STAT 443: Lab 1

Matthew Kielar (Student #96793245)

2023-01-18

Question 1

(a)

```
dat <- read.csv("LakeLevels.csv", sep = ",", header = TRUE)
dat$Date <- mdy(gsub("/", "-", dat$Date))

head(dat, 10)
```

```
##          Date LakeLevel
## 1  2007-01-01   3732.65
## 2  2007-01-02   3732.65
## 3  2007-01-03   3732.65
## 4  2007-01-04   3732.64
## 5  2007-01-05   3732.64
## 6  2007-01-06   3732.64
## 7  2007-01-07   3732.64
## 8  2007-01-08   3732.64
## 9  2007-01-09   3732.64
## 10 2007-01-10   3732.64
```

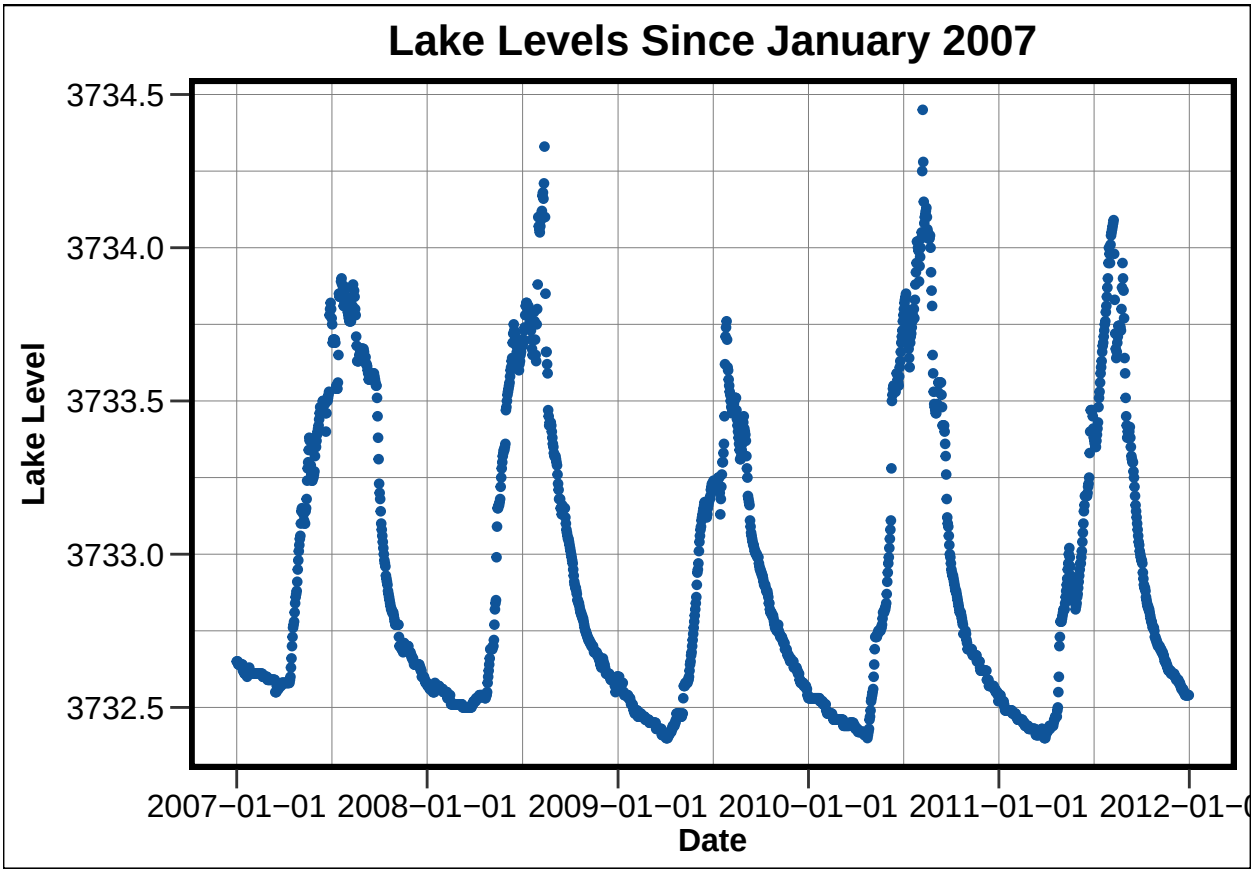
```
names(dat)
```

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

```
scatterplot <- ggplot(
  data = dat,
  mapping = aes(
    x = Date, y = LakeLevel)
) +
  geom_point(
    color = "#0F5499",
    size = 1.2
) +
  labs(
    x = "Date",
    y = "Lake Level",
    title = "Lake Levels Since January 2007"
) +
  scale_x_date(
```

```
    date_breaks = "1 year"
)

print(scatterplot)
```



Overall, this chart differs from the one I'd like since it displays the plotted points but doesn't connect them together in the way a classic lineplot does.

(b)

```
is.ts(dat)
```

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

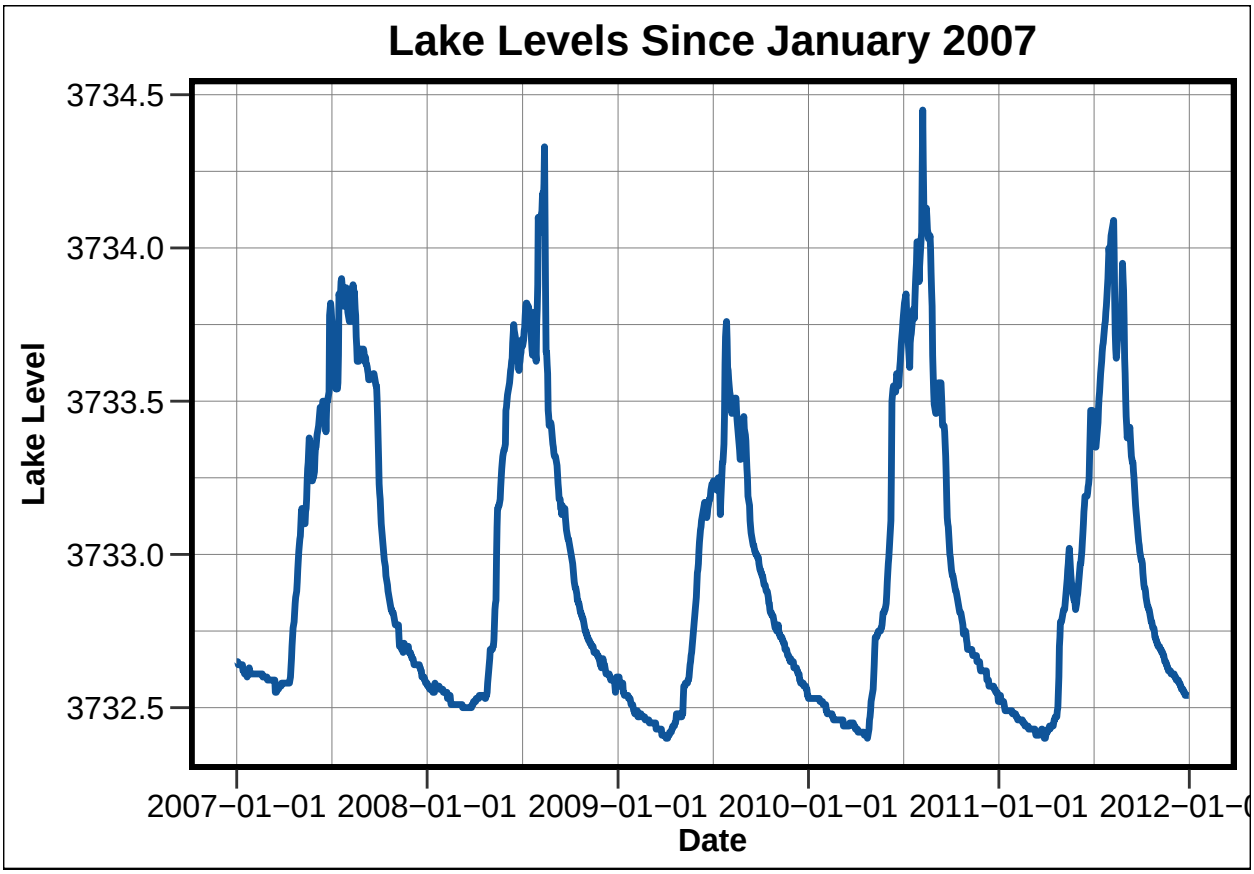
```
x <- ts(dat$LakeLevel, start = c(2007, 1, 1), frequency = 365)
```

(c)

```
lineplot <- ggplot(
  data = dat,
  mapping = aes(
    x = Date, y = LakeLevel
  )
```

```
) +
  geom_line(
    color = "#0F5499",
    linewidth = 1.2
) +
  labs(
    x = "Date",
    y = "Lake Level",
    title = "Lake Levels Since January 2007"
) +
  scale_x_date(
    date_breaks = "1 year"
)

print(lineplot)
```



Overall it looks like the series doesn't change much in the long run, but exhibits a lot of seasonality instead. Lake levels increase a lot during the summer months and then proceed to decrease again at the start of a new year.

Question 2

(a)

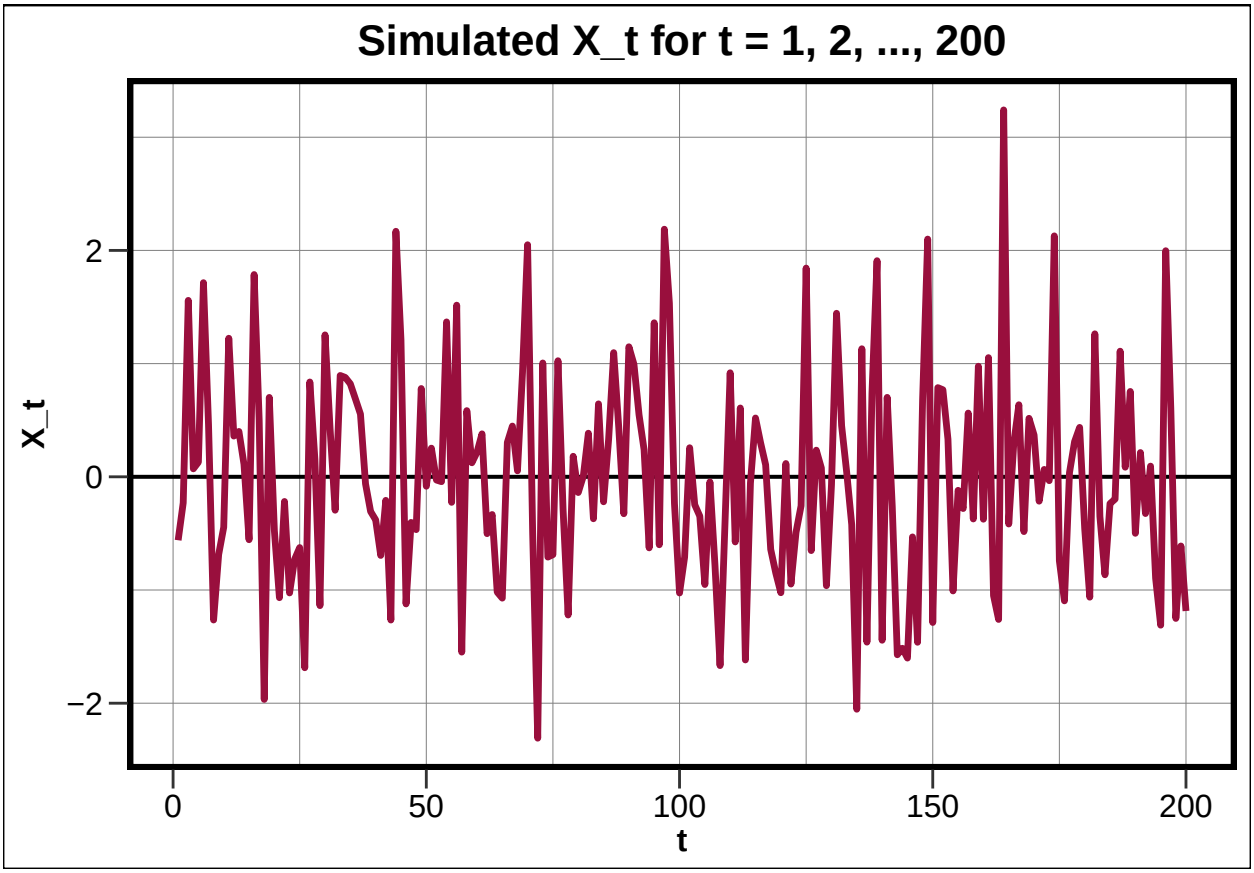
```
set.seed(123)
white_noise <- data.frame(
  t = seq(1, 200, length.out = 200),
  X_t = rnorm(n = 200, mean = 0, sd = 1)
)

white_noise_ts <- ts(white_noise$X_t, start = 1, end = 200)
```

(b)

```
simul_plot <- ggplot(
  data = white_noise,
  mapping = aes(
    x = t, y = X_t
  )
) +
  geom_hline(
    yintercept = 0,
    color = "#000000",
    linewidth = 0.7
) +
  geom_line(
    color = "#990F3D",
    linewidth = 1.2
) +
  labs(
    x = "t",
    y = "X_t",
    title = "Simulated X_t for t = 1, 2, ..., 200"
)

print(simul_plot)
```




```
nrow(white_noise[white_noise$X_t > 2, ])
```

```
## [1] 6
```

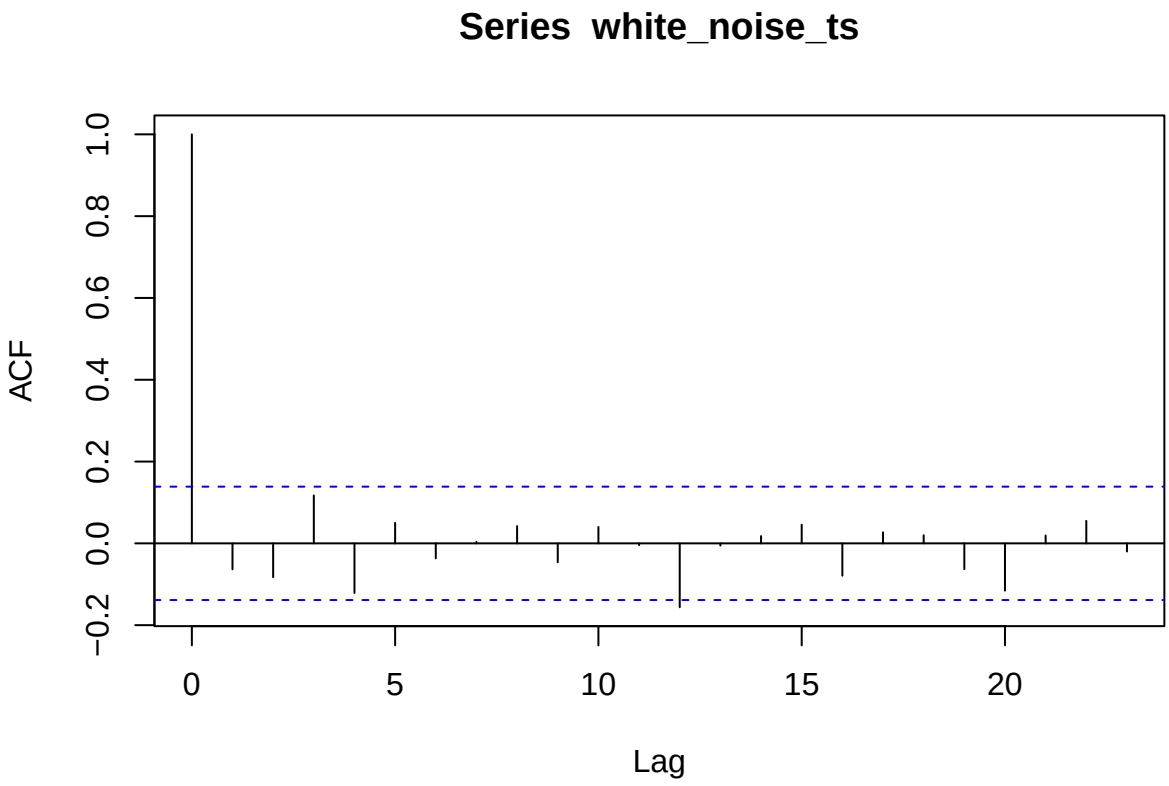
```
nrow(white_noise[white_noise$X_t < -2, ])
```

```
## [1] 2
```

So in total we have 8 observations that fall outside the interval [-2, 2]. By the empirical rule, we can expect 10 observations to take on values ± 2.

(c)

```
acf <- acf(white_noise_ts)
```



The sample acf values are quite low across all the lags, which intuitively makes sense since the data are observations drawn from i.i.d standard normal R.Vs. There are a few larger values, but those are likely larger observed values of X_t.

More information on R Markdown

This is an R Markdown document, which can be used as a template for STAT 443 labs and assignments. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

```
summary(cars)
```

```
##      speed           dist
##  Min.   : 4.0   Min.   :  2.00
##  1st Qu.:12.0   1st Qu.: 26.00
##  Median :15.0   Median : 36.00
##  Mean   :15.4   Mean   : 42.98
##  3rd Qu.:19.0   3rd Qu.: 56.00
##  Max.   :25.0   Max.   :120.00
```

Using the function *kable*, it produces a nicer table

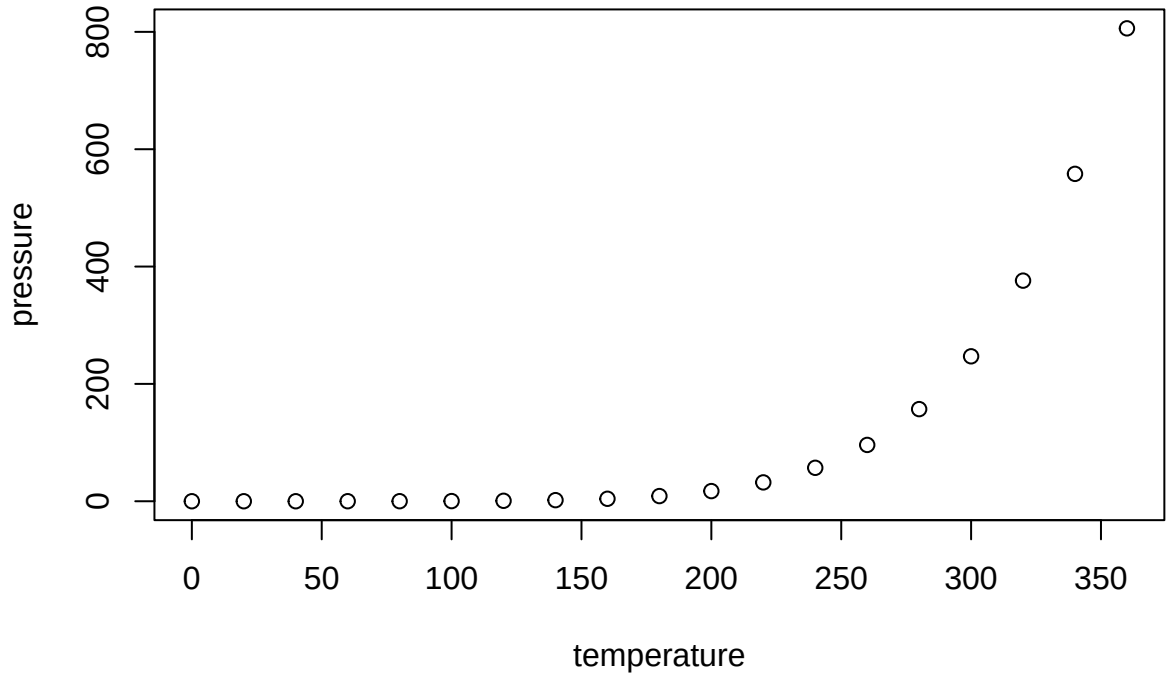
```
kable(summary(cars))
```

speed	dist
Min. : 4.0	Min. : 2.00
1st Qu.:12.0	1st Qu.: 26.00
Median :15.0	Median : 36.00
Mean :15.4	Mean : 42.98
3rd Qu.:19.0	3rd Qu.: 56.00
Max. :25.0	Max. :120.00

Including Plots

You can also embed plots, for example:

```
plot(pressure)
```



Note that specifying `echo = FALSE` parameter would prevent printing of the R code that generated the plot. This is something you may want to do for larger reports that would not require display of the R code.

You can also modify the size and alignment of the figure.

```
plot(pressure)
```

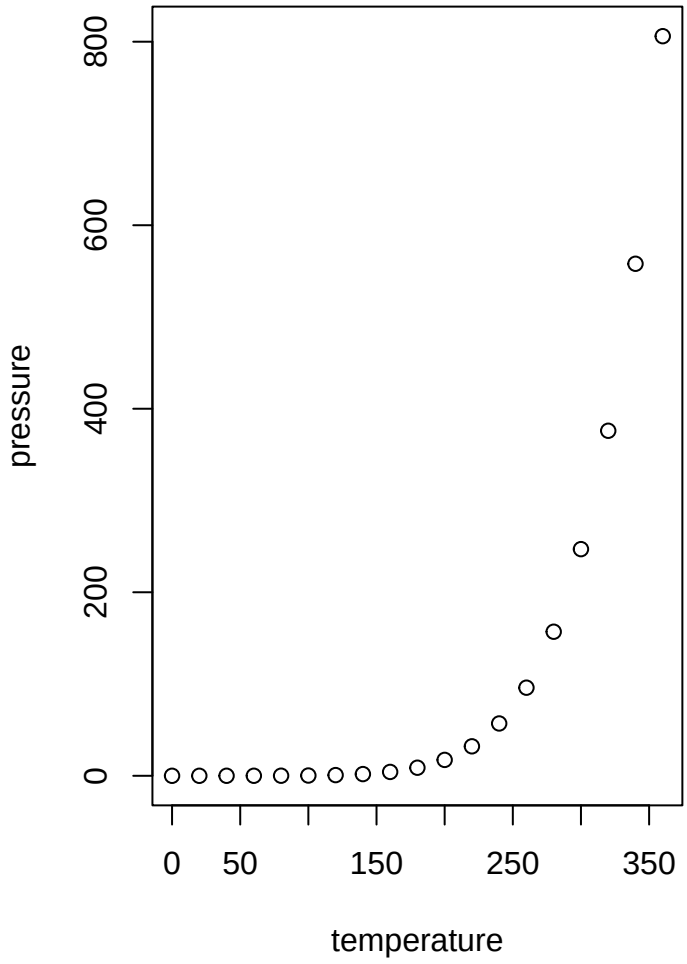


Figure 1: title