

Buoy Report

Yuxin Zeng

Abstract

In order to study whether there is evidence of global warming in the data collected by a single weather buoy in the NOAA National Data Buoy Center, I study the relationship between the air temperature measured by buoy and the year. First, I choose the data from 1997 to 2016 as the sample, and use the data at 12AM to represent the data every day. After preprocessing the data, I calculated the mean temperature of each month for each year, used it to regress the years, and visualized these regressions. Then I built a Seasonal ARIMA model to find the law and predict the next three years. Neither of these results provide a clear trend of increase, so I come to the conclusion that the answer for the beginning question is “No”.

1. Background Information

“Global warming” is an environmental issue that has been widely discussed. We hope to find out whether there is evidence of global warming in the data (esp. air temperature) collected by a single weather buoy in the NOAA National Data Buoy Center. The buoy we choose is from Station 44013 — BOSTON 16 NM East of Boston, MA (42°20'44" N 70°39'4" W).

2. Data Collection

3. Data Preprocessing

Select 20 years of data from 1997 to 2016. Choose the data at 12:00AM to represent the data of each day. Use air temperature (ATMP) as a measure of global warming, deleting the rest variables except time variables. Air temperature greater than 99.0 Celsius are regarded as abnormal points, therefore are deleted. Use the unite function to change the date format.

```
load("C:/Users/lenovo/Desktop/615 R/buoy/MR.RData")
buoy = data.frame(MR)
buoy_1 = buoy %>% filter(YYYY >= "1997")
buoy_2 = buoy_1 %>% filter(hh == "12")
buoy_3 = buoy_2[,c(1,2,3,13)]
buoy_4 = buoy_3[buoy_3$ATMP<99.0,]
buoy_5 = buoy_4 %>% unite(date,YYYY,MM,DD,sep="-")

write.csv(buoy_4,file="C:/Users/lenovo/Desktop/615 R/buoy/Everyday.csv")
write.csv(buoy_5,file="C:/Users/lenovo/Desktop/615 R/buoy/Everyday_date.csv")
```

4. Data Analysis & Visualization

Calculate the average temperature for each month from 1997 to 2016.

```
buoy=read.csv("Everyday.csv",header=TRUE)
Jan=0
for(i in 1997:2016){Jan[i-1996]=mean(buoy$ATMP[buoy$YYYY==i|buoy$MM=="01"])}
Feb=0
for(i in 1997:2016){Feb[i-1996]=mean(buoy$ATMP[buoy$YYYY==i|buoy$MM=="02"])}
# ... (similar code for other months) ...
```

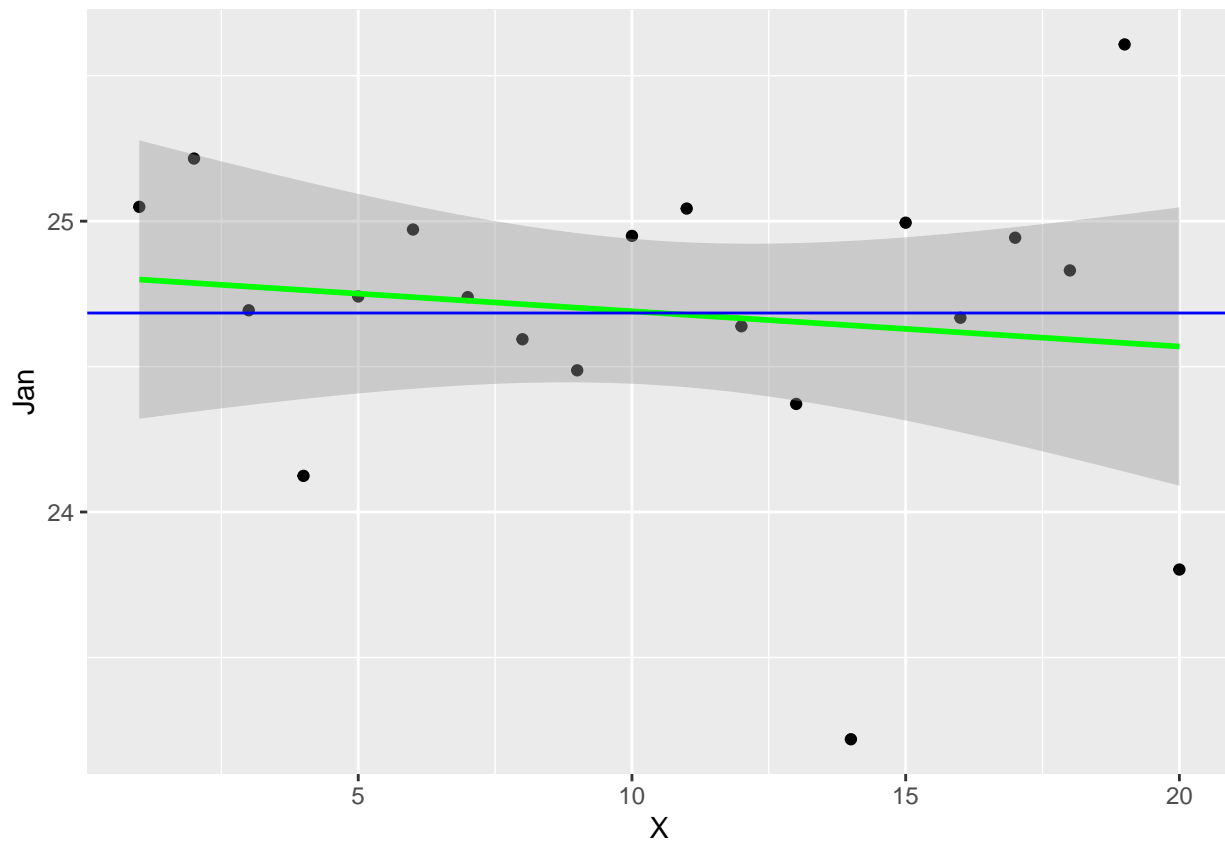
```

Mar=0
for(i in 1997:2016){Mar[i-1996]=mean(buoy$ATMP[buoy$YYYY==i | buoy$MM=="3"])}
Apr=0
for(i in 1997:2016){Apr[i-1996]=mean(buoy$ATMP[buoy$YYYY==i | buoy$MM=="4"])}
May=0
for(i in 1997:2016){May[i-1996]=mean(buoy$ATMP[buoy$YYYY==i | buoy$MM=="5"])}
Jun=0
for(i in 1997:2016){Jun[i-1996]=mean(buoy$ATMP[buoy$YYYY==i | buoy$MM=="6"])}
Jul=0
for(i in 1997:2016){Jul[i-1996]=mean(buoy$ATMP[buoy$YYYY==i | buoy$MM=="7"])}
Aug=0
for(i in 1997:2016){Aug[i-1996]=mean(buoy$ATMP[buoy$YYYY==i | buoy$MM=="8"])}
Sep=0
for(i in 1997:2016){Sep[i-1996]=mean(buoy$ATMP[buoy$YYYY==i | buoy$MM=="9"])}
Oct=0
for(i in 1997:2016){Oct[i-1996]=mean(buoy$ATMP[buoy$YYYY==i | buoy$MM=="10"])}
Nov=0
for(i in 1997:2016){Nov[i-1996]=mean(buoy$ATMP[buoy$YYYY==i | buoy$MM=="11"])}
Dec=0
for(i in 1997:2016){Dec[i-1996]=mean(buoy$ATMP[buoy$YYYY==i | buoy$MM=="12"])}

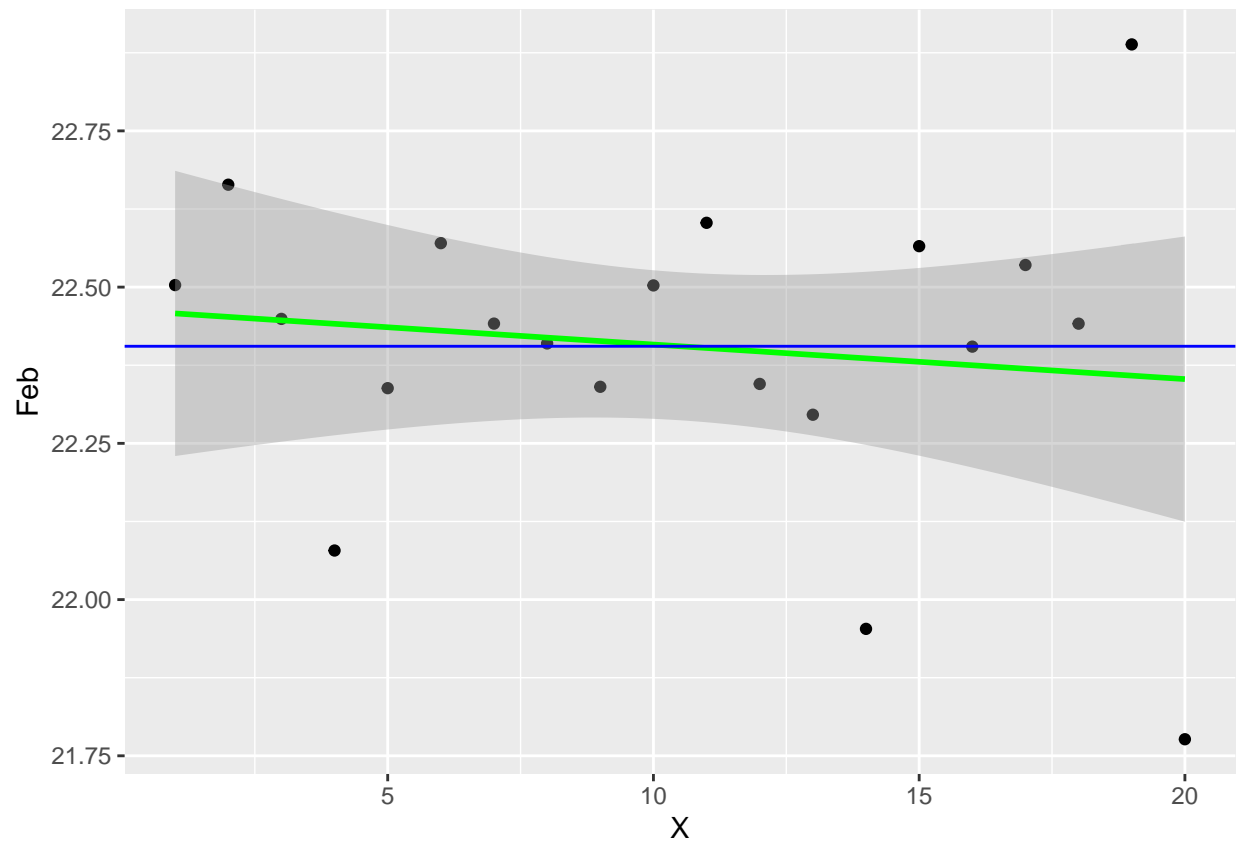
atmp=cbind.data.frame(Year=1997:2016,Jan,Feb,Mar,Apr,May,Jun,Jul,Aug,Sep,Oct,Nov,Dec)
write.csv(atmp,file="C:/Users/lenovo/Desktop/615 R/buoy/atmp.csv")

```

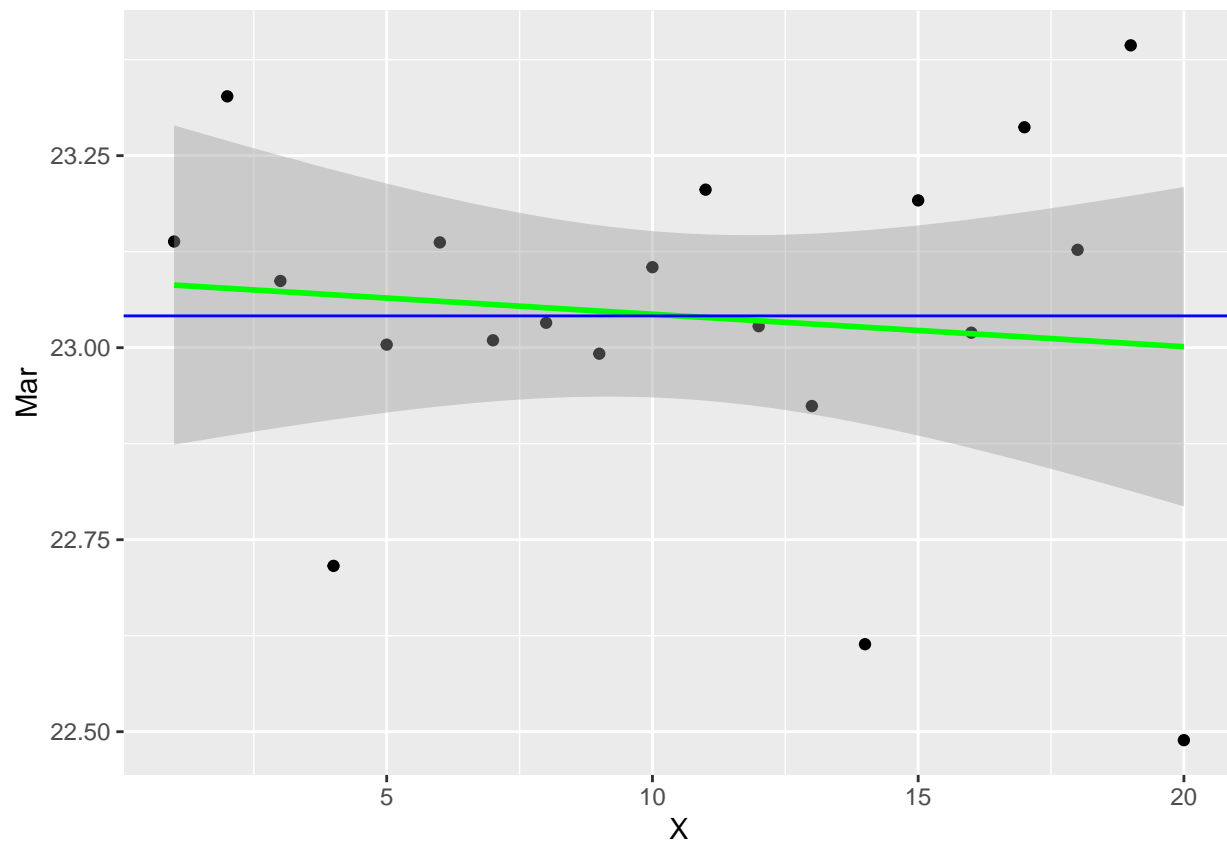
```
## `geom_smooth()` using formula 'y ~ x'
```



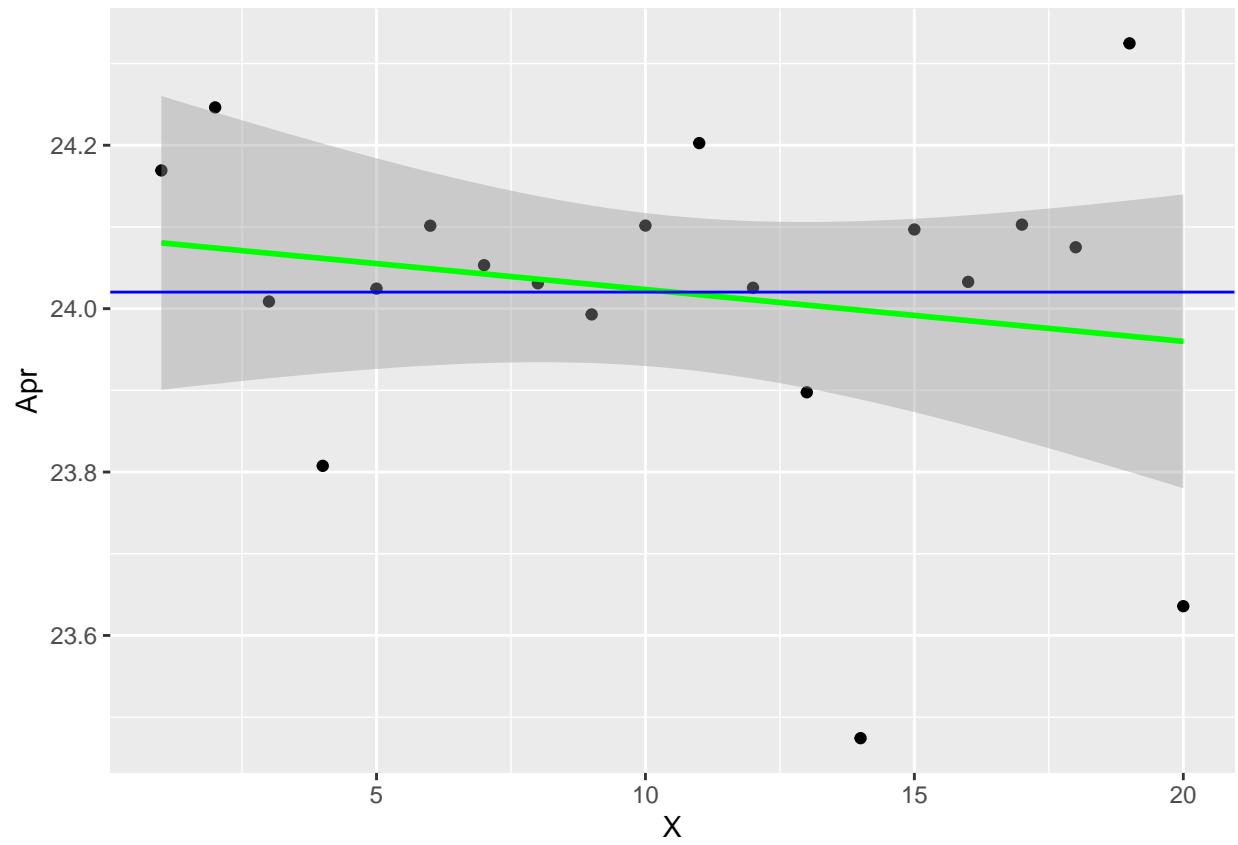
```
## `geom_smooth()` using formula 'y ~ x'
```



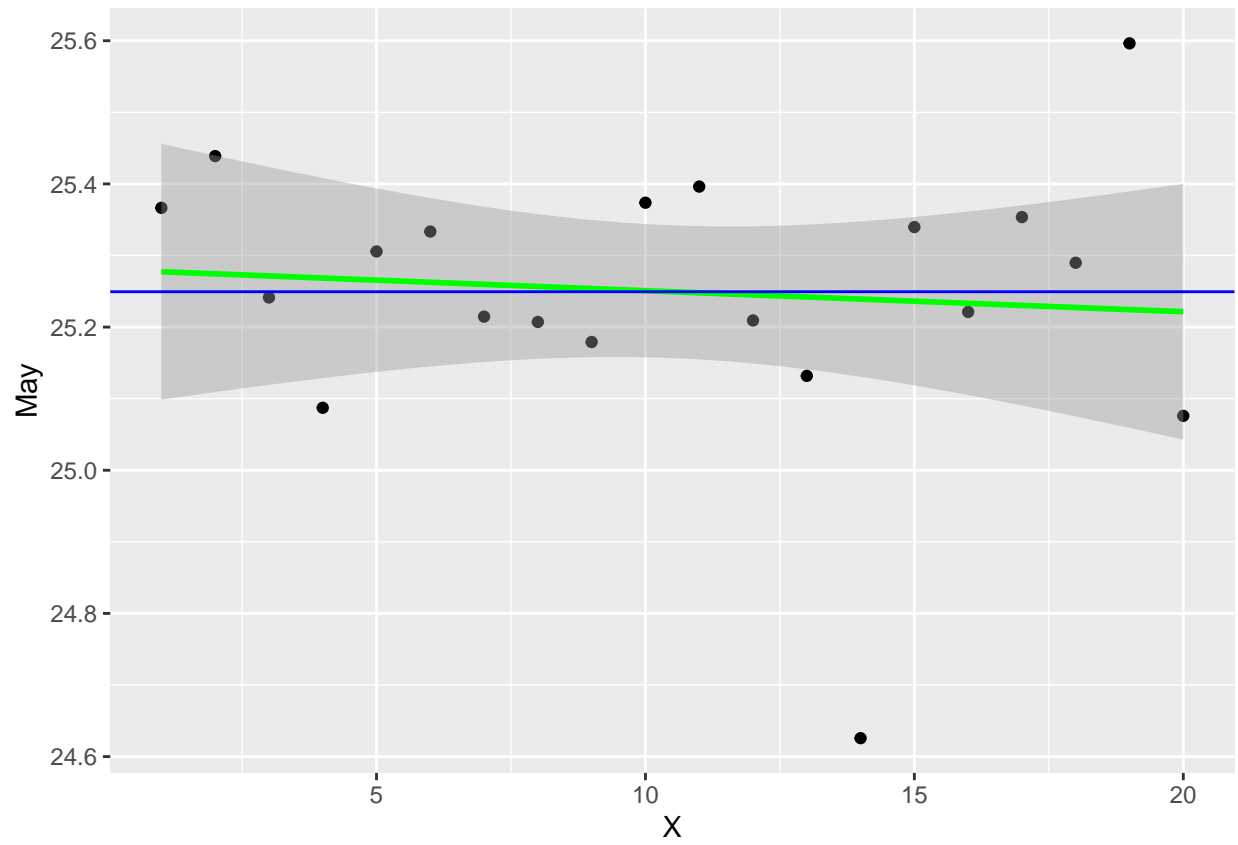
```
## `geom_smooth()` using formula 'y ~ x'
```



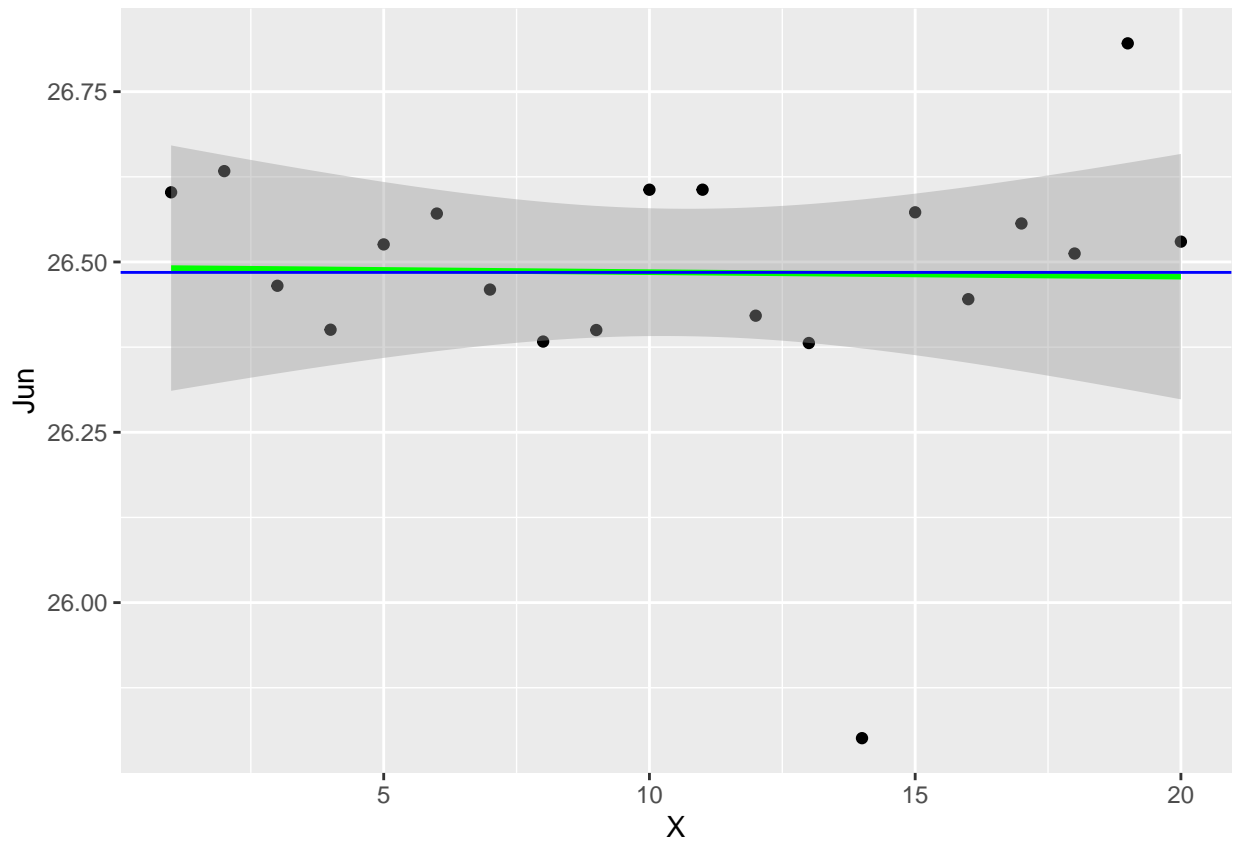
```
## `geom_smooth()` using formula 'y ~ x'
```



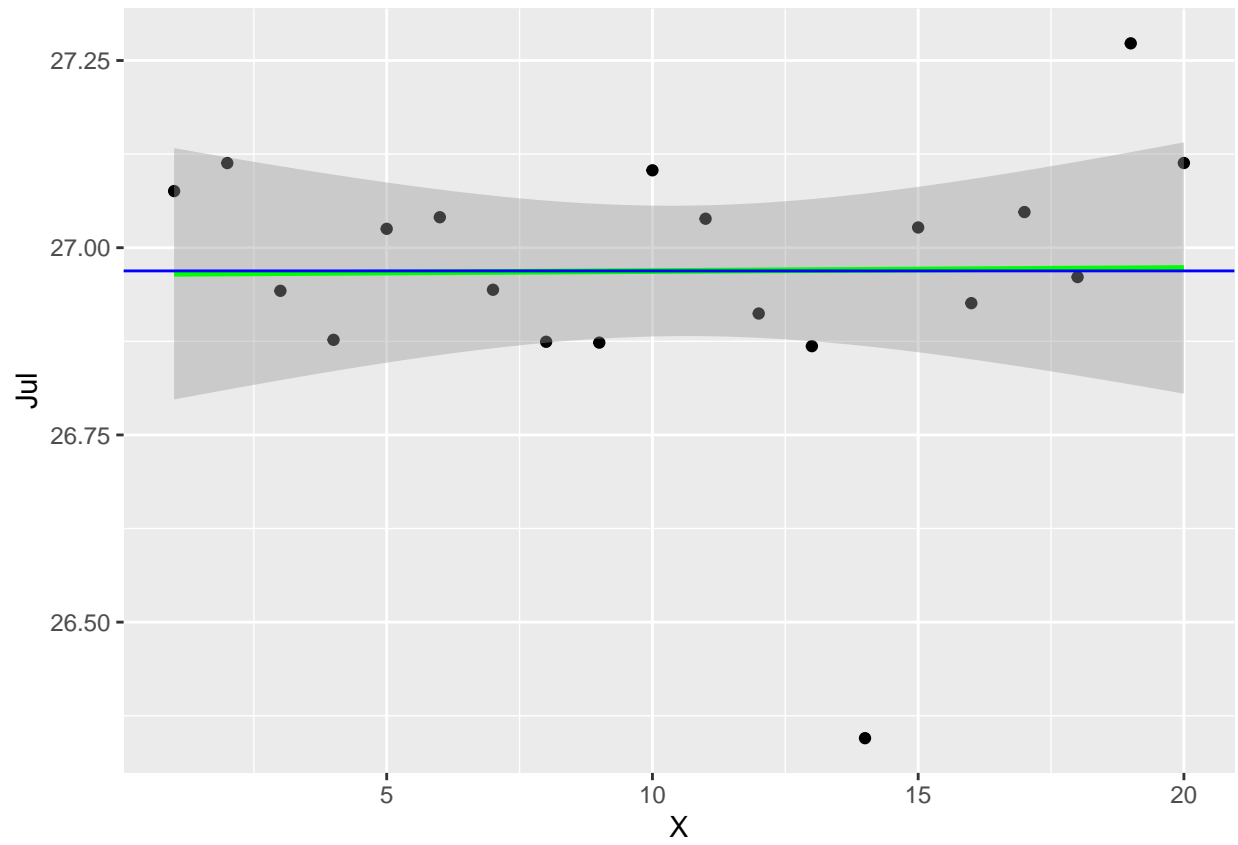
```
## `geom_smooth()` using formula 'y ~ x'
```



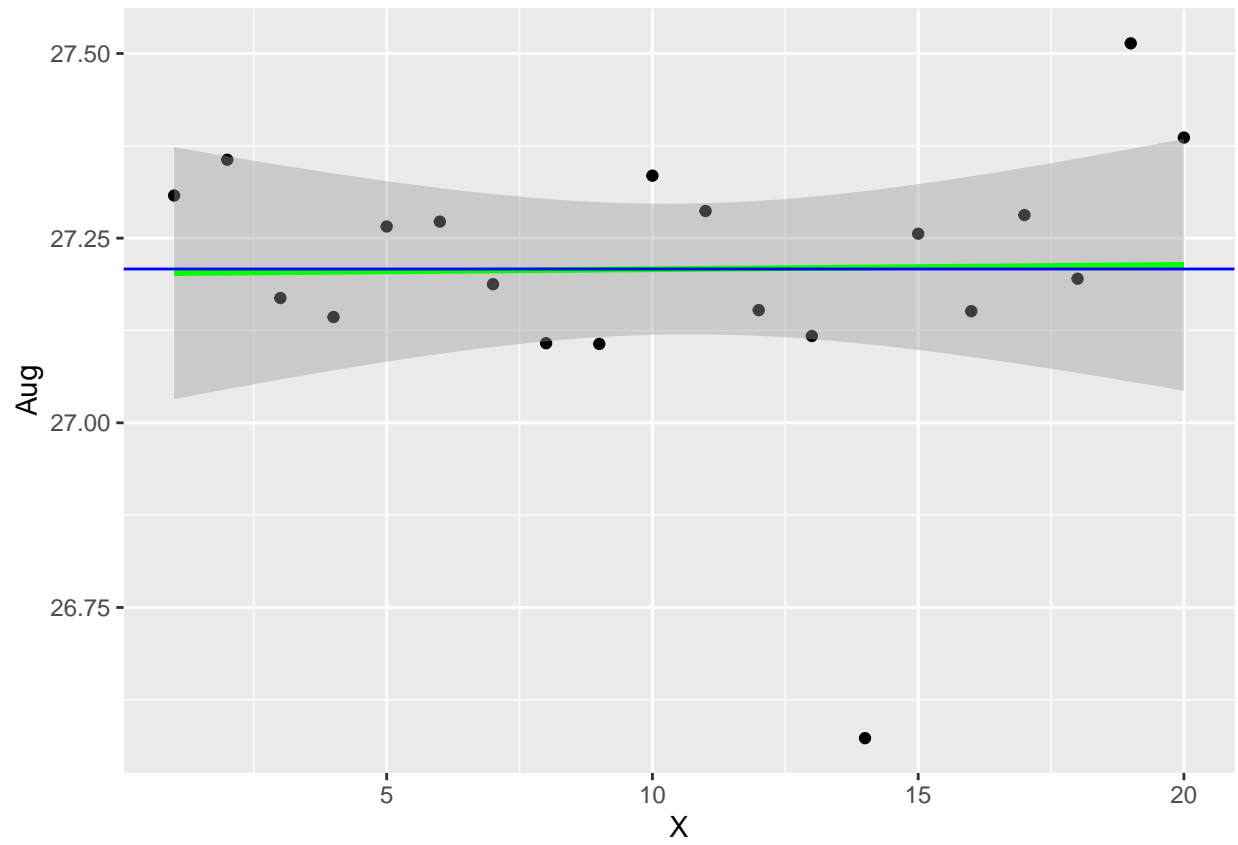
```
## `geom_smooth()` using formula 'y ~ x'
```



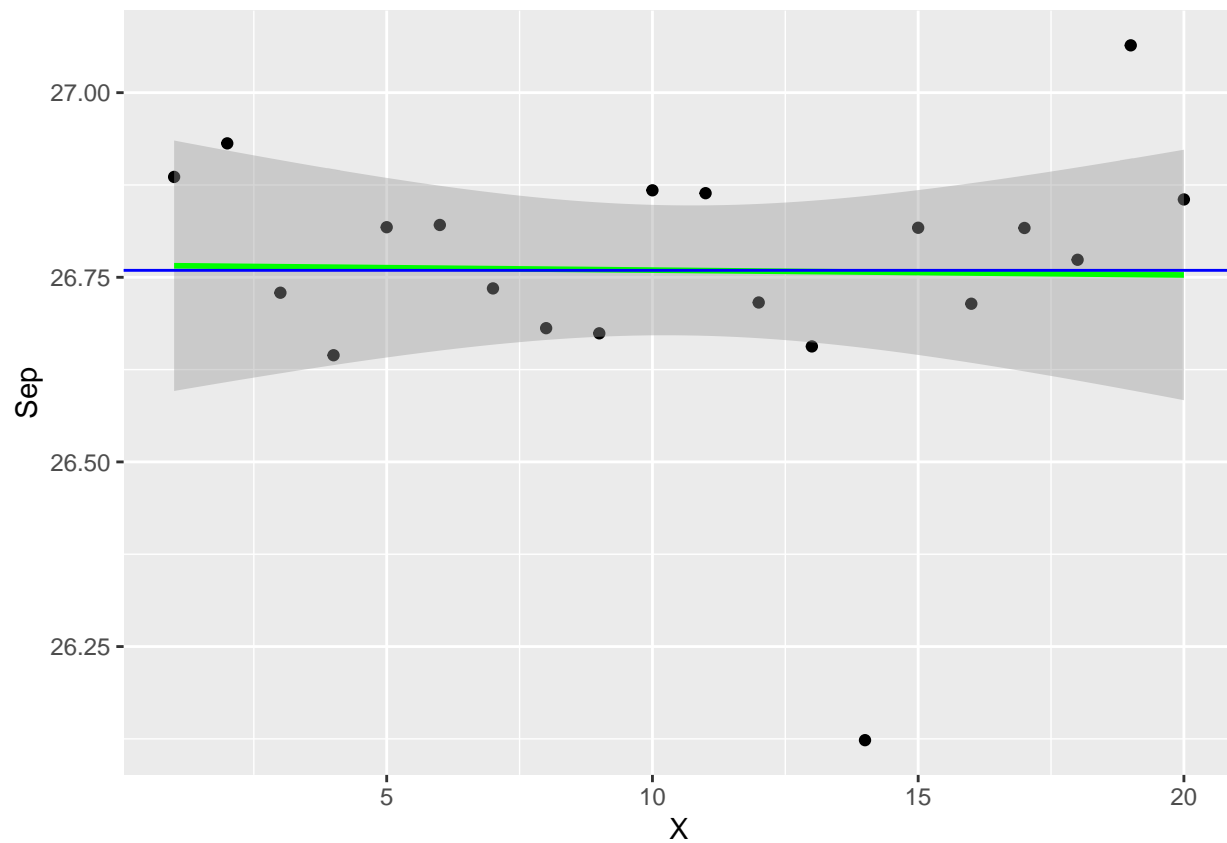
```
## `geom_smooth()` using formula 'y ~ x'
```



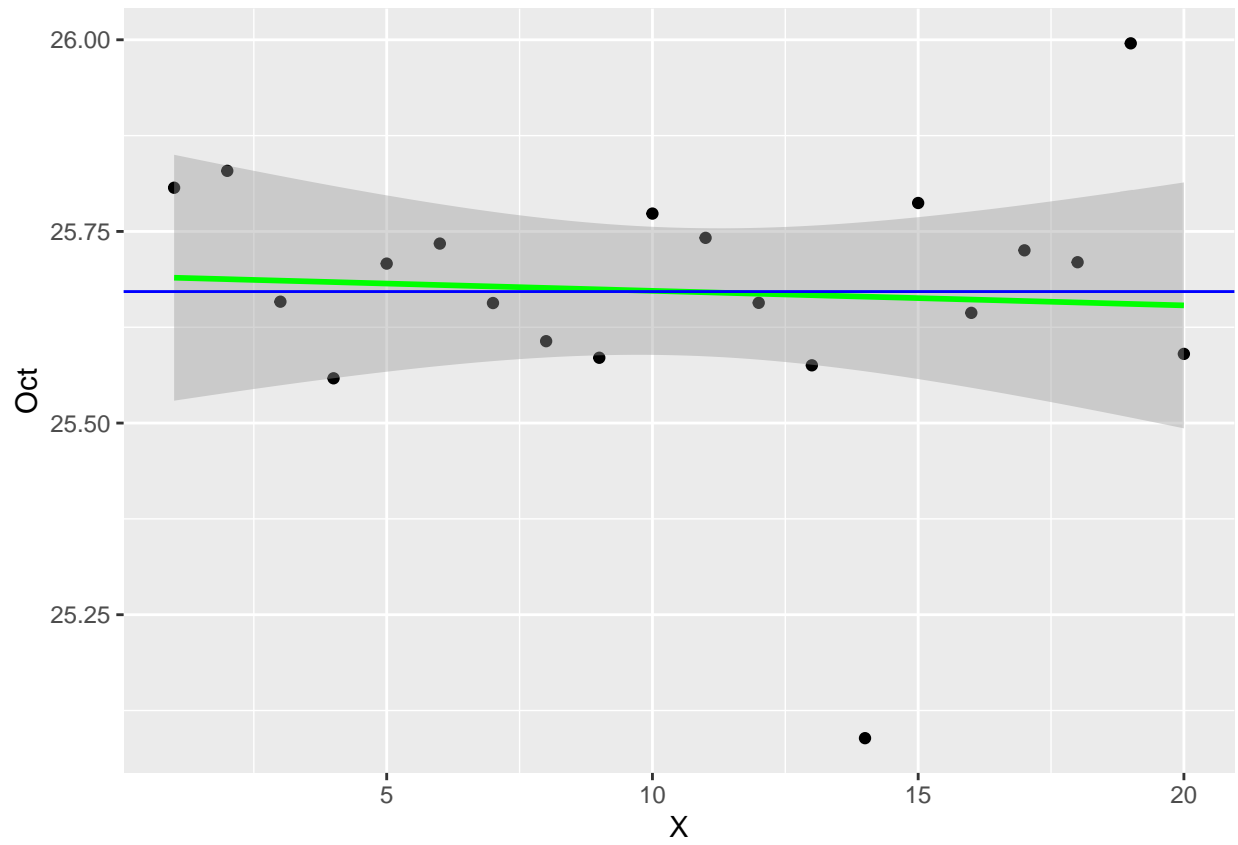
```
## `geom_smooth()` using formula 'y ~ x'
```

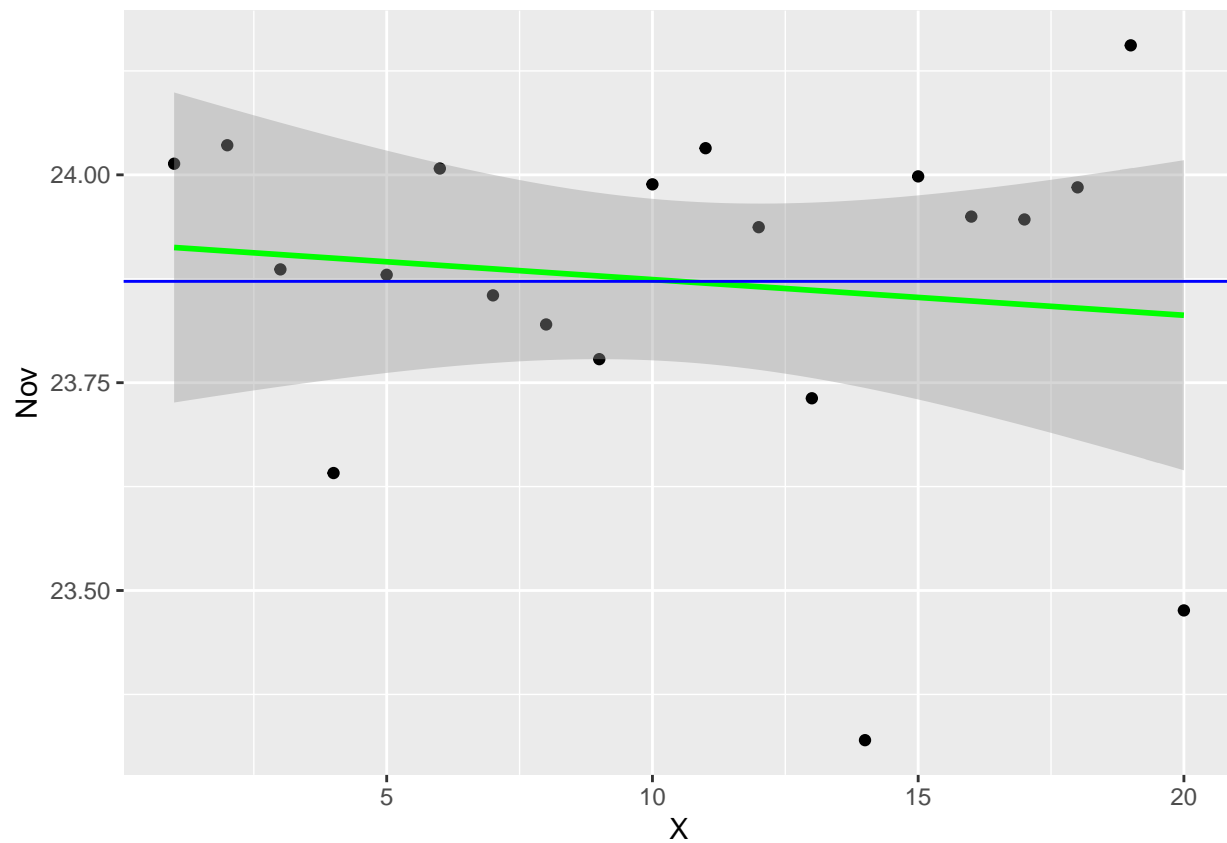
```
## `geom_smooth()` using formula 'y ~ x'
```



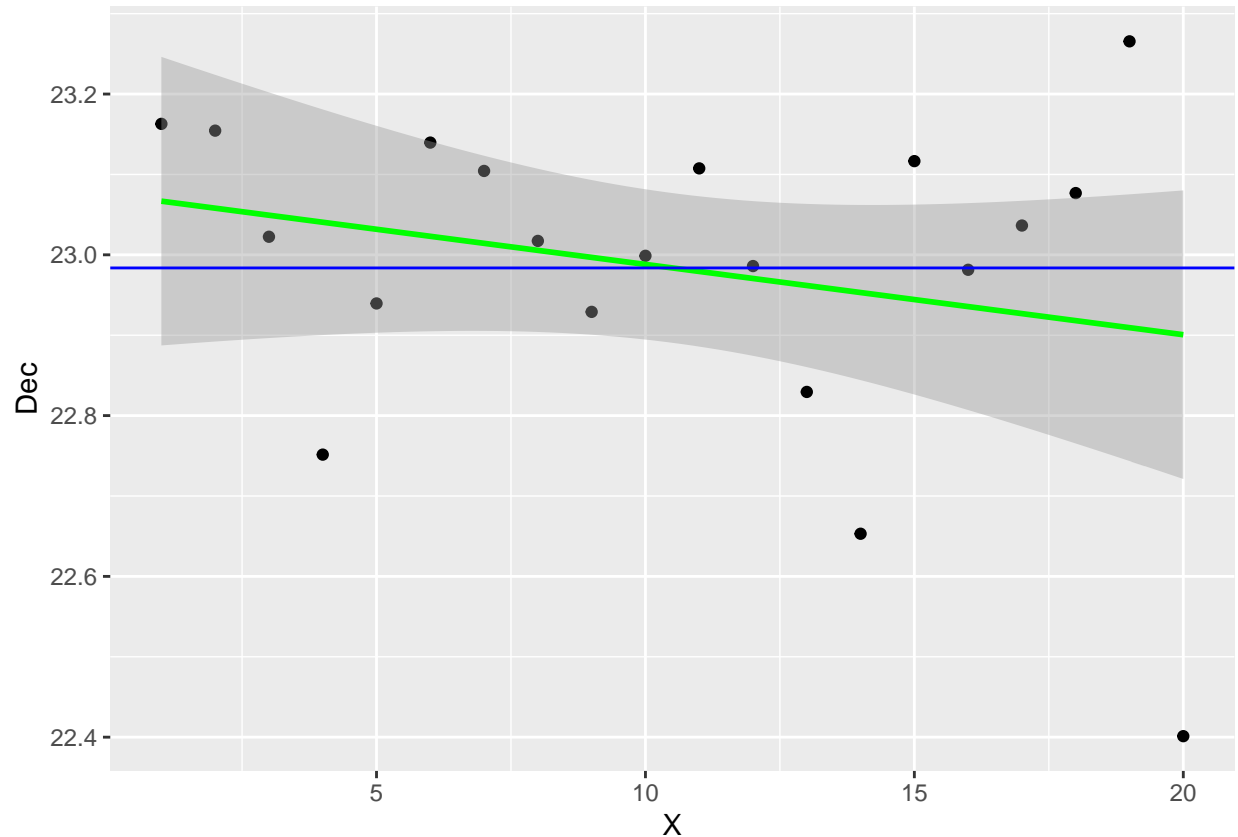
```
## `geom_smooth()` using formula 'y ~ x'
```



```
## `geom_smooth()` using formula 'y ~ x'
```



```
## `geom_smooth()` using formula 'y ~ x'
```

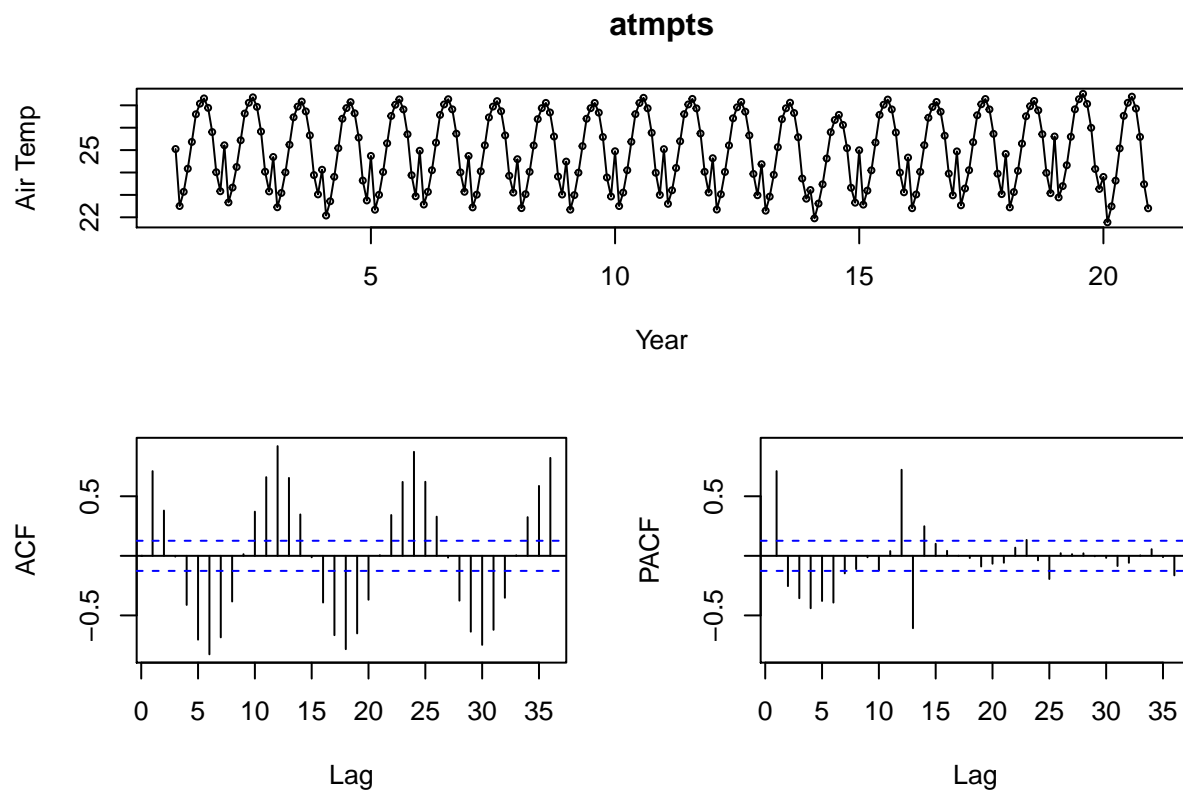


The above figures show the average air temperature of each month from 1997 to 2016. In each figure, the green line is the fitting result of the temperature regression to the year. It can be compared with the blue horizontal line, which is the average temperature of this month. From the figure, it is found that the monthly average air temperature in summer (Jul and Aug) shows a slight upward trend, while the monthly average air temperature in the rest of the year has a slight downward trend.

5. Seasonal ARIMA

#Data preparation: Read the monthly average temperature for 20 years and process it into a chronological vector. The first element of the vector is the average temperature in January 1997, and the last element is the average temperature in December 2016. Convert it into time series format using ts function.

```
atmp=read.csv(file="atmp.csv",header=TRUE)
atmpt=t(atmp)
atmpt1=atmpt[-c(1,2),]
atmpv=as.vector(atmpt1)
atmpts=ts(atmpv,frequency=12)
tsdisplay(atmpts,xlab="Year",ylab="Air Temp")
```



```
write.csv(atmpv,file="C:/Users/lenovo/Desktop/615 R/buoy/atmpv.csv")
```

```
#White Noise Test & Stationary Test
```

```
Box.test(atmpts)
```

```
##
```

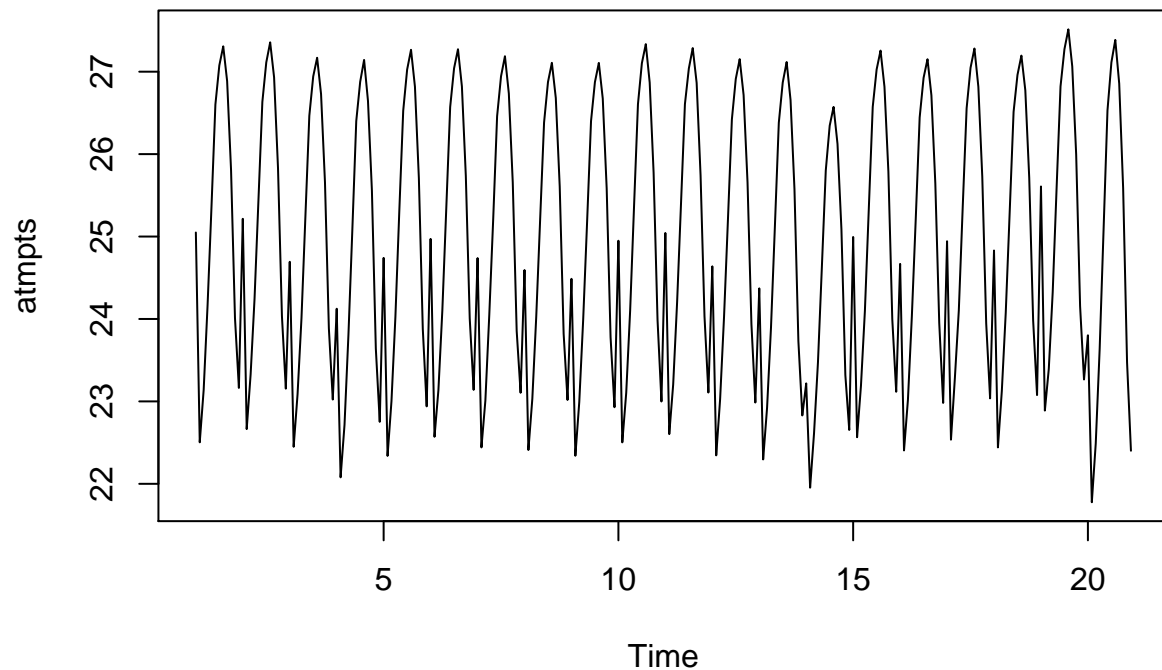
```
## Box-Pierce test
```

```
##
```

```
## data: atmpts
```

```
## X-squared = 121.3, df = 1, p-value < 2.2e-16
```

```
plot.ts(atmpts)
```



```
adf.test(attempts)
```

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

```
##
```

```
## Augmented Dickey-Fuller Test
```

```
##
```

```
## data: attempts
```

```
## Dickey-Fuller = -13.572, Lag order = 6, p-value = 0.01
```

```
## alternative hypothesis: stationary
```

Use Box.test function to complete the white noise test. The result (p-value is less than 0.05) shows that the sequence is correlated, therefore it is a non-white noise sequence. This time series can be used for modeling. According to the time series chart, the series has no obvious trend. Use adf.test function to complete the stationary test. The result (p-value is less than 0.05) shows the sequence is stationary.

```
#Seasonal ARIMA
```

```
#Fit the Time Series
```

```
fit1=auto.arima(attempts)
```

```
print(fit1)
```

```
## Series: attempts
```

```
## ARIMA(2,0,0)(2,1,0)[12]
```

```
##
```

```
## Coefficients:
```

```
##          ar1      ar2      sar1      sar2
```

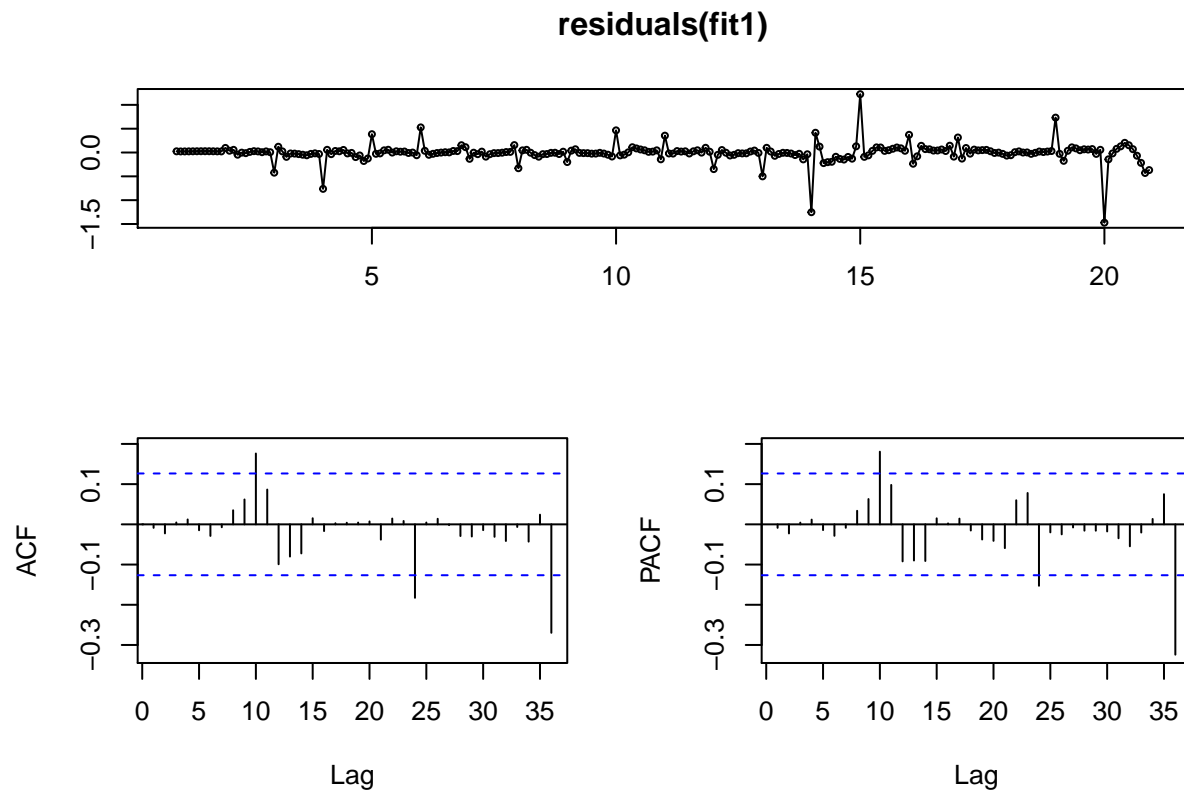
```
##          0.5639  0.2100 -0.6818 -0.2303
```

```
## s.e.    0.0652  0.0664  0.0759  0.0755
```

```
##  
## sigma^2 estimated as 0.04313: log likelihood=33.77  
## AIC=-57.54 AICc=-57.27 BIC=-40.4
```

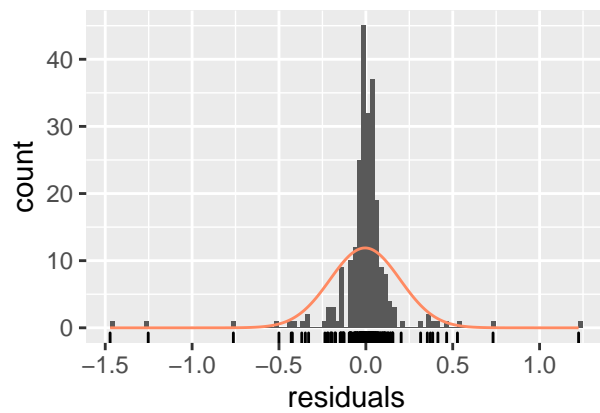
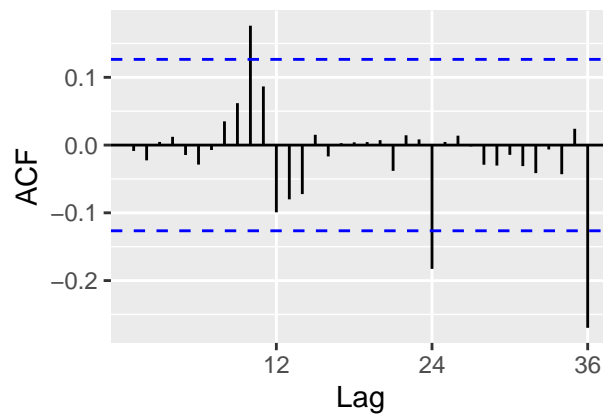
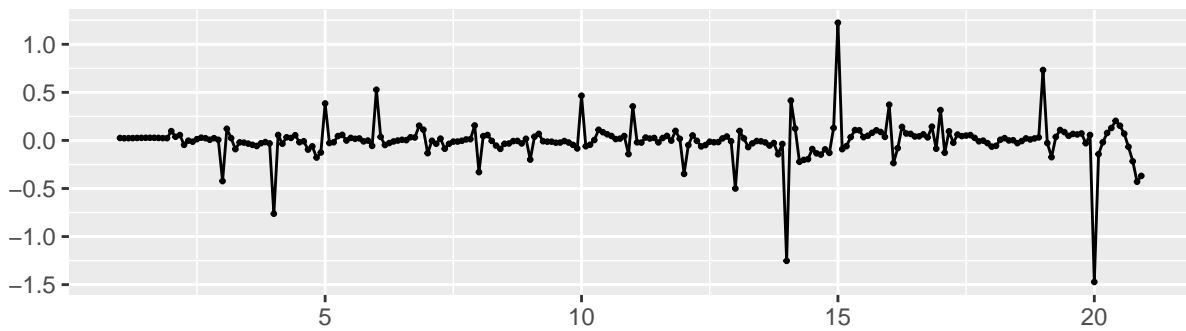
```
#Check the Residuals
```

```
tsdisplay(residuals(fit1))
```



```
checkresiduals(fit1)
```

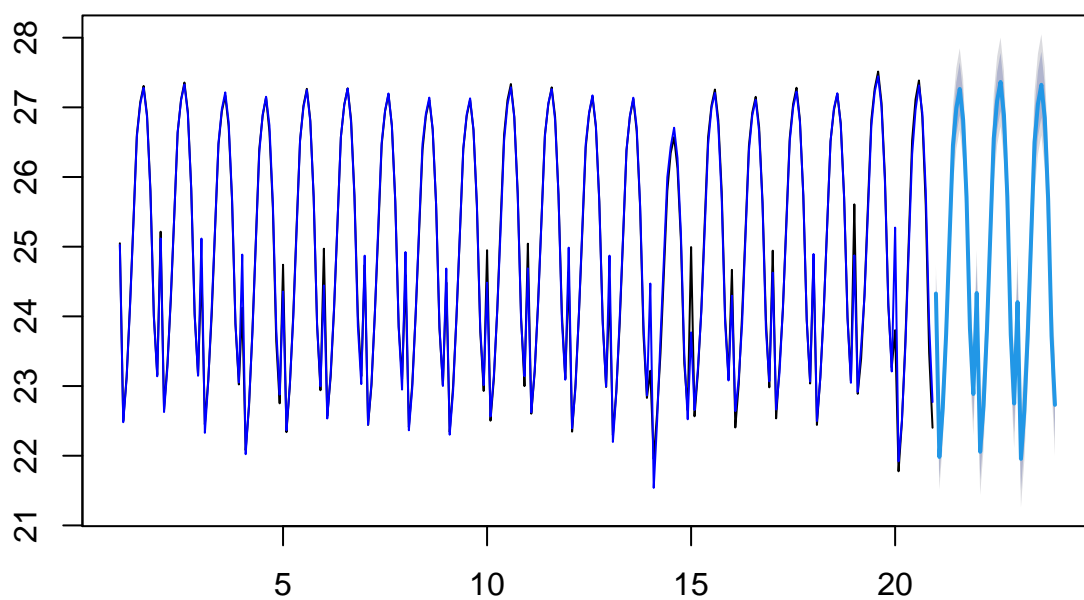

Residuals from ARIMA(2,0,0)(2,1,0)[12]



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,0,0)(2,1,0)[12]
## Q* = 26.551, df = 20, p-value = 0.1484
##
## Model df: 4.   Total lags used: 24
```

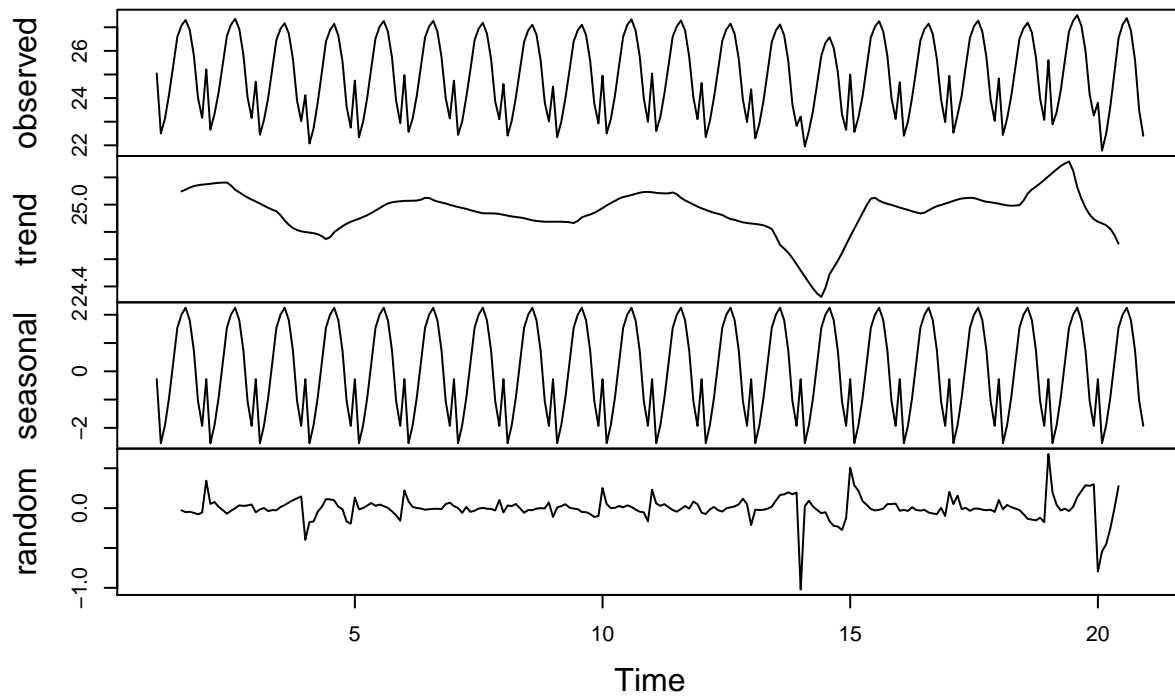
```
#Predict Air Temperature of the Next Three Years
fore1=forecast(fit1,h=3*12)
plot(fore1)
lines(fore1$fitted,col="blue")
lines(atmp,col="yellow")
```

Forecasts from ARIMA(2,0,0)(2,1,0)[12]

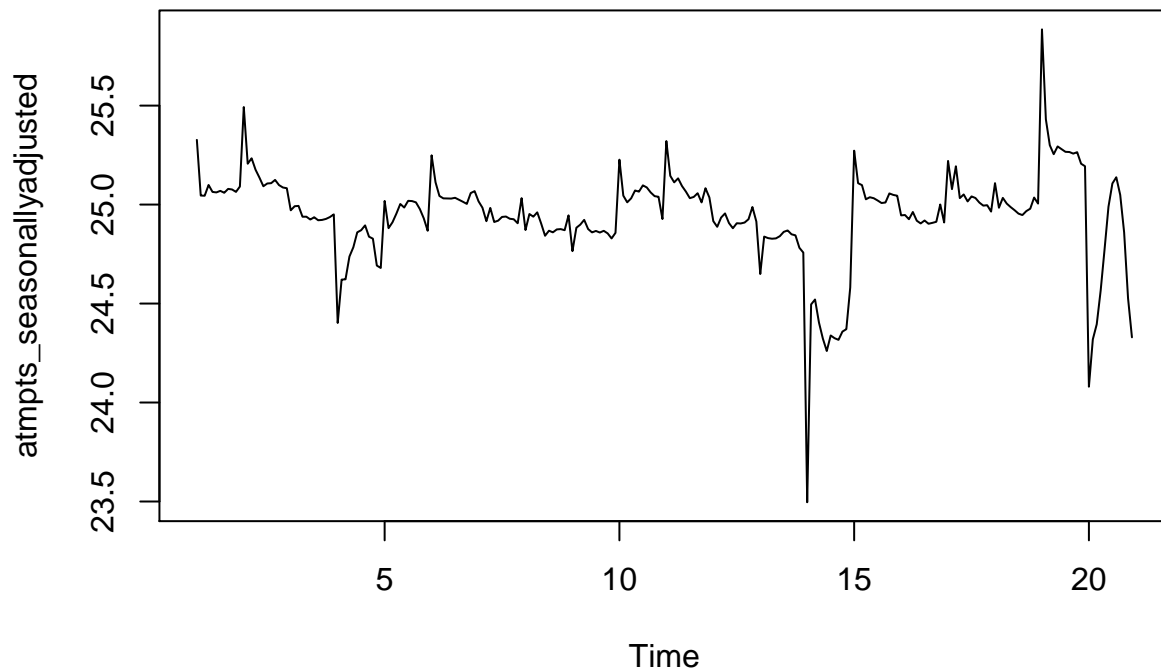


```
#Eliminate Seasonal Effects  
atmpts_components=decompose(atmpts)  
plot(atmpts_components)
```

Decomposition of additive time series



```
atmpts_seasonallyadjusted=atmpts-atmpts_components$seasonal  
plot(atmpts_seasonallyadjusted)
```

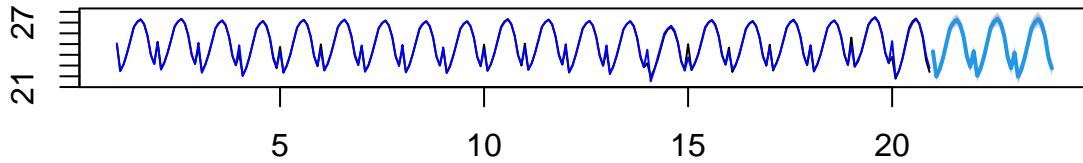


```
fit2=auto.arima(atmpts_seasonallyadjusted)
print(fit2)

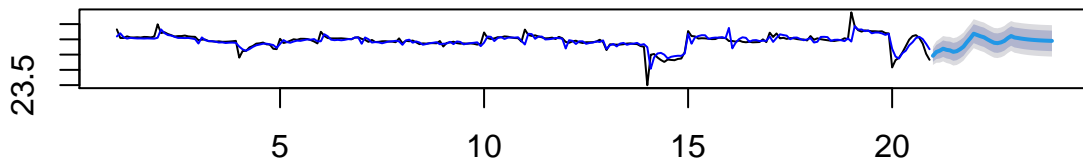
## Series: atmpts_seasonallyadjusted
## ARIMA(2,0,0)(0,0,2)[12] with non-zero mean
##
## Coefficients:
##          ar1      ar2      sma1      sma2      mean
##          0.6096  0.1882 -0.4492 -0.3152  24.9321
## s.e.      0.0650  0.0657  0.0698  0.0666  0.0149
##
## sigma^2 estimated as 0.02364: log likelihood=106.31
## AIC=-200.62 AICc=-200.26 BIC=-179.74

fore2=forecast(fit2,h=3*12)
par(mfrow=c(2,1))
plot(fore1)
lines(fore1$fitted,col="blue")
lines(atmp,col="yellow")
plot(fore2)
lines(fore2$fitted,col="blue")
lines(atmp,col="yellow")
```

Forecasts from ARIMA(2,0,0)(2,1,0)[12]



Forecasts from ARIMA(2,0,0)(0,0,2)[12] with non-zero mean



Use the `auto.arima` function to establish a Seasonal ARIMA model for the temperature time series, and get the parameter $(2,0,0)(2,1,0)[12]$. Observe the autocorrelation coefficient of the residual, except for the periodic (12) spike, the value of acf is basically within the range of the dotted line, which shows that the collinearity in the model generated by Seasonal ARIMA is not serious. The residuals are close to a normal distribution. Using this model to predict the monthly average temperature of the next three years, the fluctuation range has not changed much compared with the past, and three cycles are clear. Use `decompose` function (type is additive) to separate temperature into “random”, “seasonal”, “trend”, and “observed”. After removing seasonal factors, the periodicity of the time series is no longer obvious. At the same time, the “trend” in the `decompose` figure can also confirm that the time series is stationary. Use the `auto.arima` function to establish a Seasonal ARIMA model for new sequence. The parameter this time is $(2,0,0)(0,0,2)[12]$.

6. Conclusion

According to the result of data analysis & visualization and Seasonal ARIMA model, I would like to draw the conclusion that there may not be an evidence of global warming in the data collected by a single weather buoy in the NOAA National Data Buoy Center.