Lab 2: Trend Seasonality and Sample Autocorrelation

Questions

Question 1

Features of the data

Read the data into R using `read.csv()`.

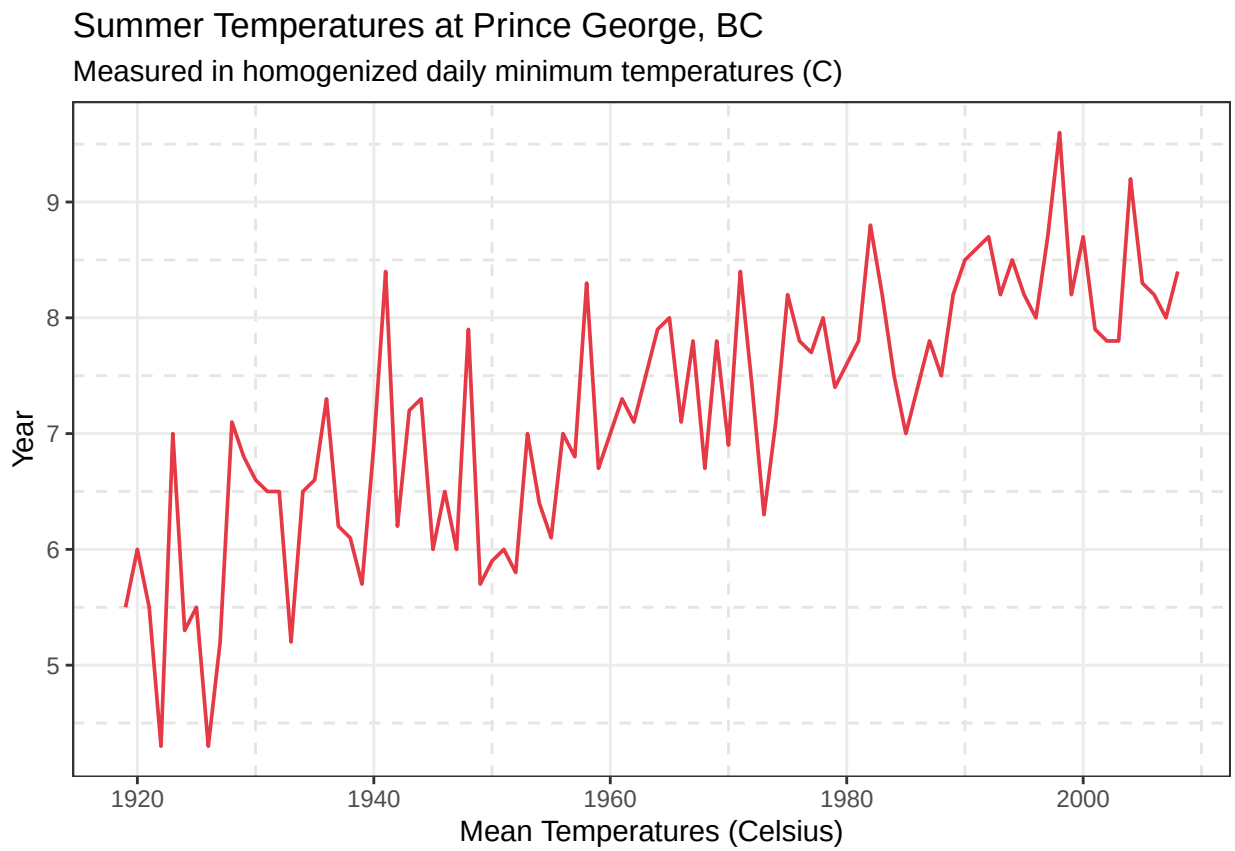
```
df1 = read.csv("dataTempPG.csv")
```

Extract the "Summer" column and convert it into a time series in R.

```
summer_data = df1$Summer

# the data are yearly starting in 1919
summer_series = ts(data = summer_data,
                   start = c(1919),
                   frequency = 1)
```

Plot the time series with appropriate labels.

```
p1data = fortify.zoo(summer_series)
p1 <- ggplot(p1data, aes(x = Index, y = summer_series)) +
  geom_line(color = "#E63946", linewidth = 0.65) +
  labs(
    title = "Summer Temperatures at Prince George, BC",
    subtitle = "Measured in homogenized daily minimum temperatures (C)",
    x = "Mean Temperatures (Celsius)",
    y = "Year"
  ) + theme_bw() +
  theme(panel.grid.minor = element_line(
    color = "grey90",
    linetype = "dashed",
    linewidth = 0.5
  ))
print(p1)
```



Comment on the main features of this series (e.g., seasonality, trend).

There doesn't appear to be much seasonality in the data. This makes sense, considering we are already isolating for a season (summer.) However, there does seem to be an increasing (positive) trend to the data.

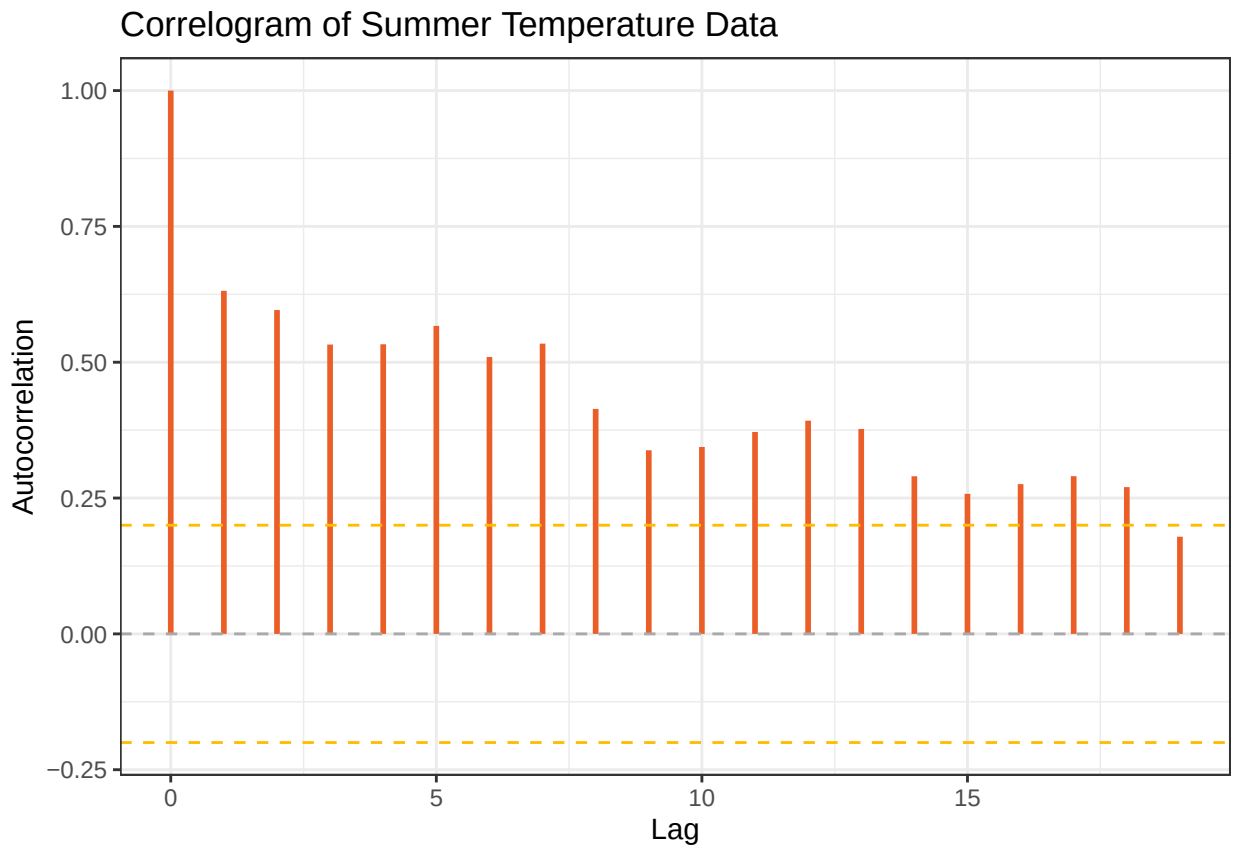
Sample Autocorrelation

Use `acf()` to create the autocorrelation function for the mean summer temperature data.

Here, we calculate and plot the ACF.

```
p2data = data.frame(
  h = 0:19,
  rh = acf(summer_series, plot = FALSE)$acf
)

p2 <- ggplot(p2data, aes(x = h, y = rh)) +
  geom_segment(aes(xend = h, yend = 0),
               color = "#eb5e28",
               linewidth = 1) +
  geom_hline(yintercept = 0.2, linetype = "dashed", col = "#ffbd00")+
  geom_hline(yintercept = -0.2, linetype = "dashed", col = "#ffbd00")+
  ylim(-0.2, 1)+
  geom_hline(yintercept = 0,
             linetype = "dashed",
             color = "darkgray") +
```

```
  labs(x = "Lag", y = "Autocorrelation",
       title = "Correlogram of Summer Temperature Data") +
  theme_bw()
print(p2)
```



Comment on the behavior of the sample autocorrelation function.

We can see from the sample acf that there appears to be positive serial correlation between the data points, declining slightly over time as a function of lag. As was discussed in lecture, this is likely indicative of a positive trend in the time series as a whole.

Smoothing

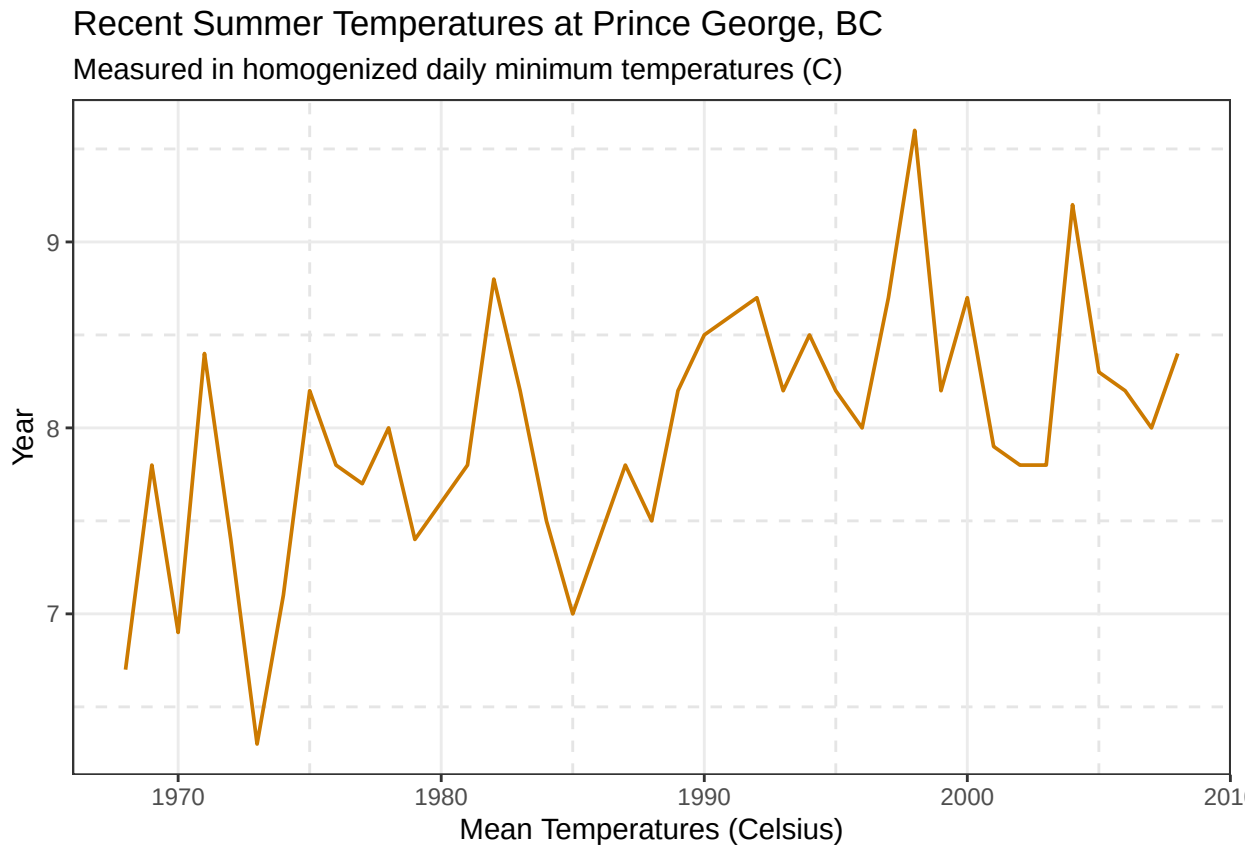
Use `window()` to extract the portion of the time series between 1968 and 2008.

```
subset = window(summer_series, start = 1968, end = 2008)
```

Plot this recent record of the data.

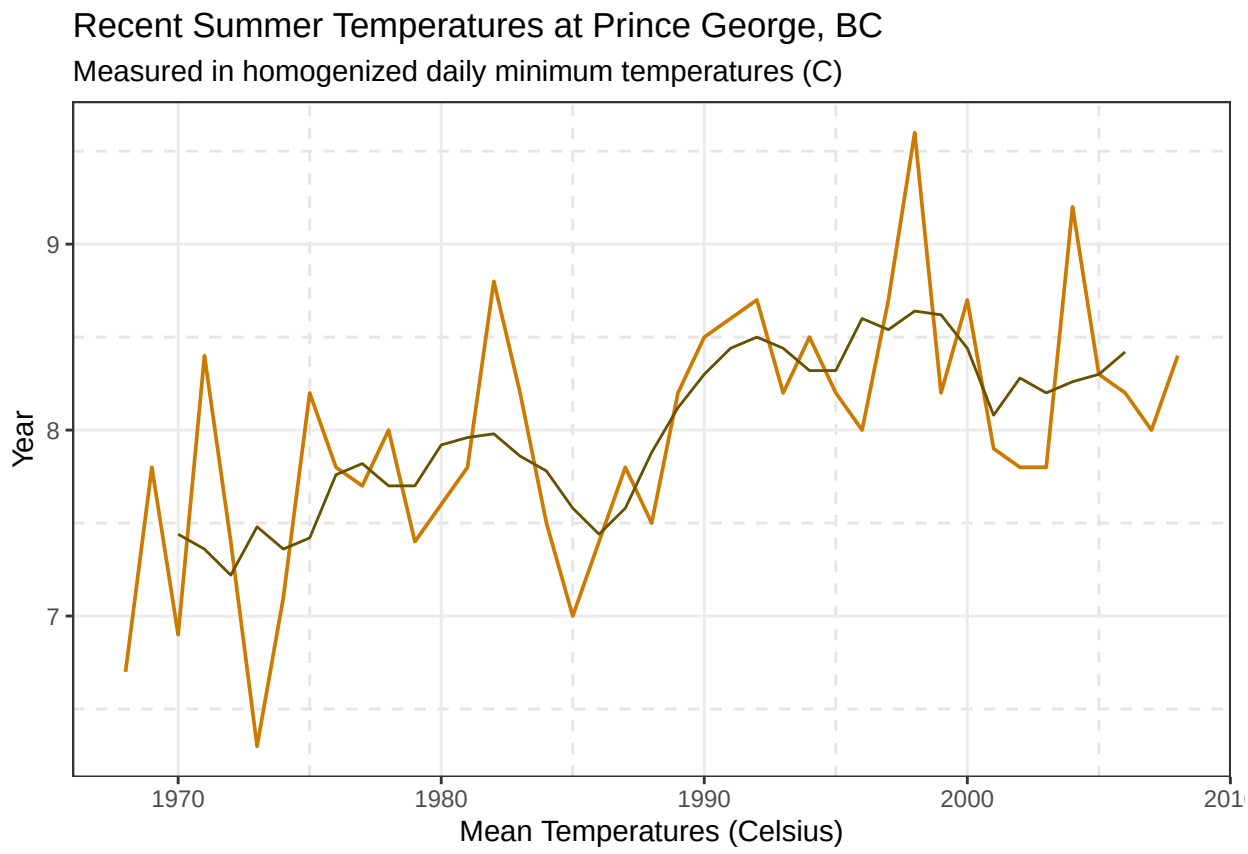
```
p3data <- fortify.zoo(subset)
p3 <- ggplot(p3data, aes(x = Index, y = subset)) +
  geom_line(color = "#cc7a00", linewidth = 0.65) +
  labs(
    title = "Recent Summer Temperatures at Prince George, BC",
    subtitle = "Measured in homogenized daily minimum temperatures (C)",
```

```
    x = "Mean Temperatures (Celsius)",
    y = "Year"
  ) + theme_bw() +
  theme(panel.grid.minor = element_line(
    color = "grey90",
    linetype = "dashed",
    linewidth = 0.5
  ))
print(p3)
```



Use `rollmean()` with a moving average parameter of k=5 to add the smoothed series to the plot.

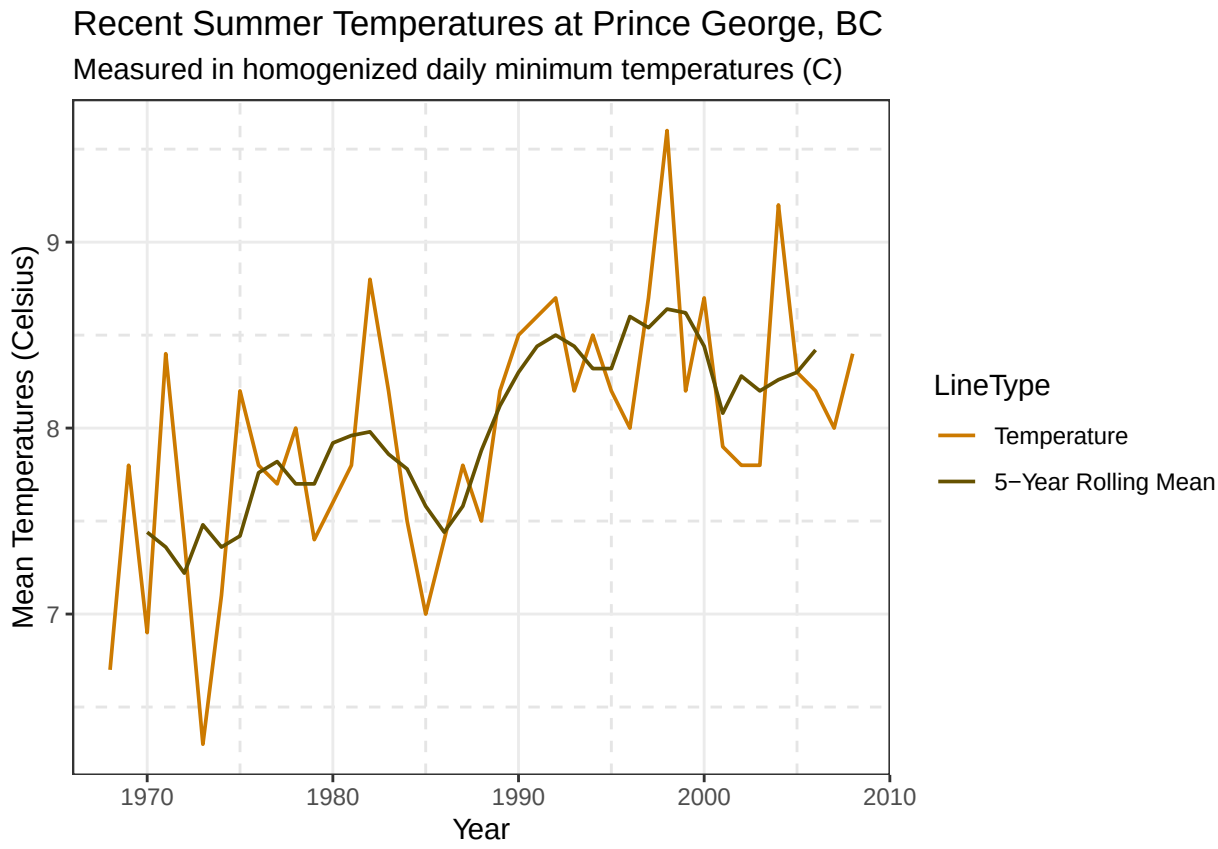
```
df_roll_mean = data.frame(RollingMean = rollmean(subset, k = 5, fill = NA),
                          Time = p3data$Index)
p3 = p3 + geom_line(
  data = df_roll_mean,
  aes(x = Time, y = RollingMean),
  color = "#665200",
  na.rm = TRUE
)
print(p3)
```



Add a legend to your plot.

```
p4df = data.frame(
  subset = p3data$subset,
  rollmea = rollmean(subset, k = 5, fill = NA),
  Index = p3data$Index
)
p4df_long <- melt(p4df, id.vars = "Index", variable.name = "LineType", value.name = "Value")

p4 <- ggplot(p4df_long, aes(x = Index, y = Value, color = LineType)) +
  geom_line(linewidth = 0.65) +
  labs(
    title = "Recent Summer Temperatures at Prince George, BC",
    subtitle = "Measured in homogenized daily minimum temperatures (C)",
    x = "Year",
    y = "Mean Temperatures (Celsius)"
  ) +
  theme_bw() +
  scale_color_manual(
    values = c("subset" = "#cc7a00", "rollmea" = "#665200"),
    labels = c("Temperature", "5-Year Rolling Mean")
  ) +
  theme(panel.grid.minor = element_line(
    color = "grey90",
    linetype = "dashed",
    linewidth = 0.5
```

```
  ))

print(p4)
```



Question 2

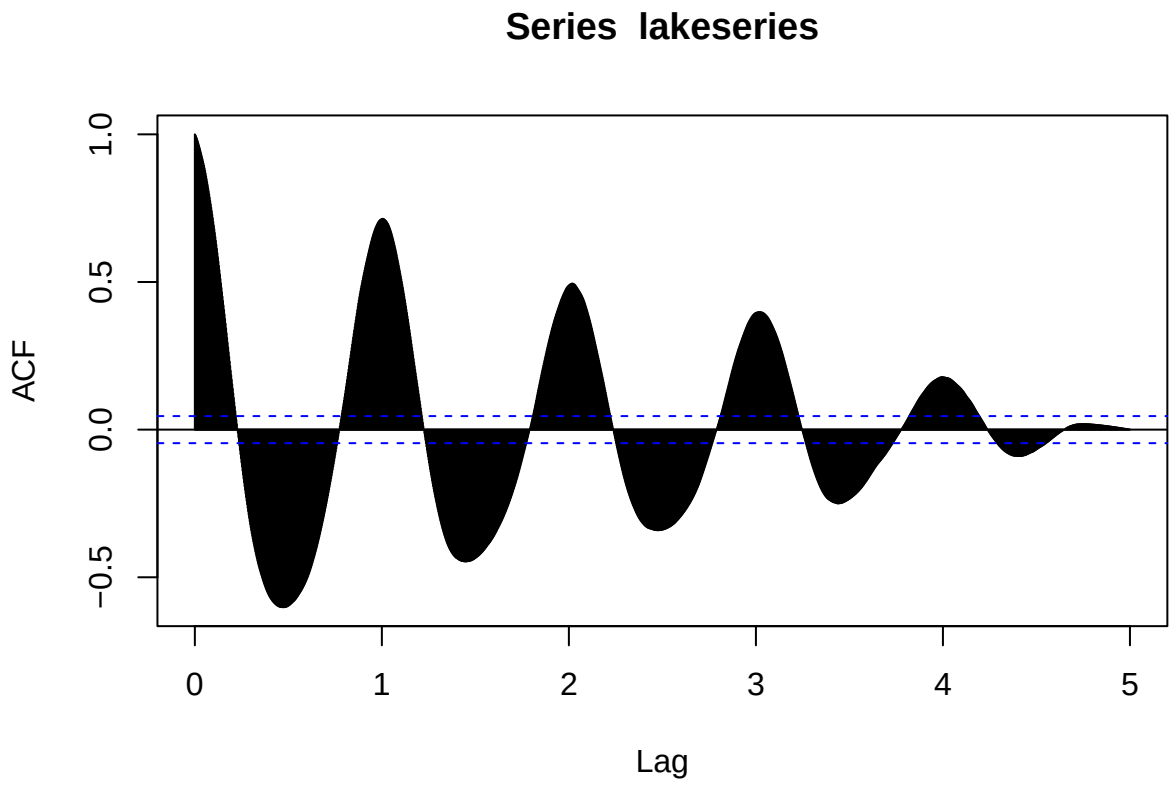
Sample Autocorrelation Function

Read the data into R and convert it into a time series.

```
df2 = read.csv("LakeLevels.csv")

lakeseries = ts(df2$LakeLevel,
                start = c(2007, 1),
                frequency = 365)
```

Use `acf()` to create the sample autocorrelation function for the lake levels.

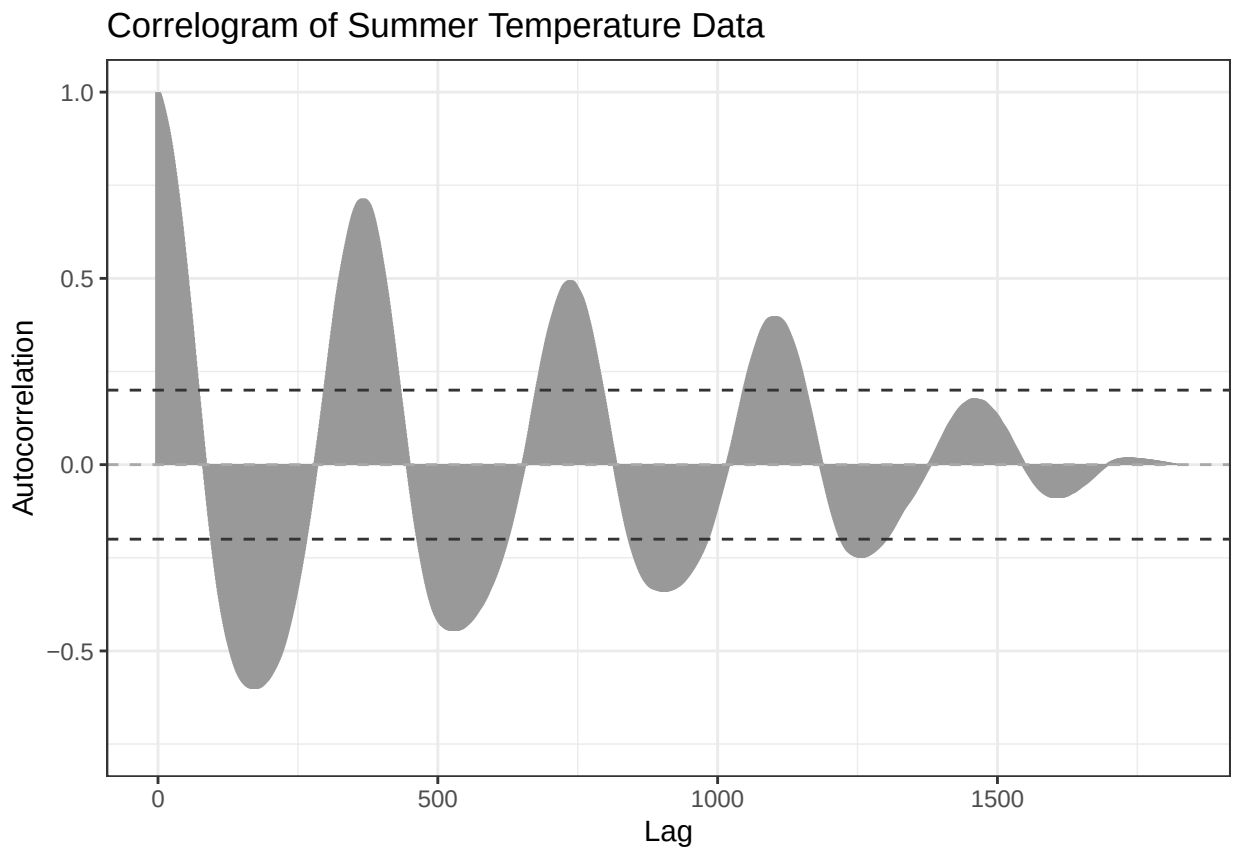
```
# acf(lakeseries, lag.max = nrow(df2))

p5data = data.frame(
  h = 0:(nrow(df2)-1),
  rh = acf(lakeseries, lag.max = nrow(df2))$acf
)
```




```
p5 <- ggplot(p5data, aes(x = h, y = rh)) +
  geom_segment(aes(xend = h, yend = 0),
               color = "grey60",
               size = 1) +
  geom_hline(yintercept = 0.2, linetype = "dashed", col = "grey20")+
  geom_hline(yintercept = -0.2, linetype = "dashed", col = "grey20")+
  ylim(-0.75, 1)+
  geom_hline(yintercept = 0,
             linetype = "dashed",
             color = "darkgray") +
  labs(x = "Lag", y = "Autocorrelation",
       title = "Correlogram of Summer Temperature Data") +
  theme_bw()
```

```
## Warning: Using ‘size‘ aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use ‘linewidth‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

```
print(p5)
```



Comment on the behavior of this autocorrelation function.

There is an obvious periodic pattern in the acf, that is declining over time. This makes sense considering that the original `lakeLevels` data has seasonality. We see this reflected in the autocorrelation function.

2. **Decomposition (Additive Model)**
    - Use `decompose()` to decompose the time series into trend, seasonal component, and error (additive model).
    - Plot the trend, seasonal component, and error.
3. **Decomposition (Loess Method)**
    - Use `stl()` with `s.window` set to "periodic" for decomposition.
    - Plot the trend, seasonal component, and error using the loess method.