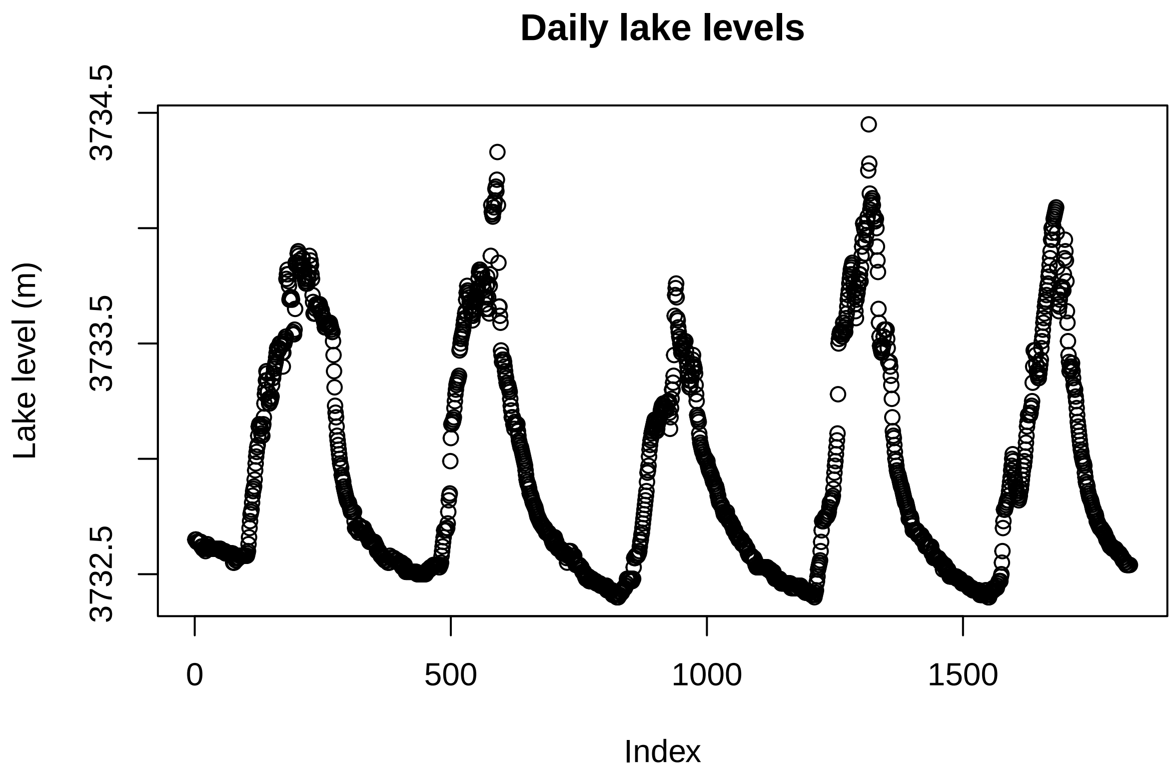
```
tail(dat, 10)
```

```
##            Date LakeLevel
## 1817 12/22/2011   3732.55
## 1818 12/23/2011   3732.55
## 1819 12/24/2011   3732.54
## 1820 12/25/2011   3732.54
## 1821 12/26/2011   3732.54
## 1822 12/27/2011   3732.54
## 1823 12/28/2011   3732.54
## 1824 12/29/2011   3732.54
## 1825 12/30/2011   3732.54
## 1826 12/31/2011   3732.54
```

```
names(dat)
```

```
## [1] "Date"      "LakeLevel"
```

```
plot(dat$LakeLevel, ylab="Lake level (m)", main="Daily lake levels")
```



This scatter plot is not exactly how we want to represent our time series data because the x-axis is not very meaningful as simply an observation index. It would be more helpful if the x-axis described a time scale. Also, we would like to join our observations with a line to approximate a continuous time process.

(b)

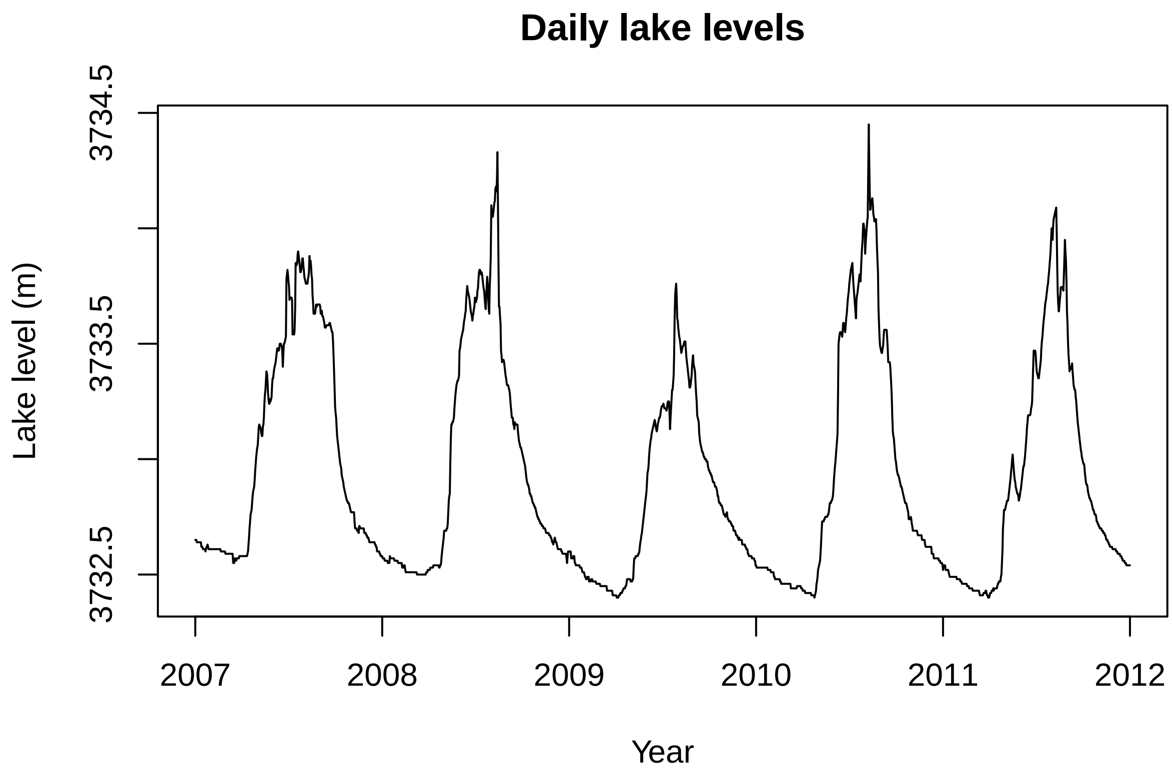
```r
# check if dat is a ts object
is.ts(dat)
```

```
## [1] FALSE
```

```r
# create time series object
x <- ts(data = dat$LakeLevel, start=c(2007, 1), frequency=365)
```

(c)

```r
# plot
plot(x, xlab="Year", ylab="Lake level (m)", main="Daily lake levels")
```



This new plot now has an x-axis that describes the time scale. In the old plot, it was harder to tell what time each observation was connected with since there were only observation indices. Also, each observation is joined by a line, giving the appearance of a continuous time process.

This time series shows seasonal variation, with generally deeper lake levels in the middle of the year, and shallower lake levels at the start and ends of years.

Question 2

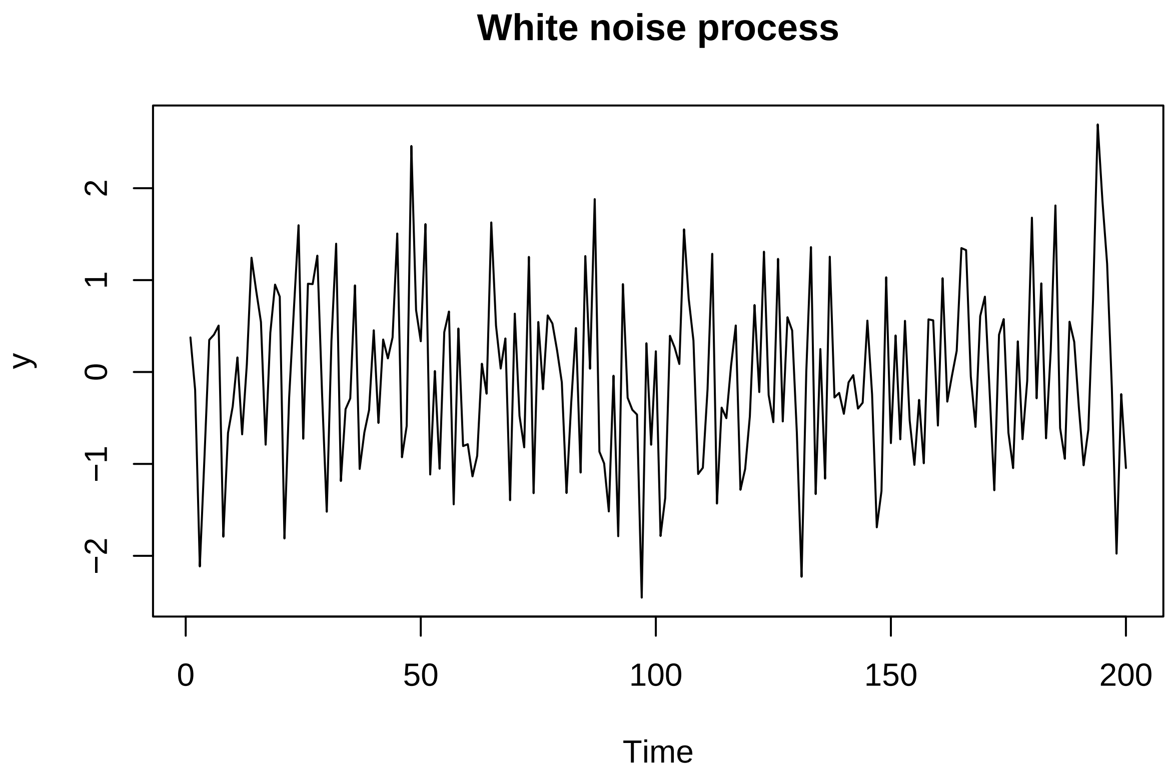
(a)

```r
# 200 independent observations
set.seed(2022)
white_noise <- rnorm(200)

# create ts
y <- ts(rnorm(200))
```

(b)

```r
# plot
plot(y, xlab="Time", main="White noise process")
```




```r
# This gives the amount of observations that are outside [-2, 2]
sum(abs(y) > 2)
```

```
## [1] 5
```

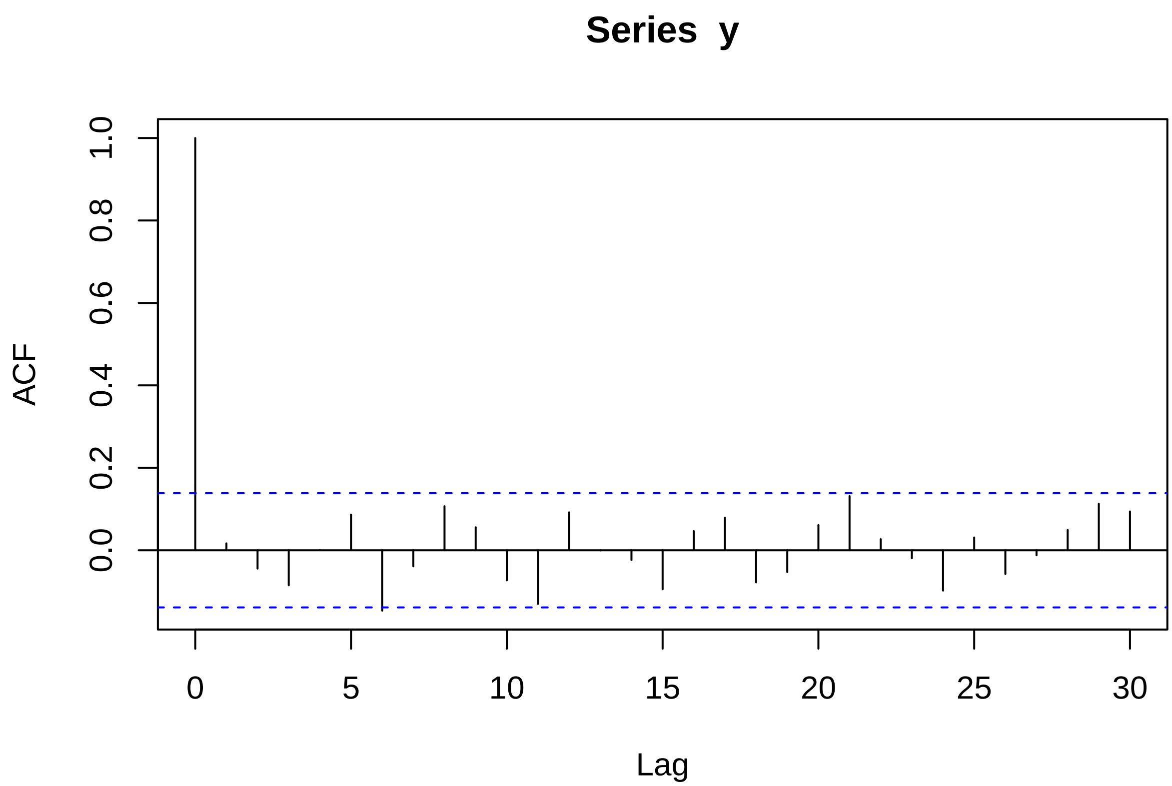
For a standard normal distribution, we expect about 5% of the observations to be outside the range [-2, 2]. Thus with 200 observations we expect 10 of them to be outside [-2, 2].

(c)

```r
# create the sample autocorrelation function
acf(y, lag.max=30, plot = F)
```

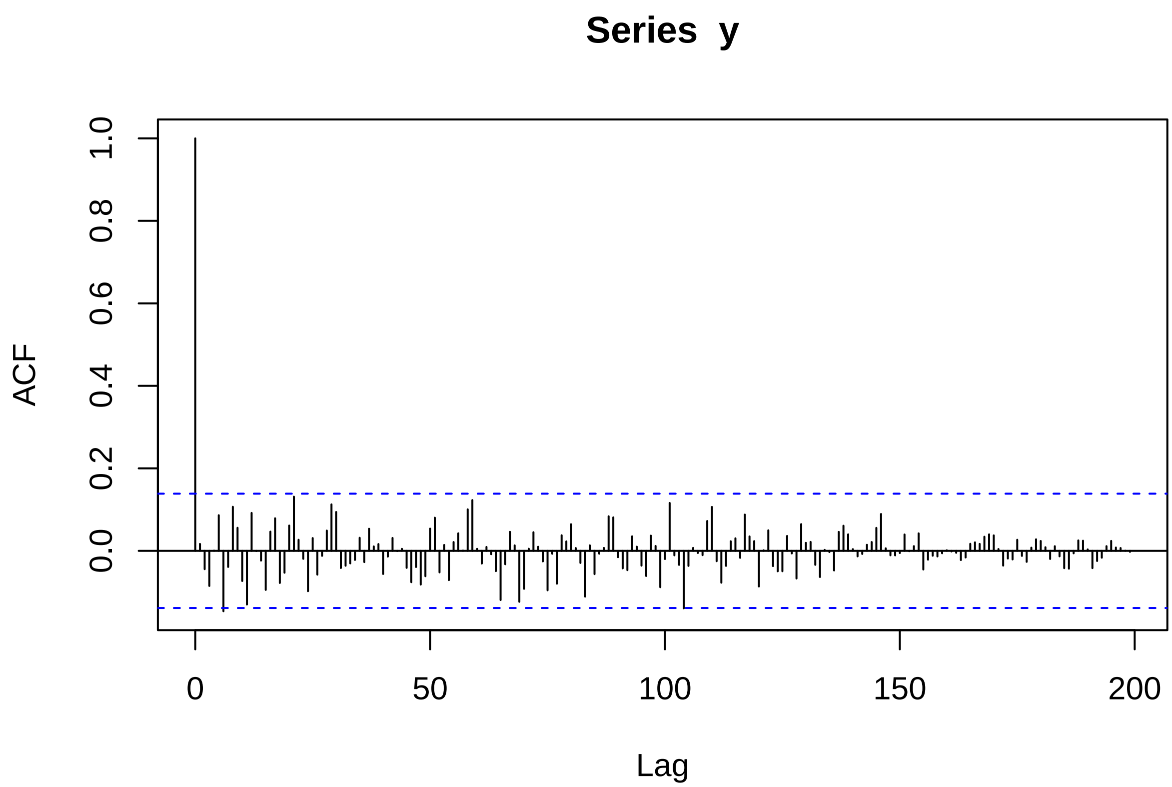
```
##
## Autocorrelations of series 'y', by lag
##
##      0      1      2      3      4      5      6      7      8      9     10
##  1.000  0.017 -0.045 -0.085  0.000  0.086 -0.146 -0.039  0.107  0.056 -0.073
##     11     12     13     14     15     16     17     18     19     20     21
## -0.130  0.092 -0.001 -0.024 -0.095  0.047  0.079 -0.078 -0.053  0.061  0.131
##     22     23     24     25     26     27     28     29     30
##  0.027 -0.019 -0.098  0.031 -0.058 -0.012  0.049  0.113  0.094
```

```r
acf(y, lag.max=30)
```



The sample acf oscillates around 0 with no obvious period and no clear trend in any direction. The sample correlation coefficients are very close to 0 (except at lag=0, since the coefficient is equal to 1 in such a case) since each observation was generated independently of other observations. In fact, about 95% of coefficients are within $\pm\frac{2}{\sqrt{200}}$. If we include up to lag=199 we see that the acf slowly decays to 0.

```r
acf(y, lag.max=199)
```


Series y
ACF
1.0
0.8
0.6
0.4
0.2
0.0
0
50
100
150
200
Lag