Report6

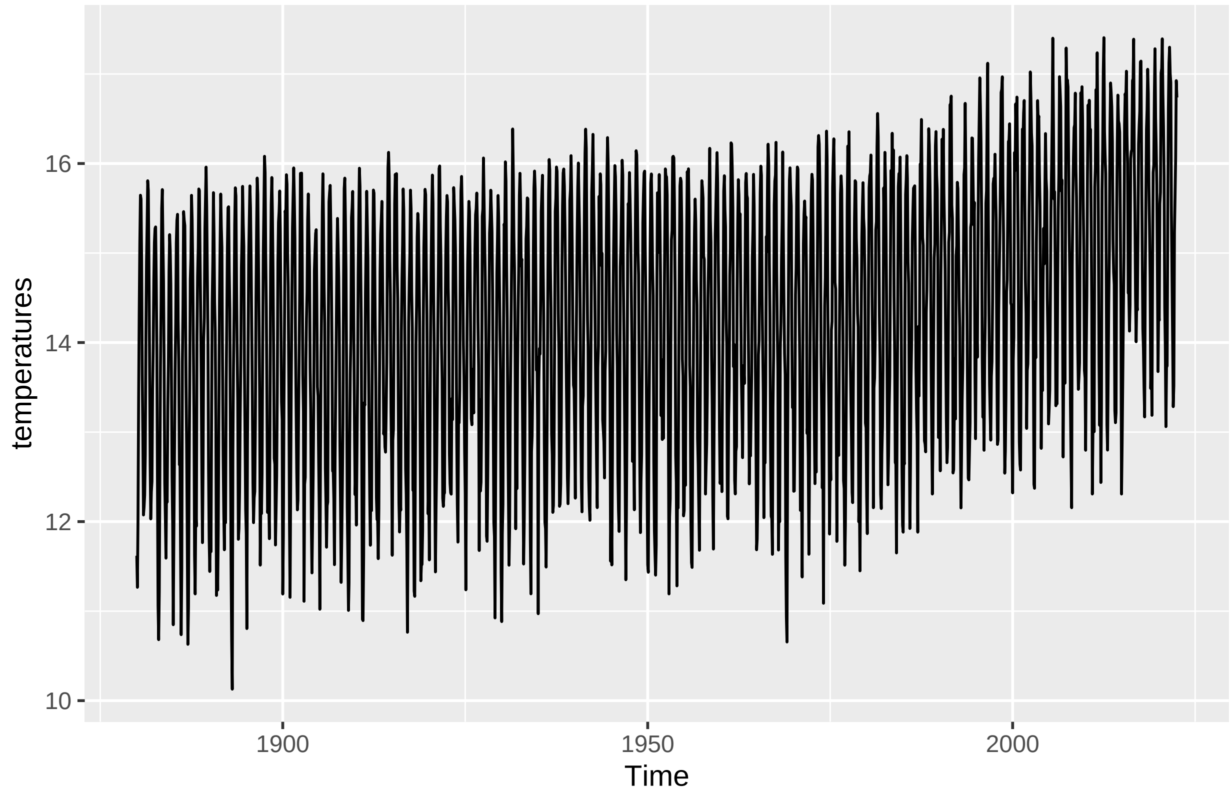
Khaled Hasan

2024-04-11

Temperature Data

```
GLOBALTEMPERATURE = read.csv(file = "C:\\Users/ss/Desktop/Time_series_Analysis/out2.csv")
temperatures <- ts(GLOBALTEMPERATURE[, "Anomaly"], frequency = 12, start = c(1880, 1))
```

```
autoplot(temperatures)
```

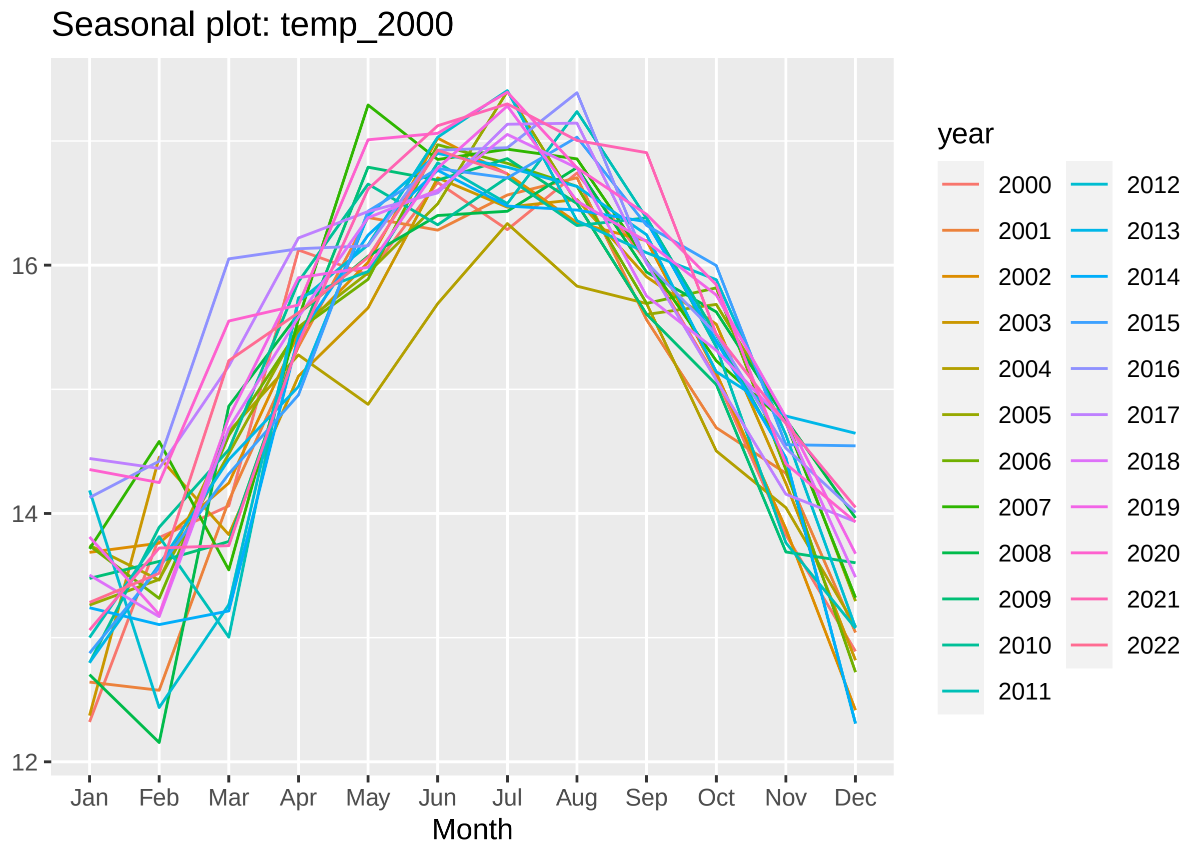


The data looks too noisy, and it seems impossible to dray any useful conclusions from it

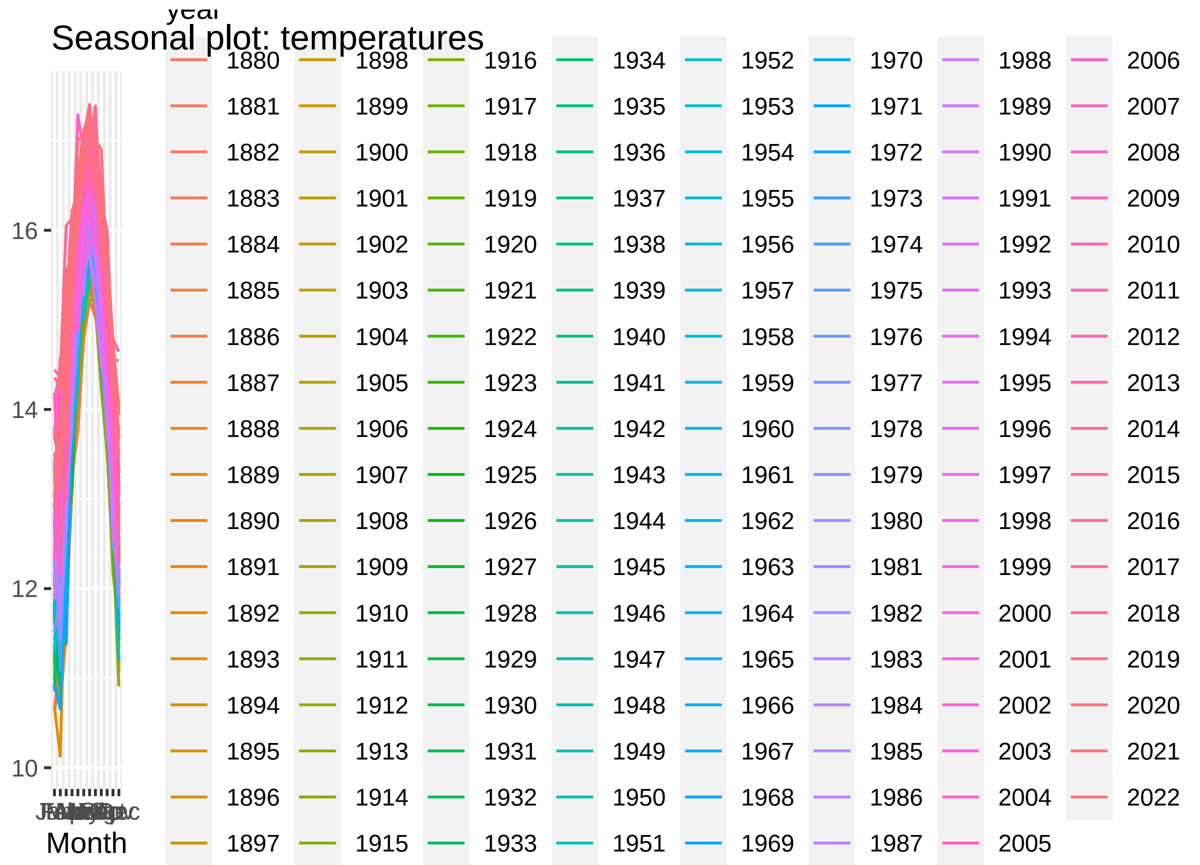
I will first cut the data from year 2000 onwards

```r
temp_2000 <- window(temperatures, start = c(2000, 1))
```

```r
ggseasonplot(temp_2000)
```



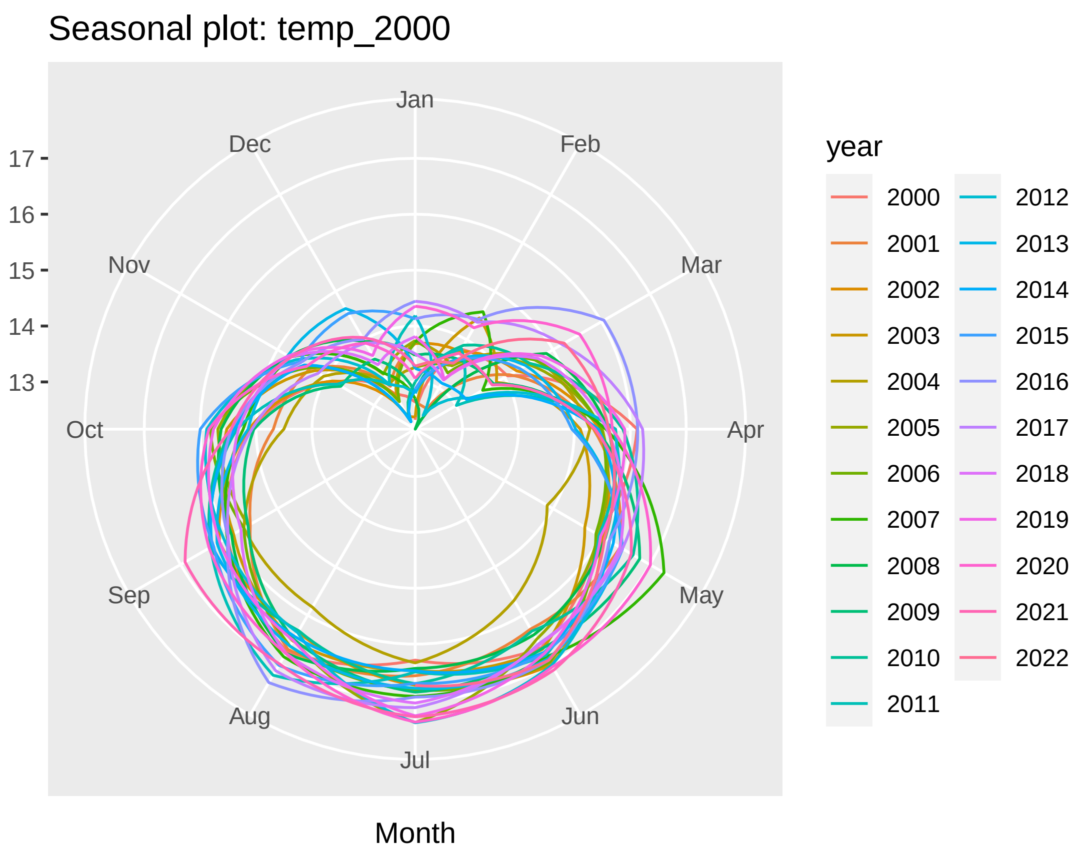

```r
ggseasonplot(temperatures)
```



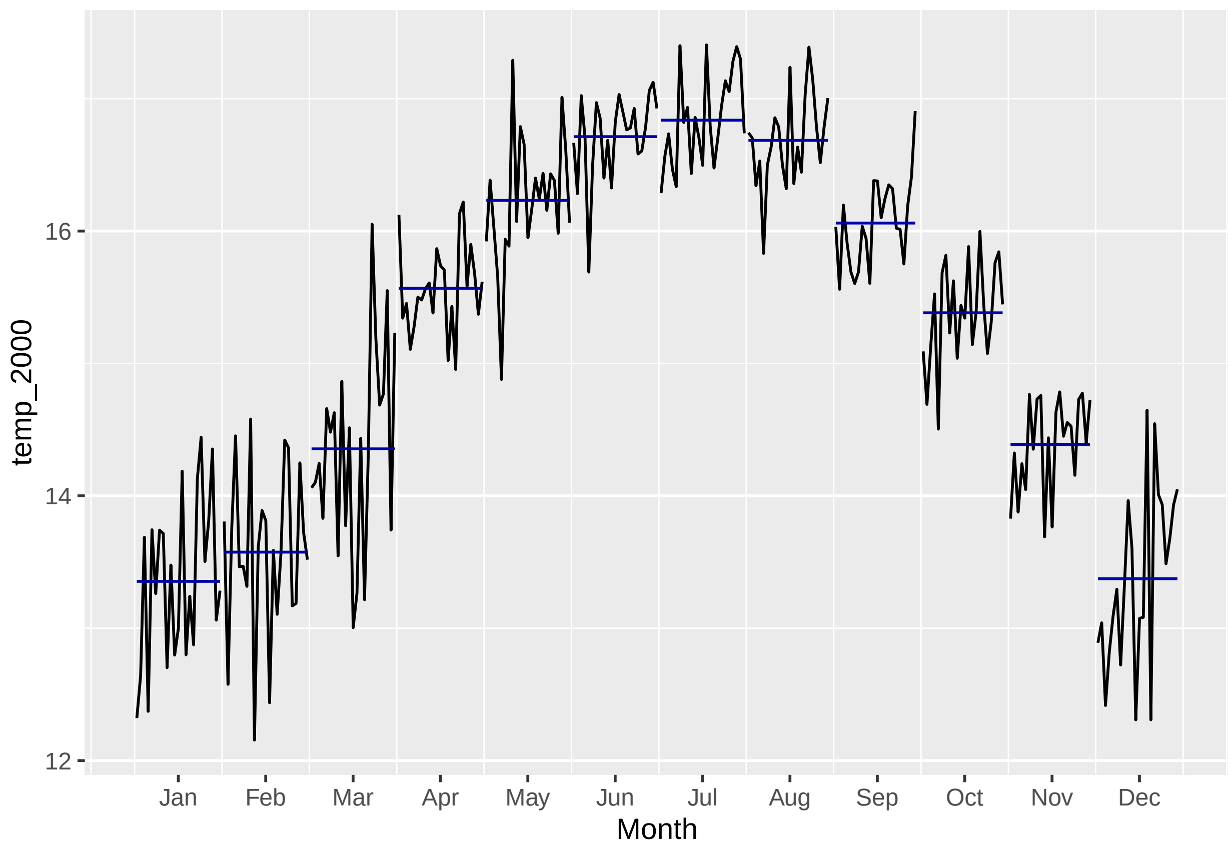
a clear seasonality is shown, where the temperature tend to be at a maximum between june and August (summer), there is also a general upward trend in the data.

if we looked at a polar version of the data

```r
ggseasonplot(temp_2000, polar = TRUE)
```




```
ggsubseriesplot(temp_2000)
```



Stationarity

Dicky-Fuller test (H_0: data is not stationary (H_0: unit root exists):

```
adf.test(temp_2000, k = 1)
```

```
## Warning in adf.test(temp_2000, k = 1): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  temp_2000
## Dickey-Fuller = -8.5867, Lag order = 1, p-value = 0.01
## alternative hypothesis: stationary
```

The dicky Fuller test suggested that the data is stationary (H_a is accepted) which is weird considering the general upward trend one can notice in the data. Trying with KPASS test (H_0 : data is stationary)

```
kpss.test(temp_2000)
```

```
## Warning in kpss.test(temp_2000): p-value greater than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  temp_2000
## KPSS Level = 0.21703, Truncation lag parameter = 5, p-value = 0.1
```

again, the hypothesis that the data is stationary was not rejected, which means that the data might be stationary, despite the general trend in the data.

by applying both tests on the original dataset:

```
{print(adf.test(temperatures))
print(kpss.test(temperatures))}
```

```
## Warning in adf.test(temperatures): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  temperatures
## Dickey-Fuller = -5.4253, Lag order = 11, p-value = 0.01
## alternative hypothesis: stationary
```

```
## Warning in kpss.test(temperatures): p-value smaller than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  temperatures
## KPSS Level = 8.7164, Truncation lag parameter = 8, p-value = 0.01
```

since $p \ll 0.01$ in KPSS test, H_0 is rejected which implies that the data is not stationary. the p value for the Dicky-Fuller test is also small.

check the differentiated data:

```
{print(adf.test(diff(temperatures)))
print(kpss.test(diff(temperatures)))}
```

```
## Warning in adf.test(diff(temperatures)): p-value smaller than printed p-value
```

```
##
##  Augmented Dickey-Fuller Test
##
## data:  diff(temperatures)
## Dickey-Fuller = -29.159, Lag order = 11, p-value = 0.01
## alternative hypothesis: stationary
```

```
## Warning in kpss.test(diff(temperatures)): p-value greater than printed p-value
```

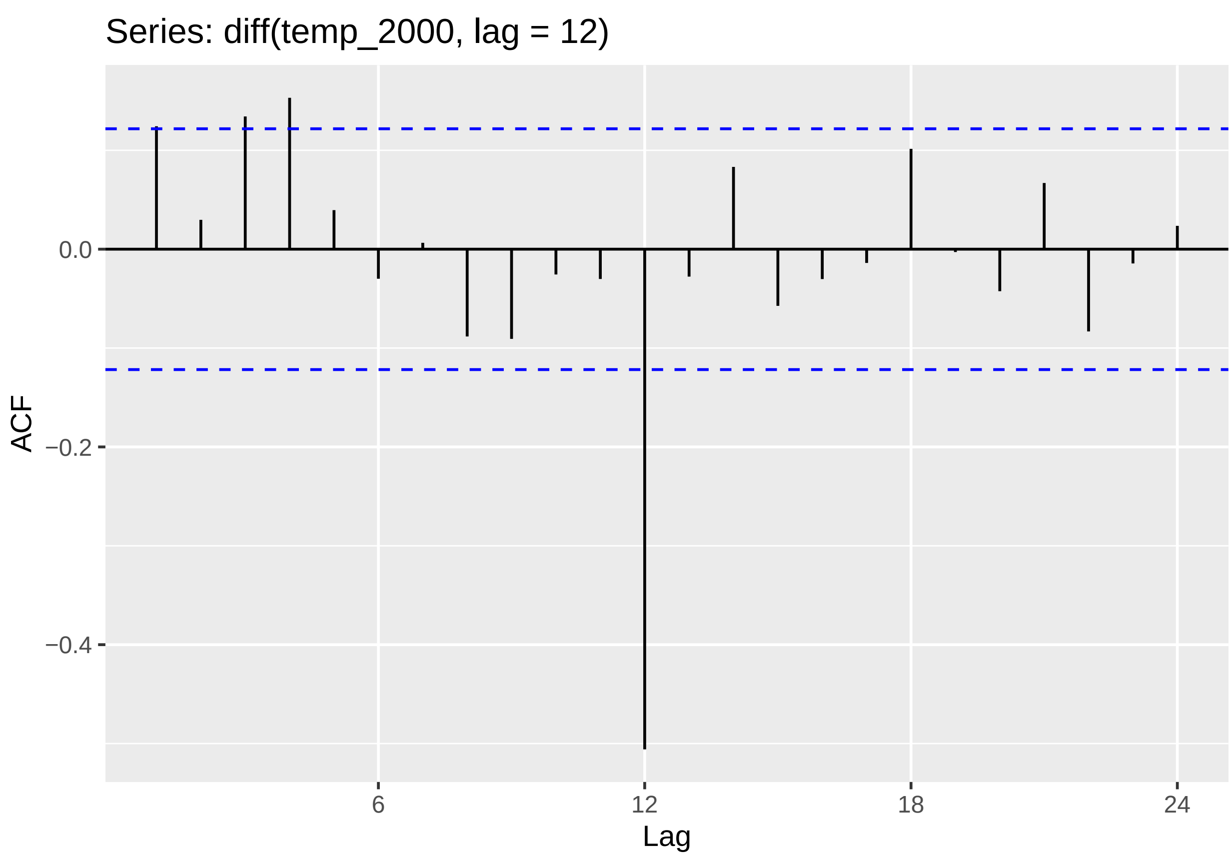
```
##
##  KPSS Test for Level Stationarity
##
## data:  diff(temperatures)
## KPSS Level = 0.0039969, Truncation lag parameter = 8, p-value = 0.1
```

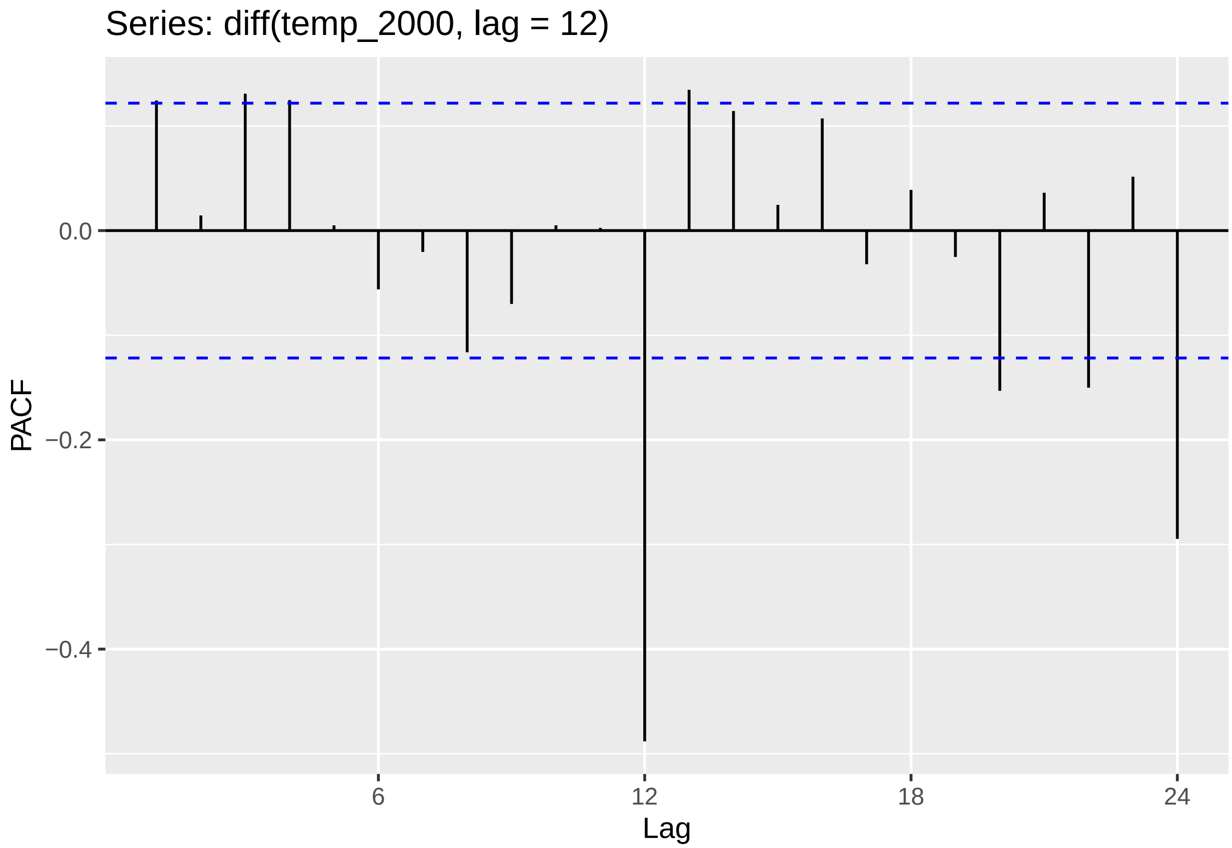
the differentiated data is stationary.

Despite no indication that the data has changed its trend, using a subset of the data seems to have introduced a bias in which the subset seemed stationary while it is not. maybe in this specific case, the predicted overall slope was considered to be too small to be significant. which made it the unstability unpredictable by the KPSS and ADF tests.

Fitting arima model

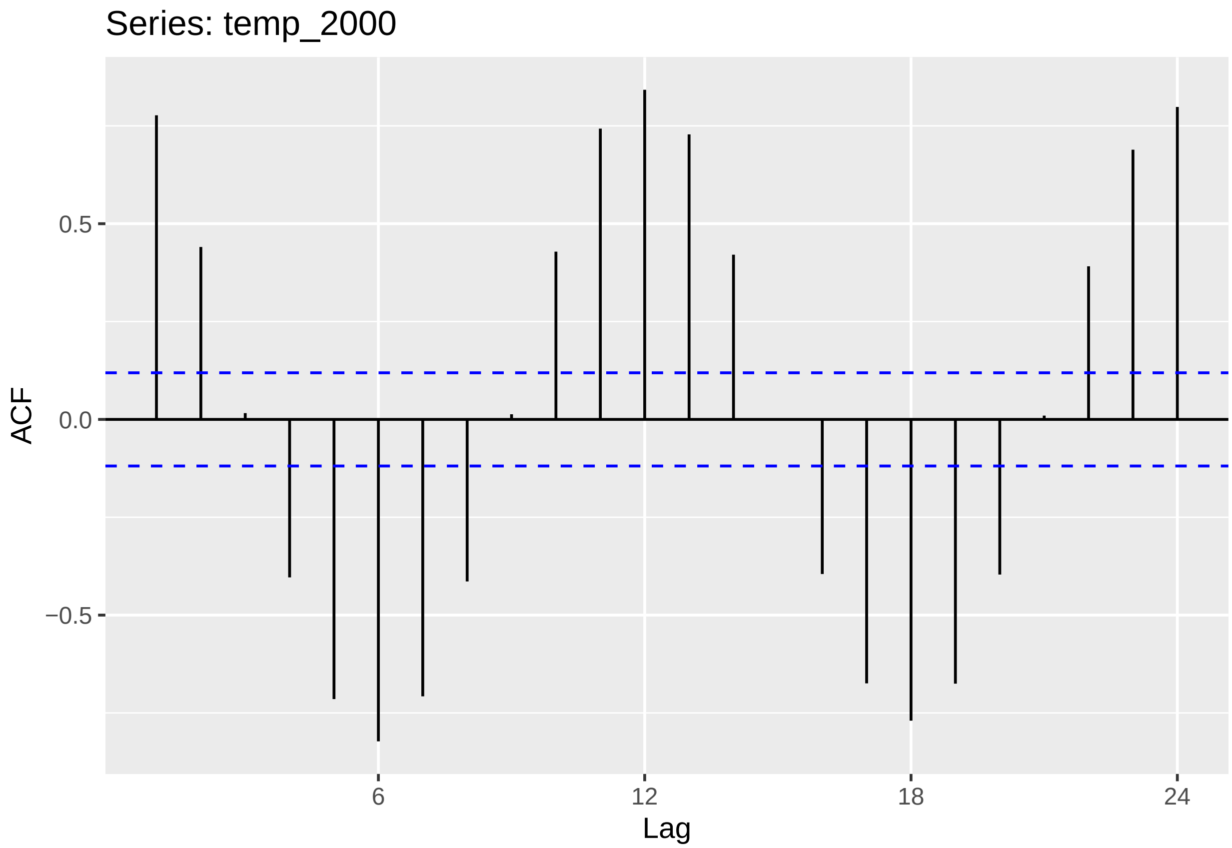
```
{
  show(ggAcf(diff(temp_2000, lag = 12)))
  show(ggPacf(diff(temp_2000, lag = 12)))

}
```

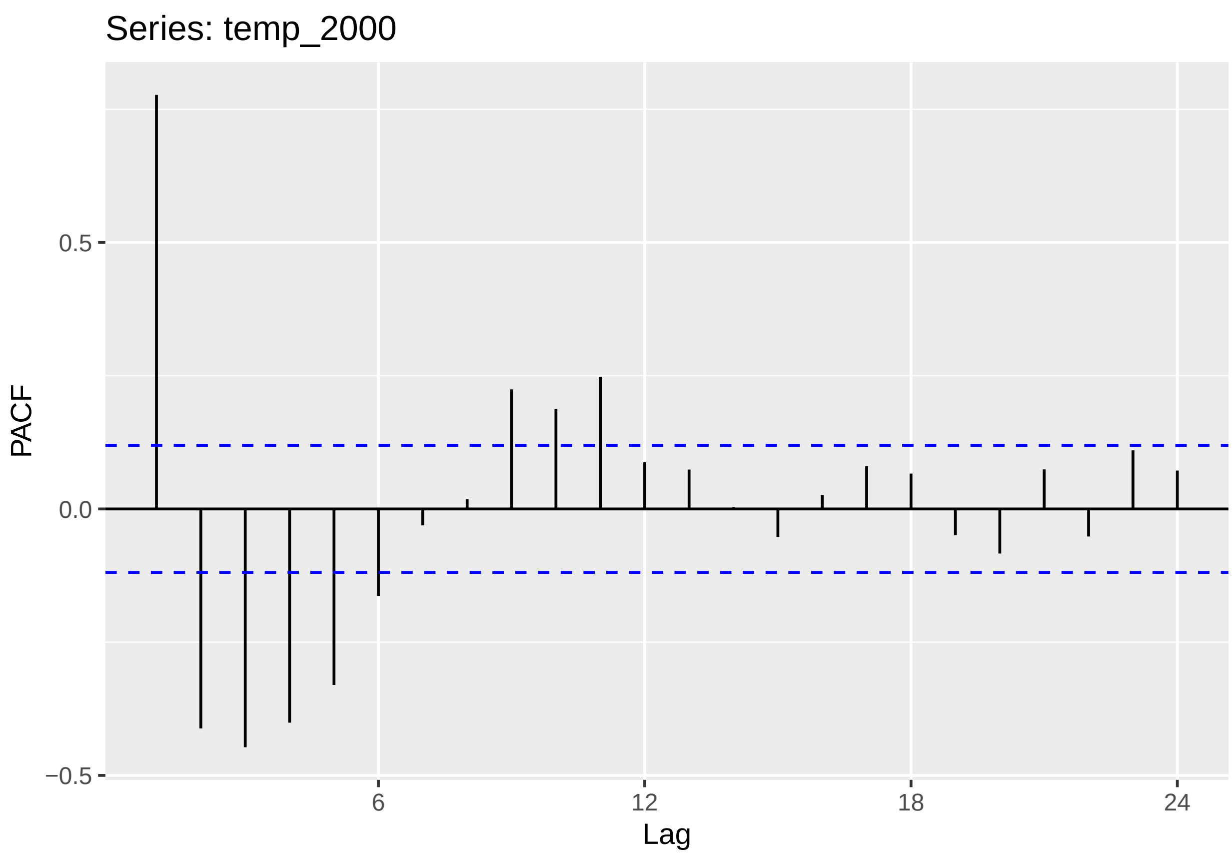




according to the figures above, there is a strong evidence the seasonal order is suspected to be $(0, 1, 1)_{[12]}$ since the Acf has one significant spike at $1 \times (12 \text{ lag})$. Whereas in terms of the non-seasonal part we notice a smaller spike at lag $= 4$ in the Acf, which suggests a non-seasonal order of $(0, 0, 4)$:

```
{
  show(ggAcf(temp_2000))
  show(ggPacf(temp_2000))
}
```


Series: temp_2000
ACF
0.5
0.0
−0.5
6
12
18
24
Lag



Using auto.arima:

```
temp_arima_fit <- auto.arima(temp_2000, approximation = FALSE, stepwise = FALSE)
summary(temp_arima_fit)
```

```
## Series: temp_2000
## ARIMA(0,0,4)(0,1,1)[12]
##
## Coefficients:
##          ma1     ma2     ma3     ma4     sma1
##       0.1650  0.0590  0.1279  0.1821  -0.8734
## s.e.  0.0631  0.0628  0.0599  0.0642   0.0446
##
## sigma^2 = 0.2313:  log likelihood = -184.1
## AIC=380.2   AICc=380.54   BIC=401.55
##
## Training set error measures:
##                      ME      RMSE     MAE       MPE     MAPE      MASE
## Training set 0.07849683 0.4656159 0.35167 0.4385436 2.377555 0.7018711
##                     ACF1
## Training set -0.00885666
```

trying other models

```r
minimum = temp_arima_fit$aic
ord = c(0, 0, 4)
for (i in 0:5) {
  for (j in 0:5) {
    tord = c(i, 0, j)
    if(Arima(temp_2000, tord, c(0, 1, 1), include.drift = TRUE)$aic < minimum){
      minimum = Arima(temp_2000, tord, c(0, 1, 1), include.drift = TRUE)$aic
      ord = tord
    }
  }
}

model = Arima(temp_2000, ord, c(0, 1, 1), include.drift = TRUE)
summary(model)
```

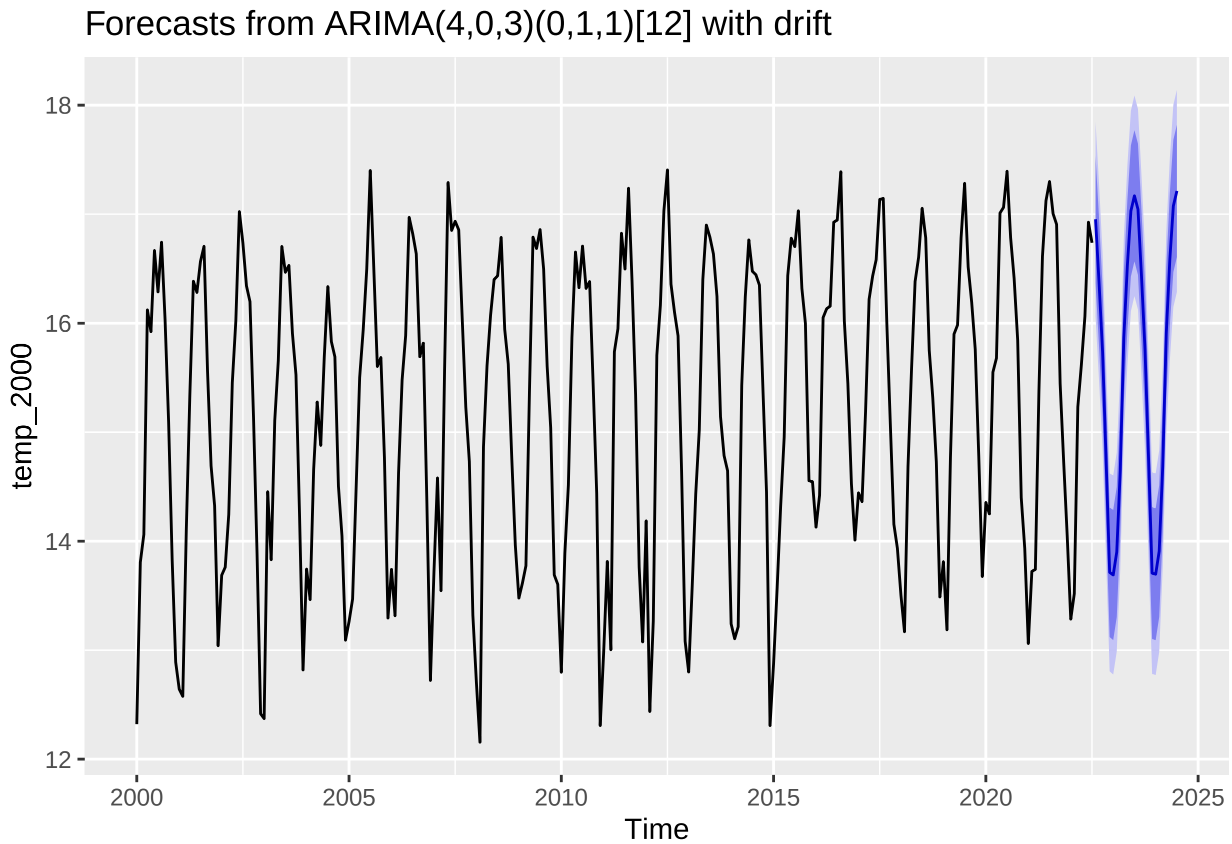
```
## Series: temp_2000
## ARIMA(4,0,3)(0,1,1)[12] with drift
##
## Coefficients:
##          ar1     ar2      ar3     ar4      ma1      ma2     ma3    sma1   drift
##       0.8041  0.4153  -0.8899  0.1020  -0.6737  -0.5237  0.9427  -1.000  0.0023
## s.e.  0.1029  0.1682   0.1362  0.0694   0.0848   0.1344  0.0845   0.074  0.0004
##
## sigma^2 = 0.1978:  log likelihood = -173.59
## AIC=367.18   AICc=368.07   BIC=402.75
##
## Training set error measures:
##                        ME      RMSE       MAE         MPE     MAPE      MASE
## Training set -0.002199258 0.4271384 0.3175886 -0.08912132 2.153839 0.6338508
##                       ACF1
## Training set -0.0009752795
```

Forcasting

```r
arima_Forecast_temp = forecast(model, h = 24)
arima_Forecast_temp$mean
```

```
##           Jan      Feb      Mar      Apr      May      Jun      Jul      Aug
## 2022                                                                16.95278
## 2023 13.68914 13.90013 14.65278 15.86646 16.52883 17.02913 17.16847 17.04413
## 2024 13.69639 13.90824 14.68582 15.90470 16.58236 17.07502 17.21248
##           Sep      Oct      Nov      Dec
## 2022 16.37392 15.73291 14.73236 13.71387
## 2023 16.42912 15.74794 14.74551 13.70641
## 2024
```

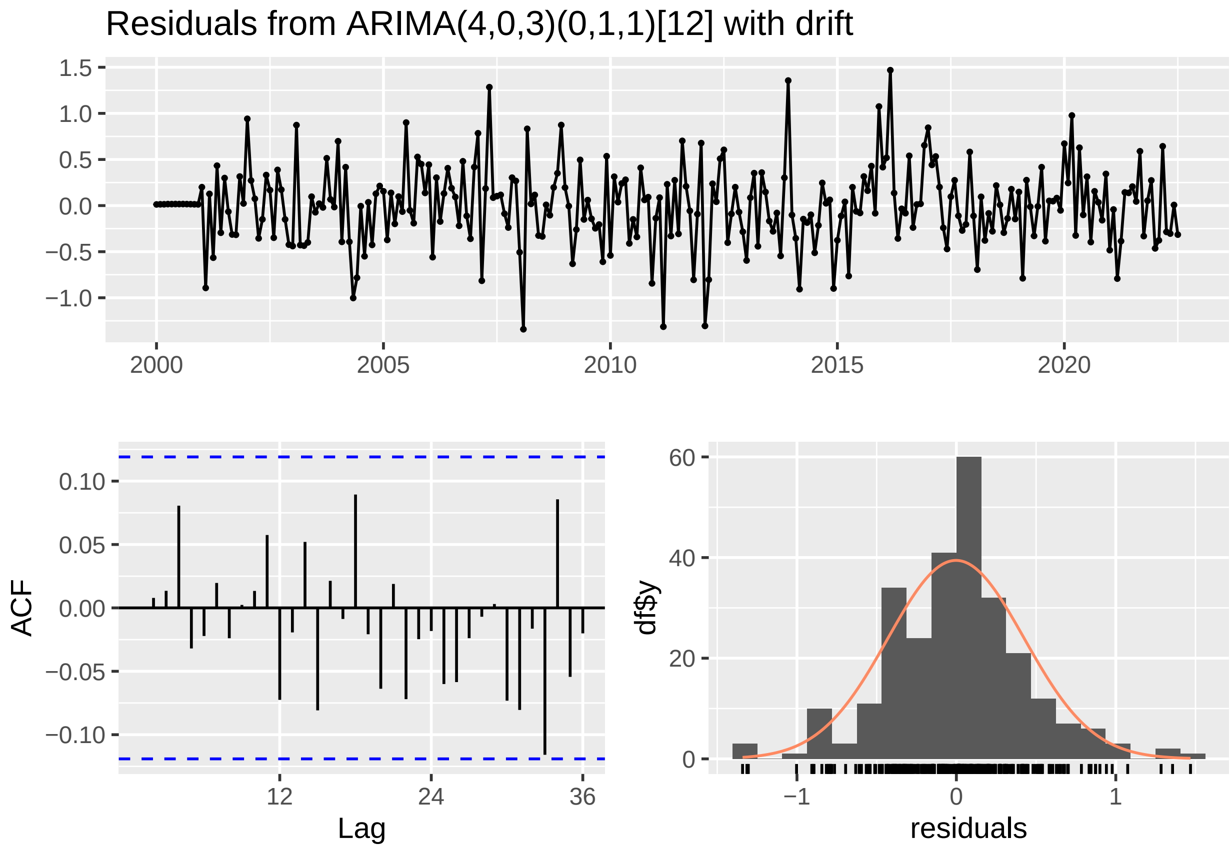
```r
autoplot(arima_Forecast_temp)
```



the forecasted value on march 2024 is 14.69 ± 0.60 ^oC (with 80% confidence interval) which is agrees with the experimental result of 14.14 ^oC.

residuals check

```
checkresiduals(model, lag = 12)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(4,0,3)(0,1,1)[12] with drift
## Q* = 5.0544, df = 4, p-value = 0.2818
##
## Model df: 8.   Total lags used: 12
```

implying that the null hypothosis is not rejected, thus the data is not correlated and corresponds to white noise

Regression

since the data shows strong seasonality, it will be good to fit it using periodic functions (sin and cos) of period 12

```
t = seq_along(temp_2000)

xreg0 = cbind(
  sin(pi*t/6),
  cos(pi*t/6),
  t
)
```

```r
regression_fit = auto.arima(temp_2000, xreg = xreg0, approximation = FALSE, stepwise = FALSE, seasonal =

autoplot(forecast(regression_fit, xreg = xreg0))
```


```r
model2 <- regression_fit
summary(model2)
```

```
## Series: temp_2000
## Regression with ARIMA(0,0,1) errors
##
## Coefficients:
##          ma1  intercept                          t
##       0.1714    14.9027  -0.8395  -1.5988  0.0023
## s.e.  0.0611     0.0664   0.0461   0.0460  0.0004
##
## sigma^2 = 0.2212:  log likelihood = -177.6
## AIC=367.21   AICc=367.52   BIC=388.82
##
## Training set error measures:
##                        ME      RMSE       MAE        MPE     MAPE      MASE
## Training set 0.0004926113 0.4659714 0.3602271 -0.1056169 2.453849 0.7189497
##                      ACF1
## Training set -0.002494955
```

This model has slightly better stats in terms of aicc, but with fewer coefficient, notice also that the seasonality no longer appeared in the error terms, suggesting that it was captured completely by the $\sin(\omega t)$ and $\cos(\omega t)$ terms.

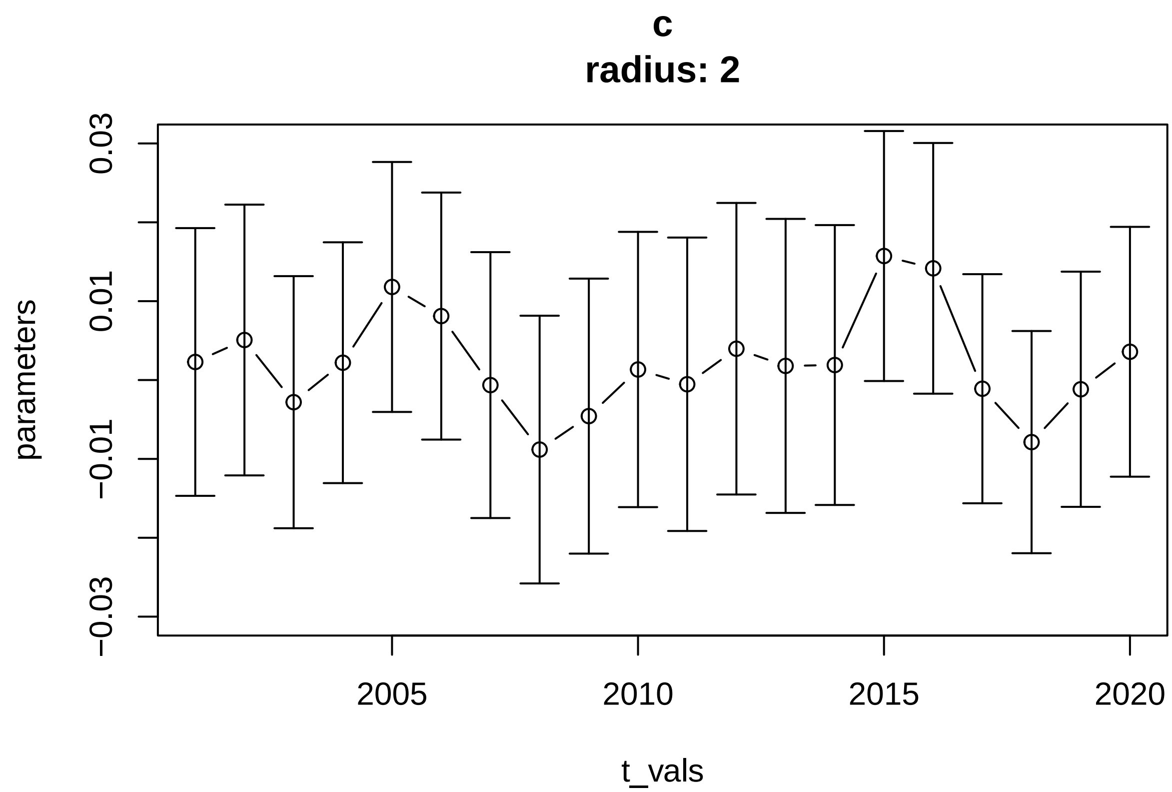
By including a linear term, we compensate for the drift in the origional sArima model. The significance of the linear term t coefficient is an evidence of non-stationarity of the original data, which in this case reflects global warming.

to observe the effects of global warming, I will take segment of the data centered at year x, I will then fit a corresponding model with arima(0, 0, 1) errors, and observe the stability of the linear term coefficient ct

```
linear_coef <- function(DATA, x, radius = 2){
  temporary_data = window(DATA, start = c(x-radius, 1), end = c(x+radius, 1))
  new_t <- seq_along(temporary_data)
  temporary_xreg = cbind(
    sin(new_t),
    cos(new_t),
    new_t
  )
  temporary_model = Arima(y = temporary_data, order = c(0, 0, 1), seasonal = c(0, 0, 0), xreg = tempora
  std_error <- sqrt(diag(vcov(temporary_model)))
  return(c(as.numeric(temporary_model$coef["new_t"]), as.numeric(sqrt(diag(vcov(temporary_model)))["new
}
```

```
parameters =c()
errors = c()

for (i in 1:20){
    u = linear_coef(temperatures, 2000 + i)
    parameters <- cbind(parameters, u[1])
    errors <- cbind(errors, u[2])
}

t_vals = c(2001:2020)
```

```
plot(t_vals, parameters, type='b', main="c\nradius: 2", ylim = c(-0.03, 0.03)) +
  arrows(x0=t_vals, y0=parameters-errors, x1 = t_vals, y1=parameters+errors, code=3, angle = 90, length
```

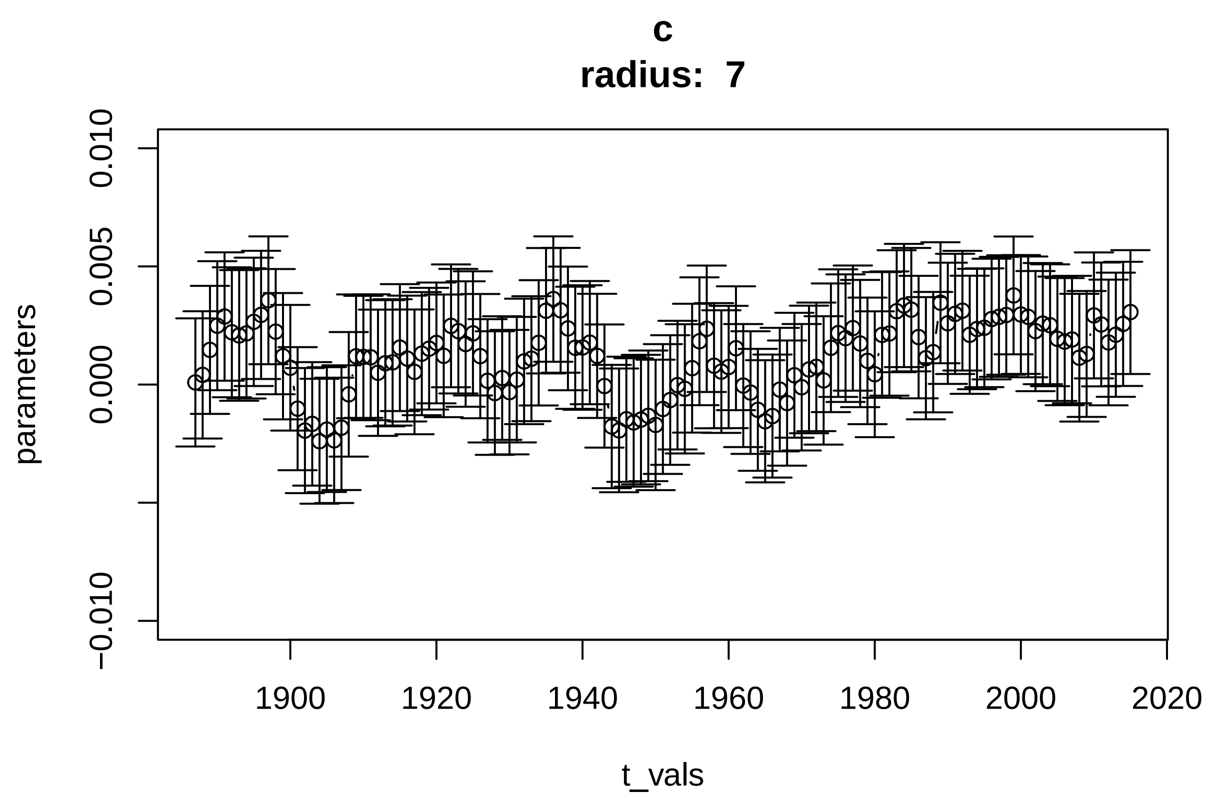



```
## integer(0)
```

The values of the trending coeficient is not significant for any value of x, this is most likely due to the short time intervals we were considering.

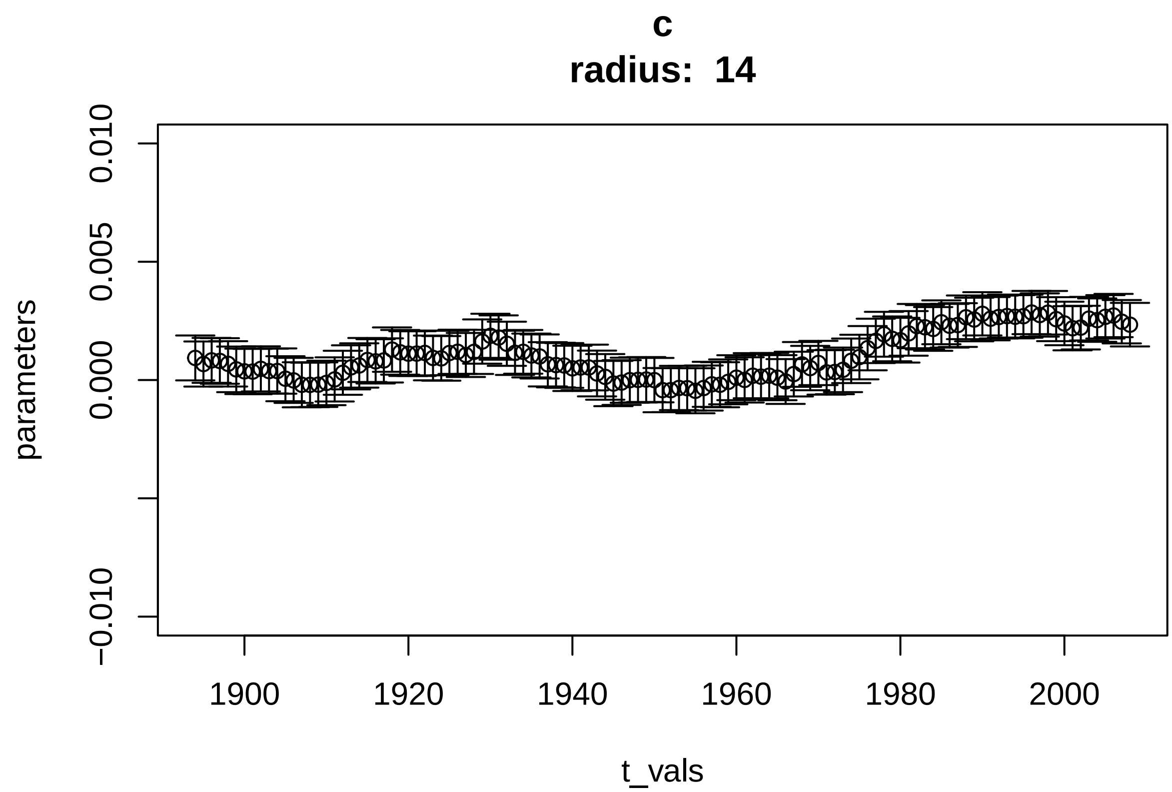
I will now increase the radius:

```
parameters =c()
errors = c()
rad = 7

for (i in (1880 + rad):(2022 - rad)){
    u = linear_coef(temperatures,  i , rad)
    parameters <- cbind(parameters, u[1])
    errors <- cbind(errors, u[2])
}

t_vals = c((1880 + rad):(2022 - rad))

plot(t_vals, parameters, type='b', main=paste("c\nradius: ", toString(rad)), ylim = c(-0.010, 0.010)) +
  arrows(x0=t_vals, y0=parameters-errors, x1 = t_vals, y1=parameters+errors, code=3, angle = 90, length
```



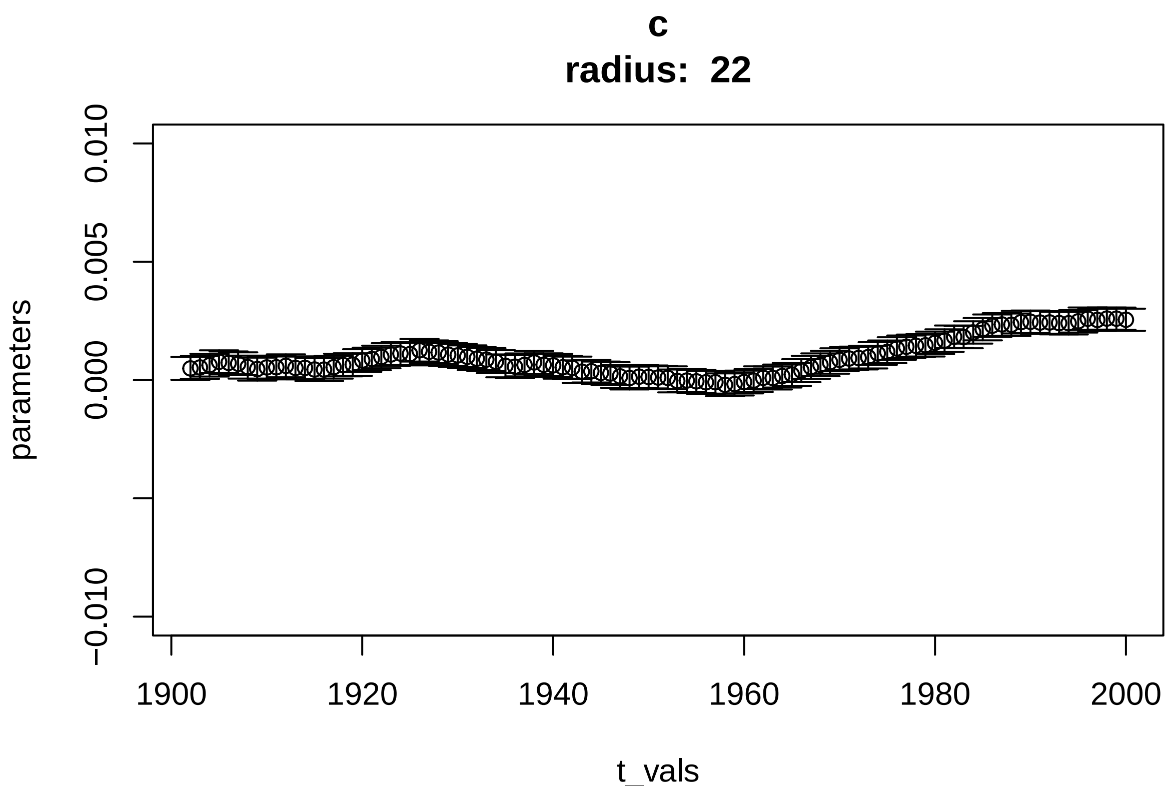

```
## integer(0)
```

```
parameters =c()
errors = c()
rad = 14

for (i in (1880 + rad):(2022 - rad)){
    u = linear_coef(temperatures,  i , rad)
    parameters <- cbind(parameters, u[1])
    errors <- cbind(errors, u[2])
}

t_vals = c((1880 + rad):(2022 - rad))

plot(t_vals, parameters, type='b', main=paste("c\nradius: ", toString(rad)), ylim = c(-0.010, 0.010)) +
  arrows(x0=t_vals, y0=parameters-errors, x1 = t_vals, y1=parameters+errors, code=3, angle = 90, length
```




```
## integer(0)
```

```
parameters =c()
errors = c()
rad = 22

for (i in (1880 + rad):(2022 - rad)){
    u = linear_coef(temperatures,  i , rad)
    parameters <- cbind(parameters, u[1])
    errors <- cbind(errors, u[2])
}

t_vals = c((1880 + rad):(2022 - rad))

plot(t_vals, parameters, type='b', main=paste("c\nradius: ", toString(rad)), ylim = c(-0.010, 0.010)) +
  arrows(x0=t_vals, y0=parameters-errors, x1 = t_vals, y1=parameters+errors, code=3, angle = 90, length
```




```
## integer(0)
```

by increasing the radius (to $r = 14$), the linear coefficient looks more stable and significant, but this made it impossible to track-down any effect of the covid era on the increasing temperature coefficient.

Conclusion

The temperature data is non stationary, as evident by the nonzero trend factor $T \propto ct \neq 0$. it also shows strong seasonality

The fact that the aforementioned coefficient is positive implies that there is global warming.

Due to the small number of cycles (years) after the covid era it is not possible to study the effect of covid on global warming, this is because of the instability in the model parameters

Using rolling window approach, it was shown that the global warming effect have 'accelerated' since around 1960, with notable improvement since 1985.