ENV 797 - Time Series Analysis for Energy and Environment Applications | Spring 2024

Assignment 7 - Due date 03/07/24

Ina Liao

Directions

You should open the .rmd file corresponding to this assignment on RStudio. The file is available on our class repository on Github. And to do so you will need to fork our repository and link it to your RStudio.

Once you have the file open on your local machine the first thing you will do is rename the file such that it includes your first and last name (e.g., "LuanaLima_TSA_A07_Sp24.Rmd"). Then change "Student Name" on line 4 with your name.

Then you will start working through the assignment by **creating code and output** that answer each question. Be sure to use this assignment document. Your report should contain the answer to each question and any plots/tables you obtained (when applicable).

When you have completed the assignment, **Knit** the text and code into a single PDF file. Submit this pdf using Sakai.

Packages needed for this assignment: "forecast","tseries". Do not forget to load them before running your script, since they are NOT default packages.\

Set up

```
library(forecast)
library(tseries)
library(ggplot2)
library(Kendall)
library(lubridate)
library(tidyverse)
library(here)
library(knitr)
library(ggthemes)
library(cowplot)
library(dplyr)
#install.packages("sarima")
library(sarima)
#install.packages('tinytex')
#remotes::install_github('rstudio/tinytex')
#update.packages(ask = FALSE, checkBuilt = TRUE)
#tinytex::tlmgr_update()
#tinytex::reinstall_tinytex()
```

Importing and processing the data set

Consider the data from the file "Net_generation_United_States_all_sectors_monthly.csv". The data corresponds to the monthly net generation from January 2001 to December 2020 by source and is provided by the US Energy Information and Administration. **You will work with the natural gas column only**.

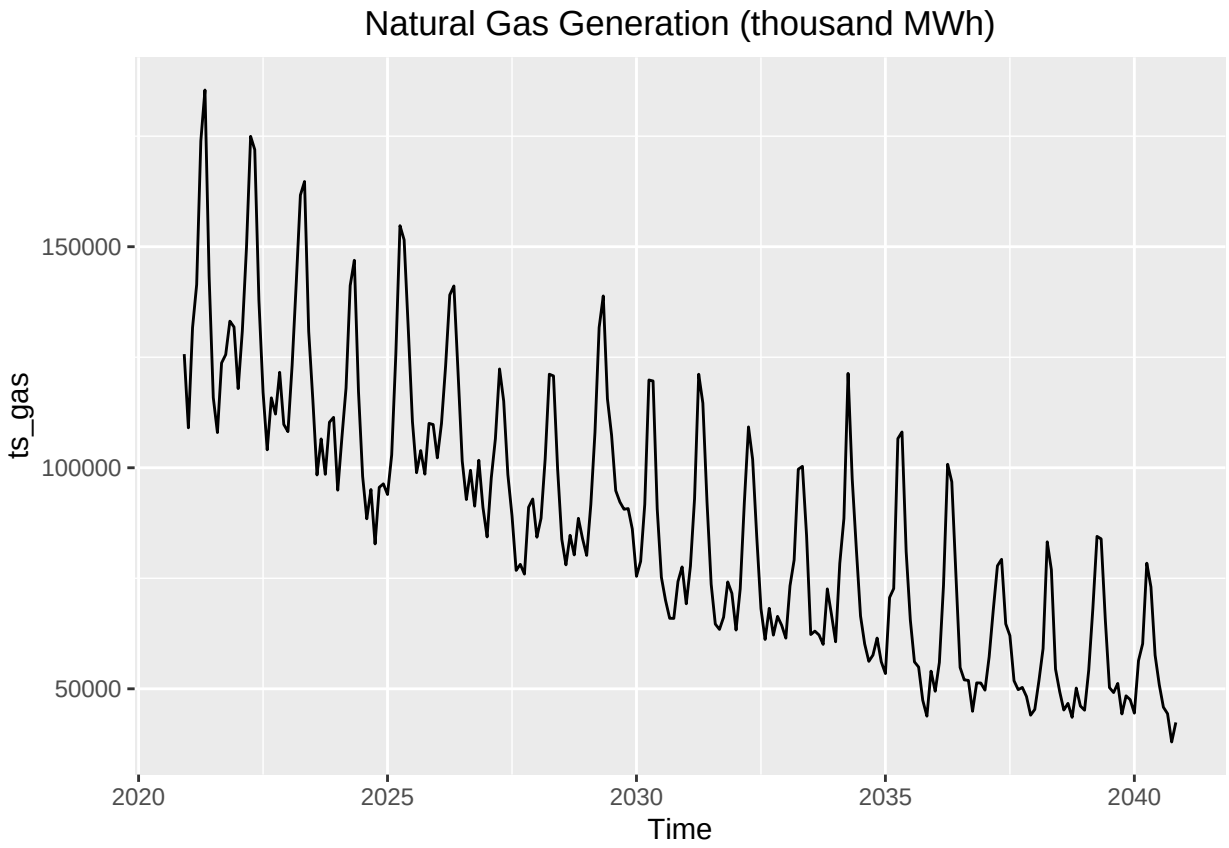
Q1

Import the csv file and create a time series object for natural gas. Make you sure you specify the **start=** and **frequency=** arguments. Plot the time series over time, ACF and PACF.

```
raw_generation<-read.csv(
  here("Data/Net_generation_United_States_all_sectors_monthly.csv"),
  skip=4,header=TRUE)

#change the format of date
Date<-my(raw_generation$Month)
raw_generation<-cbind(Date, raw_generation[,2:6])

#select needed columns
df_gas<-raw_generation[,c(1,4)]
```

```
#create time series object
year1<-year(df_gas$Date[1])
month1<-month(df_gas$Date[1])
ts_gas<-ts(df_gas[,2],start=c(year1,month1),frequency=12) #monthly data

#plot time series
autoplot(ts_gas,lag=40)+
  labs(title="Natural Gas Generation (thousand MWh)")+
  theme(plot.title=element_text(color="black",hjust=0.5,vjust=1))
```

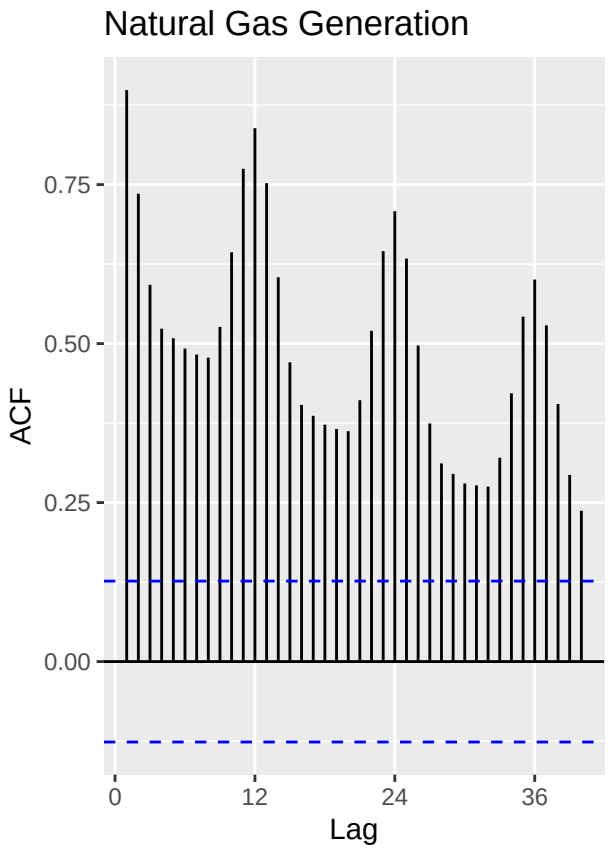
```
## Warning in ggplot2::geom_line(na.rm = TRUE, ...): Ignoring unknown parameters:
## `lag`
```

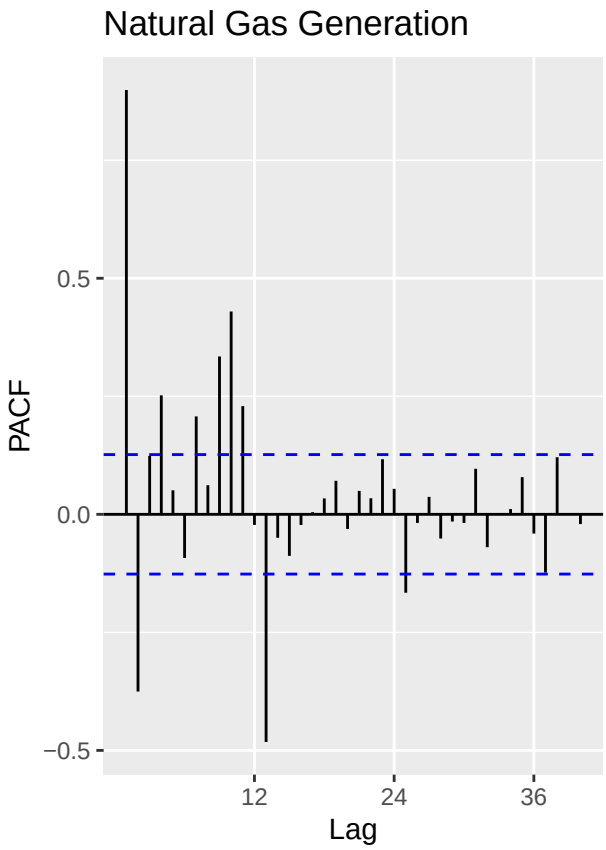



```
#plot ACF and PACF
plot_grid(
  autoplot(Acf(ts_gas,lag=40, plot=FALSE),main="Natural Gas Generation"),
  autoplot(Pacf(ts_gas,lag=40, plot=FALSE),main="Natural Gas Generation")
)
```

```
## Warning in ggplot2::geom_segment(lineend = "butt", ...): Ignoring unknown
## parameters: `main`
```

```
## Warning in ggplot2::geom_segment(lineend = "butt", ...): Ignoring unknown
## parameters: `main`
```

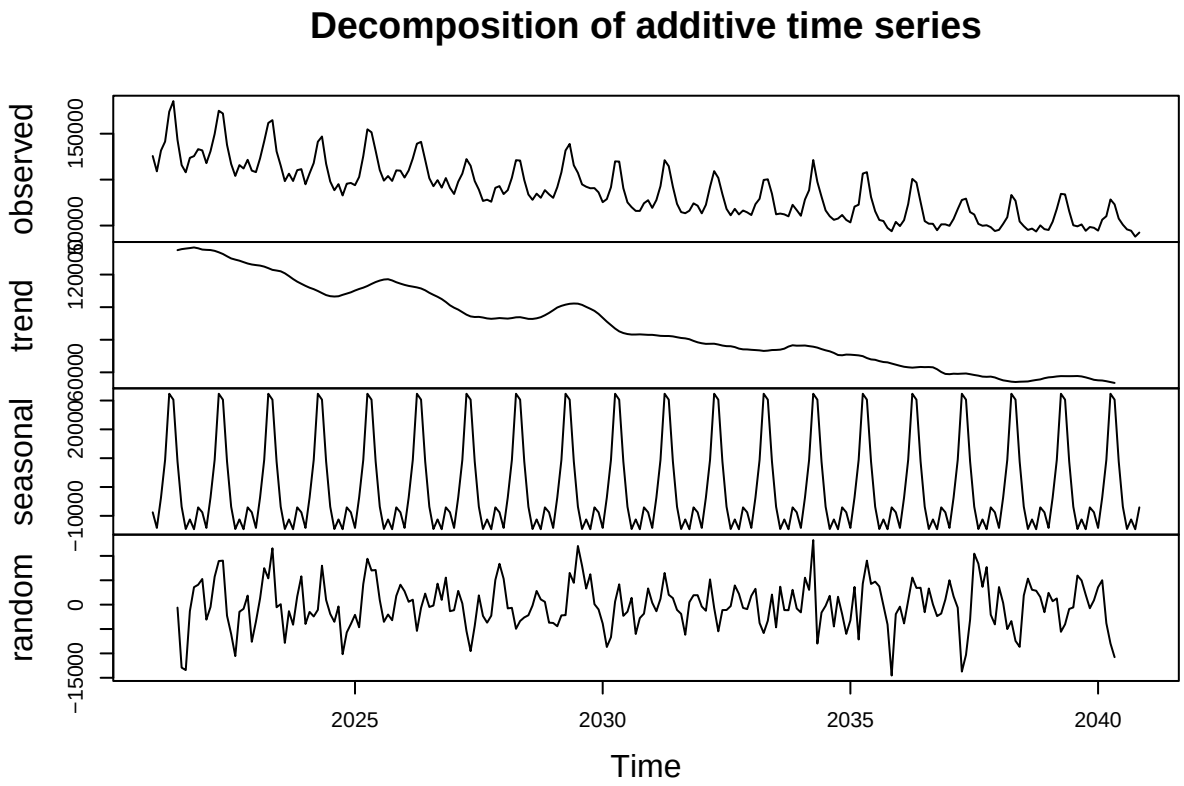




Q2

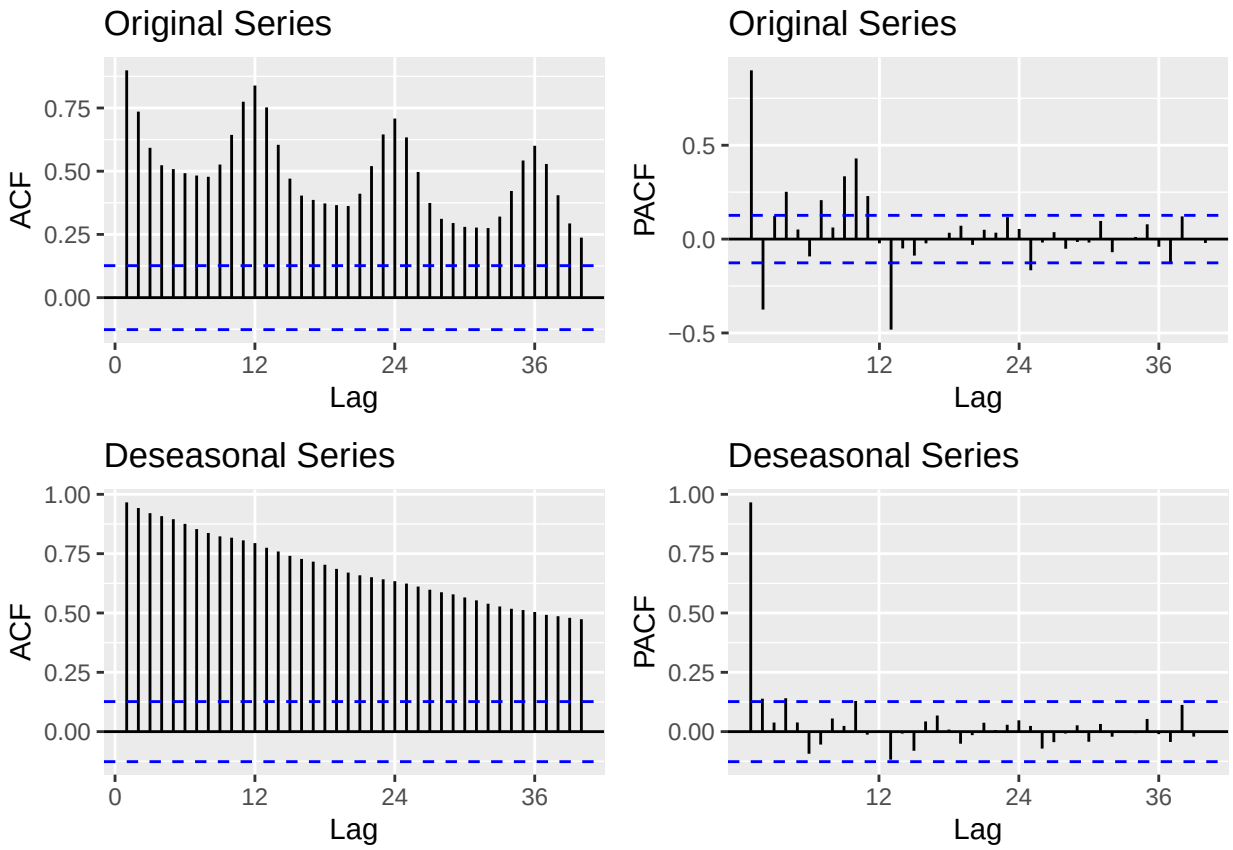
Using the *decompose*() or *stl*() and the *seasadj*() functions create a series without the seasonal component, i.e., a deseasonalized natural gas series. Plot the deseasonalized series over time and corresponding ACF and PACF. Compare with the plots obtained in Q1.

```
decompose_gas<-decompose(ts_gas,"additive")
plot(decompose_gas)
```




```
#create non-seasonal series
deseason_gas<-seasadj(decompose_gas)

#plot the ACF and PACF
plot_grid(
  autoplot(Acf(ts_gas,lag = 40, plot=FALSE),main="Original Series"),
  autoplot(Pacf(ts_gas,lag = 40, plot=FALSE),main="Original Series"),
  autoplot(Acf(deseason_gas,lag = 40, plot=FALSE),main="Deseasonal Series"),
  autoplot(Pacf(deseason_gas,lag = 40, plot=FALSE),main="Deseasonal Series")
)
```

```
## Warning in ggplot2::geom_segment(lineend = "butt", ...): Ignoring unknown parameters: `main`
## Ignoring unknown parameters: `main`
## Ignoring unknown parameters: `main`
## Ignoring unknown parameters: `main`
```



Modeling the seasonally adjusted or deseasonalized series

Q3

Run the ADF test and Mann Kendall test on the deseasonalized data from Q2. Report and explain the results.

```
#Mann-Kendall
trend::mk.test(deseason_gas) #p-value < 2.2e-16
```

```
##
##  Mann-Kendall trend test
##
## data:  deseason_gas
## z = -19.454, n = 240, p-value < 2.2e-16
## alternative hypothesis: true S is not equal to 0
## sample estimates:
##             S          varS           tau
## -2.418600e+04  1.545533e+06 -8.433054e-01
```

```
#ADF test
adf.test(deseason_gas,alternative = "stationary") #p-value = 0.01
```

```
## Warning in adf.test(deseason_gas, alternative = "stationary"): p-value smaller
## than printed p-value
```

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

```
## data:  deseason_gas
## Dickey-Fuller = -4.0574, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

The Mann-Kendall test result suggests that there is a significant decreasing trend in the time series data (p-value<0.05, and tau<0). The ADF test result rejects the null hypothesis of a unit root in the time series, meaning that the time series is stationary (p-value<0.05).

Q4

Using the plots from Q2 and test results from Q3 identify the ARIMA model parameters p, d and q. Note that in this case because you removed the seasonal component prior to identifying the model you don't need to worry about seasonal component. Clearly state your criteria and any additional function in R you might use. DO NOT use the *auto.arima*() function. You will be evaluated on ability to understand the ACF/PACF plots and interpret the test results.

> Answer: The ACF plot shows a slow decay, and the PACF plot has a clear cutoff at lag 2, suggesting that an AR process would be suitable for modeling the time series data.

```r
#012
#find out how many time we need to difference
n_diff <- ndiffs(deseason_gas)
cat("Number of differencing needed: ",n_diff)
```

```
## Number of differencing needed:  1
```

```r
#p=1, d=1, q=0
Model1<-Arima(deseason_gas,order=c(1,1,0),include.drift=TRUE)
print(Model1)
```

```
## Series: deseason_gas
## ARIMA(1,1,0) with drift
##
## Coefficients:
##           ar1      drift
##       -0.1479  -348.3913
## s.e.   0.0644   308.8367
##
## sigma^2 = 30254130:  log likelihood = -2396.54
## AIC=4799.07   AICc=4799.18   BIC=4809.5
```

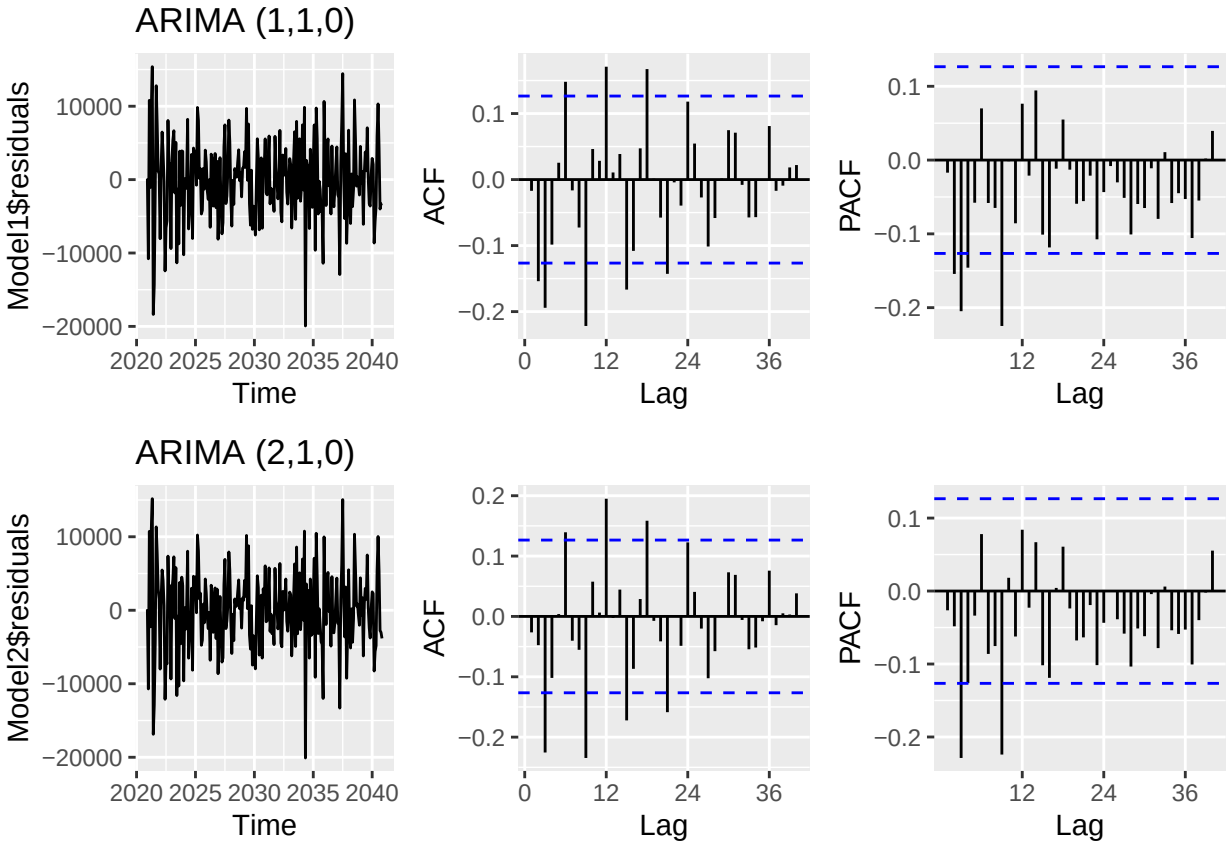
```r
#p=2, d=1, q=0
Model2<-Arima(deseason_gas,order=c(2,1,0),include.drift=TRUE)
print(Model2)
```

```
## Series: deseason_gas
## ARIMA(2,1,0) with drift
##
## Coefficients:
##           ar1      ar2      drift
##       -0.1647  -0.1283  -346.9085
## s.e.   0.0645   0.0650   272.1655
##
## sigma^2 = 29891484:  log likelihood = -2394.61
## AIC=4797.21   AICc=4797.39   BIC=4811.12
```

```r
#plot ACF and PACF
plot_grid(
```

```
  autoplot(Model1$residuals,main="ARIMA (1,1,0)"),
  autoplot(Acf(Model1$residuals,lag.max=40, plot = FALSE),main=""),
  autoplot(Pacf(Model1$residuals,lag.max=40, plot = FALSE),main=""),
  autoplot(Model2$residuals,main="ARIMA (2,1,0)"),
  autoplot(Acf(Model2$residuals,lag.max=40, plot = FALSE),main=""),
  autoplot(Pacf(Model2$residuals,lag.max=40, plot = FALSE),main=""),
  nrow=2
)
```

```
## Warning in ggplot2::geom_segment(lineend = "butt", ...): Ignoring unknown parameters: `main`
## Ignoring unknown parameters: `main`
## Ignoring unknown parameters: `main`
## Ignoring unknown parameters: `main`
```



Q5

Use `Arima()` from package "forecast" to fit an ARIMA model to your series considering the order estimated in Q4. You should allow constants in the model, i.e., `include.mean = TRUE` or `include.drift=TRUE`. **Print the coefficients** in your report. Hint: use the `cat()` r `print()` function to print.

```
#p=1, d=1, q=0
#feed the order with the best performance
Model<-Arima(deseason_gas,order=c(1,1,0),include.drift=TRUE)

#store the model coefficients
coefficients<-coef(Model)

print(paste("coefficients for the autoregressive term",round(coefficients[1],4)))
```

```
## [1] "coefficients for the autoregressive term -0.1479"
```

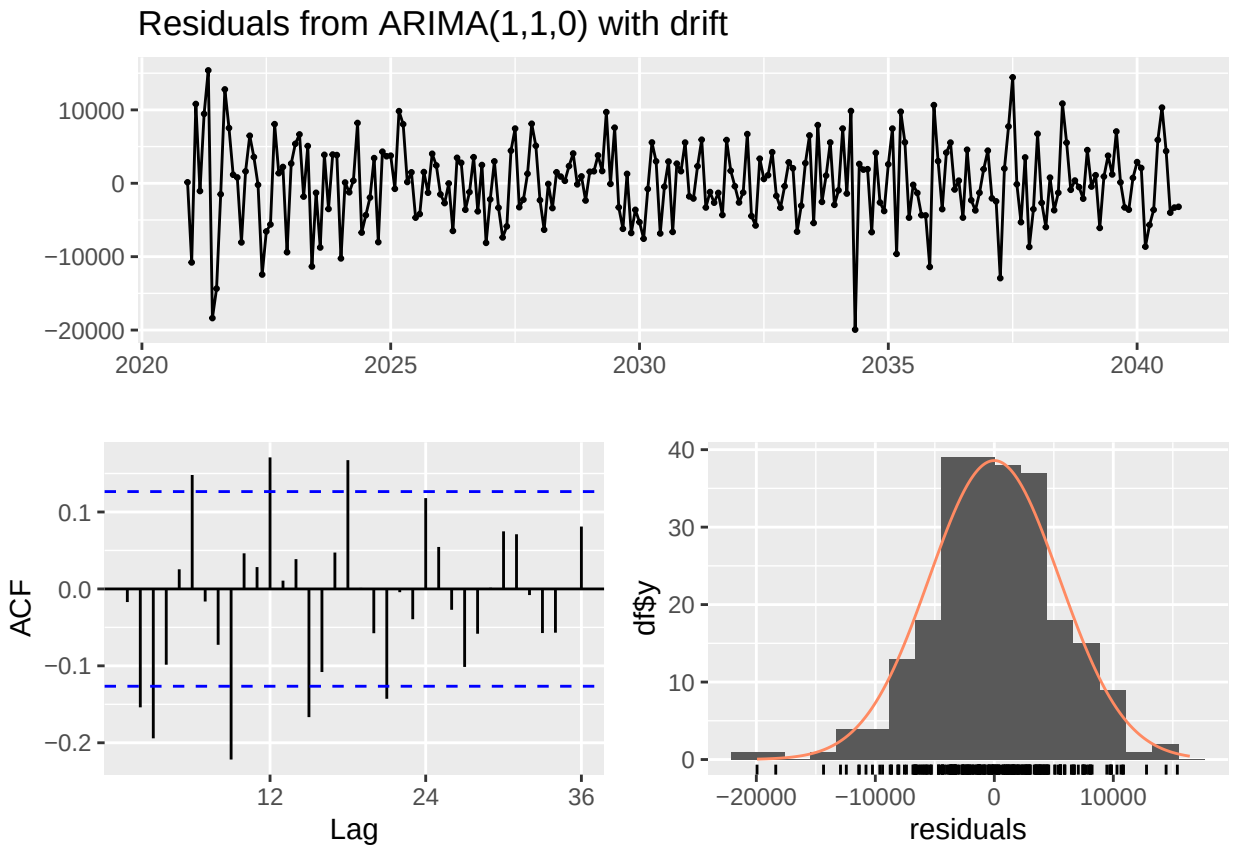
```
print(paste("coefficients for the drifting term",round(coefficients[2],4)))
```

```
## [1] "coefficients for the drifting term -348.3913"
```

Q6

Now plot the residuals of the ARIMA fit from Q5 along with residuals ACF and PACF on the same window. You may use the *checkresiduals*() function to automatically generate the three plots. Do the residual series look like a white noise series? Why?

```
#check the residuals
Model_res<-checkresiduals(Model)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,0) with drift
## Q* = 73.991, df = 23, p-value = 2.899e-07
##
## Model df: 1.   Total lags used: 24
```

It seems that the mean of the residuals is fluctuating around zero, with a few spikes observed during the time periods of 2020-2025 and 2035-2040. Additionally, the ACF plot indicates that the seasonality has been removed as the coefficient values at lag 12, 24, and 36 have decreased. Moreover, the residuals of a time series model follow a normal distribution, suggesting that the model capture the underlying patterns and variations in the data.

Modeling the original series (with seasonality)

Q7

Repeat Q4-Q6 for the original series (the complete series that has the seasonal component). Note that when you model the seasonal series, you need to specify the seasonal part of the ARIMA model as well, i.e., P, D and Q.

> Answer: In the original time series, the ACF plot shows a slow decay, and the PACF plot has a clear cutoff at lag 1 and a spike at lag 12, suggesting that P=1. Given the constraints of P+Q = 1, Q should be 0.

```
#d+D less than or equal to 2: No drift term fitted as the order of difference is 2 or more
Model1_season<-Arima(ts_gas,order=c(1,1,0),seasonal=c(1,1,1),include.drift=FALSE)
print(Model1_season) #minimum AIC
```

```
## Series: ts_gas
## ARIMA(1,1,0)(1,1,1)[12]
##
## Coefficients:
##           ar1     sar1     sma1
##       -0.1819  -0.0400  -0.6674
## s.e.   0.0655   0.0979   0.0813
##
## sigma^2 = 30744866:  log likelihood = -2281.34
## AIC=4570.69   AICc=4570.87   BIC=4584.39
```

```
Model2_season<-Arima(ts_gas,order=c(1,1,0),seasonal=c(1,0,1),include.drift=TRUE)
print(Model2_season)
```

```
## Series: ts_gas
## ARIMA(1,1,0)(1,0,1)[12] with drift
##
## Coefficients:
##           ar1    sar1     sma1      drift
##       -0.1551  0.9928  -0.6950  -408.6346
## s.e.   0.0650  0.0038   0.0566  3422.9828
##
## sigma^2 = 30750412:  log likelihood = -2412.87
## AIC=4835.75   AICc=4836.01   BIC=4853.13
```

```
Model3_season<-Arima(ts_gas,order=c(2,1,0),seasonal=c(1,1,1),include.drift=FALSE)
print(Model3_season)
```

```
## Series: ts_gas
## ARIMA(2,1,0)(1,1,1)[12]
##
## Coefficients:
##           ar1      ar2     sar1     sma1
##       -0.2101  -0.1643  -0.0026  -0.6725
## s.e.   0.0658   0.0671   0.1023   0.0836
##
## sigma^2 = 30143008:  log likelihood = -2278.39
## AIC=4566.78   AICc=4567.05   BIC=4583.91
```

```
Model4_season<-Arima(ts_gas,order=c(2,1,0),seasonal=c(1,0,1),include.drift=TRUE)
print(Model4_season)
```

```
## Series: ts_gas
## ARIMA(2,1,0)(1,0,1)[12] with drift
##
## Coefficients:
##           ar1      ar2    sar1     sma1      drift
##       -0.1800  -0.1548  0.9926  -0.6794  -308.4634
## s.e.   0.0653   0.0648  0.0040   0.0594  3012.7479
##
## sigma^2 = 30126556:  log likelihood = -2410.06
## AIC=4832.12   AICc=4832.48   BIC=4852.98
```

Q8

Compare the residual series for Q7 and Q6. Can you tell which ARIMA model is better representing the Natural Gas Series? Is that a fair comparison? Explain your response.

> Answer: The results obtained from Q7 and Q6 cannot be compared with each other. This is because Q6's non-seasonal ARIMA model only considers the trend of the time series and does not account for any seasonal variations. In contrast, seasonal ARIMA takes into account both the trend and seasonal components of the time series, making it a better representation of the original data.

Checking your model with the auto.arima()

Please do not change your answers for Q4 and Q7 after you ran the *auto.arima*(). It is **ok** if you didn't get all orders correctly. You will not loose points for not having the same order as the *auto.arima*().

Q9

Use the *auto.arima*() command on the **deseasonalized series** to let R choose the model parameter for you. What's the order of the best ARIMA model? Does it match what you specified in Q4?

```
Model_auto_deseason<-auto.arima(deseason_gas)
print(Model_auto_deseason)
```

```
## Series: deseason_gas
## ARIMA(3,1,0)(1,0,1)[12] with drift
##
## Coefficients:
##           ar1      ar2      ar3    sar1     sma1      drift
##       -0.2028  -0.1851  -0.1378  0.6609  -0.4698  -331.8138
## s.e.   0.0645   0.0655   0.0682  0.1918   0.2120   328.8253
##
## sigma^2 = 27791547:  log likelihood = -2384.89
## AIC=4783.79   AICc=4784.27   BIC=4808.12
```

> Answer: The auto ARIMA result indicates ARIMA(3,1,0)(1,0,1)[12] with drift, which is not consistent with the order of the best-performing model in Q4. This suggests that relying solely on ACF and PACF plots to determine the order of parameters in the ARIMA model may not be the most effective approach.

Q10

Use the *auto.arima*() command on the **original series** to let R choose the model parameters for you. Does it match what you specified in Q7?

```r
Model_auto<-auto.arima(ts_gas)
print(Model_auto)
```

```
## Series: ts_gas
## ARIMA(2,0,1)(2,1,2)[12] with drift
##
## Coefficients:
##          ar1      ar2      ma1     sar1     sar2     sma1    sma2      drift
##       1.1650  -0.2834  -0.4837  -0.0667  -0.0785  -0.6371  0.0072  -357.8435
## s.e.  0.4164   0.3180   0.3993   1.3218   0.1014   1.3211  0.9212    44.1285
##
## sigma^2 = 27958165:  log likelihood = -2278.46
## AIC=4574.91   AICc=4575.74   BIC=4605.77
```

Answer: The auto ARIMA result indicates ARIMA(2,0,1)(2,1,2)[12] with drift, which is not consistent with the order of the best-performing model in Q7 either. This result also suggests that relying solely on ACF and PACF plots to determine the order of parameters in the ARIMA model may not be the most effective approach.