ENV 790.30 - Time Series Analysis for Energy Data | Spring 2023

Assignment 7 - Due date 03/20/23

Kexin(Kassie) Huang

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_Sp23.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.

Set up

```
#Load/install required package here
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(ggplot2)
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(Kendall)
library(tseries)
library(outliers)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --

## v tibble  3.1.8     v dplyr   1.1.0
## v tidyr   1.3.0     v stringr 1.5.0
## v readr   2.1.3     v forcats 1.0.0
## v purrr   1.0.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x lubridate::as.difftime() masks base::as.difftime()
## x lubridate::date()        masks base::date()
## x dplyr::filter()          masks stats::filter()
## x lubridate::intersect()   masks base::intersect()
## x dplyr::lag()             masks stats::lag()
## x lubridate::setdiff()     masks base::setdiff()
## x lubridate::union()       masks base::union()
```

```
library(cowplot)
```

```
##
## Attaching package: 'cowplot'
##
## The following object is masked from 'package:lubridate':
##
##     stamp
```

Importing and processing the data set

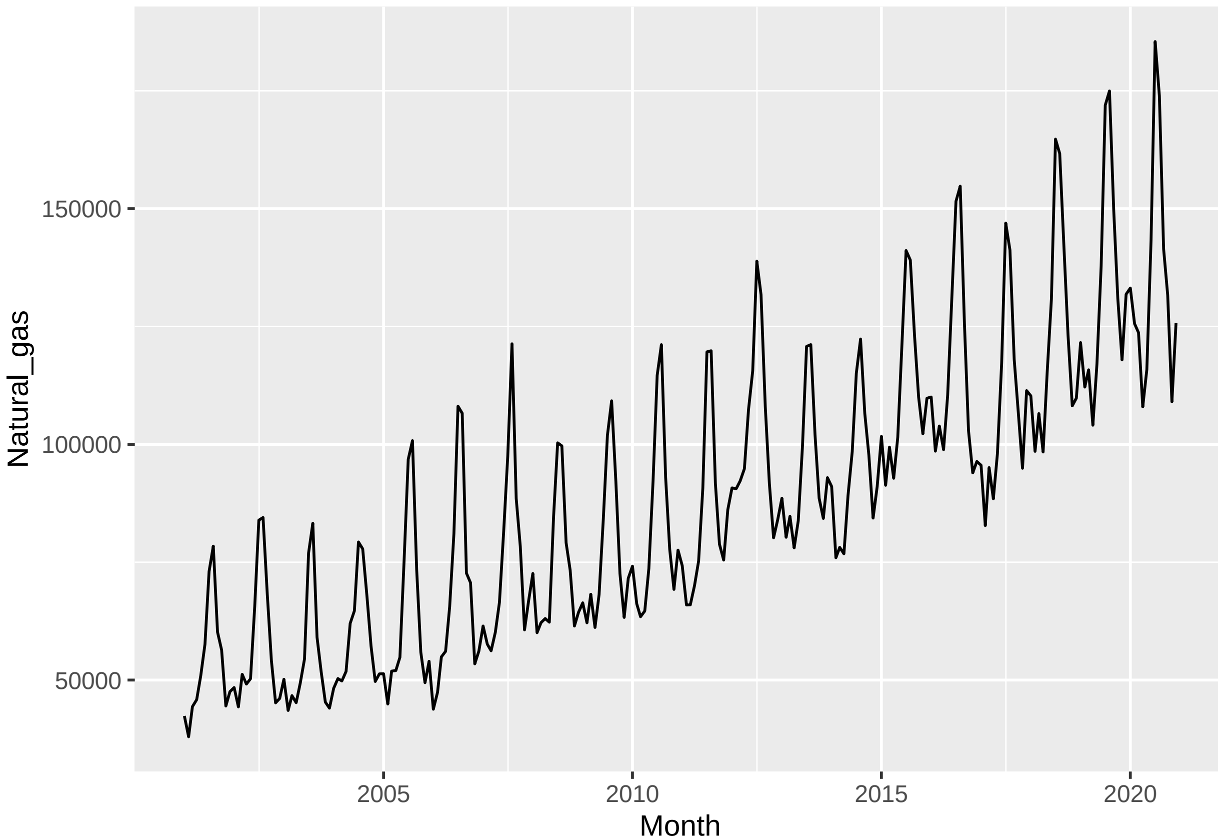
Consider the data from the file "Net_generation_United_States_all_sectors_monthly.csv". The data corresponds to the monthly net generation from January 2001 to December 2020 by source and is provided by the US Energy Information and Administration. **You will work with the natural gas column only**.

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

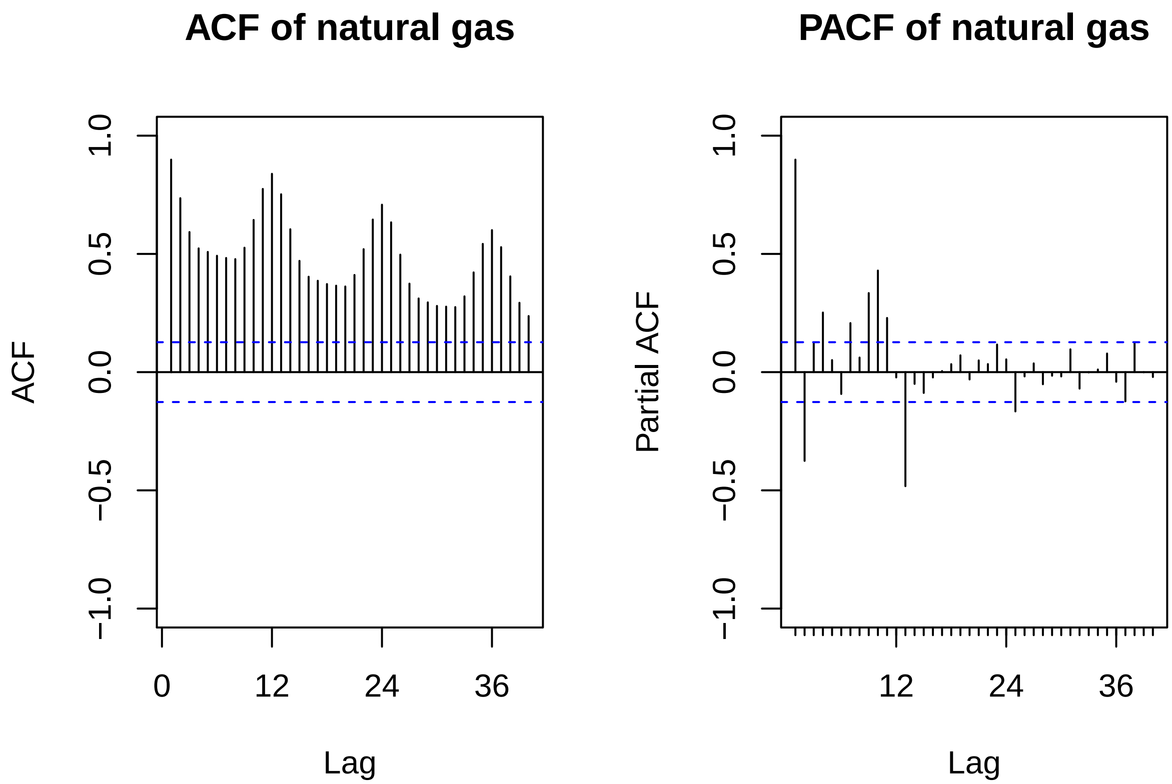
Q1

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.

```
#import the csv
read_data <- read.csv(file='./Data/Net_generation_United_States_all_sectors_monthly.csv',header = TRUE,

#fix the reversed order
data_processed <-
  data.frame(Month=my(read_data$Month), Natural_gas=read_data$natural.gas.thousand.megawatthours) %>%
  arrange(Month)
my_date <- data_processed$Month

#create a time series object
ts_natural_gas <- ts(
  data_processed$Natural_gas,
  start=c(year(my_date[1]),month(my_date[1])),
  frequency = 12)

#plot the series, ACF and PACF
plot(ggplot(data = data_processed, aes(x=Month,y=Natural_gas))+
  geom_line()
  )
```



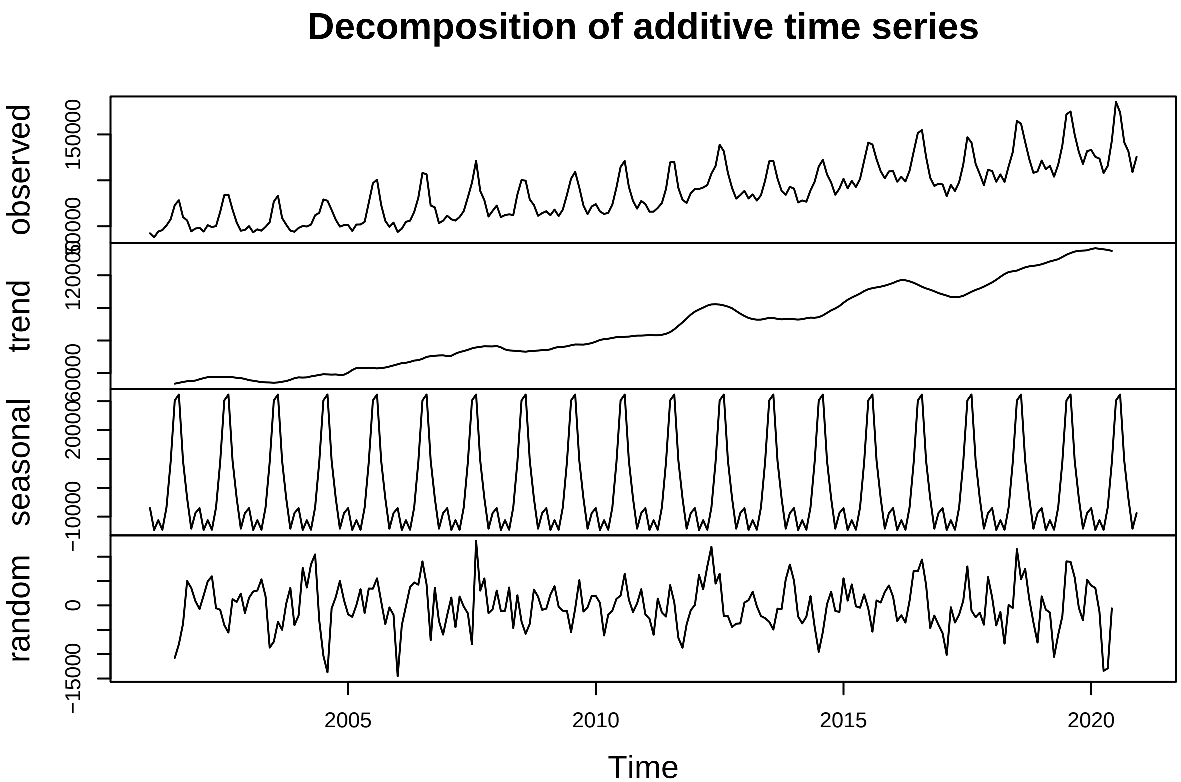

```
par(mfrow=c(1,2))
Acf(ts_natural_gas,lag.max = 40, ylim=c(-1,1),main="ACF of natural gas")
Pacf(ts_natural_gas,lag.max = 40, ylim=c(-1,1),main="PACF of natural gas")
```



Q2

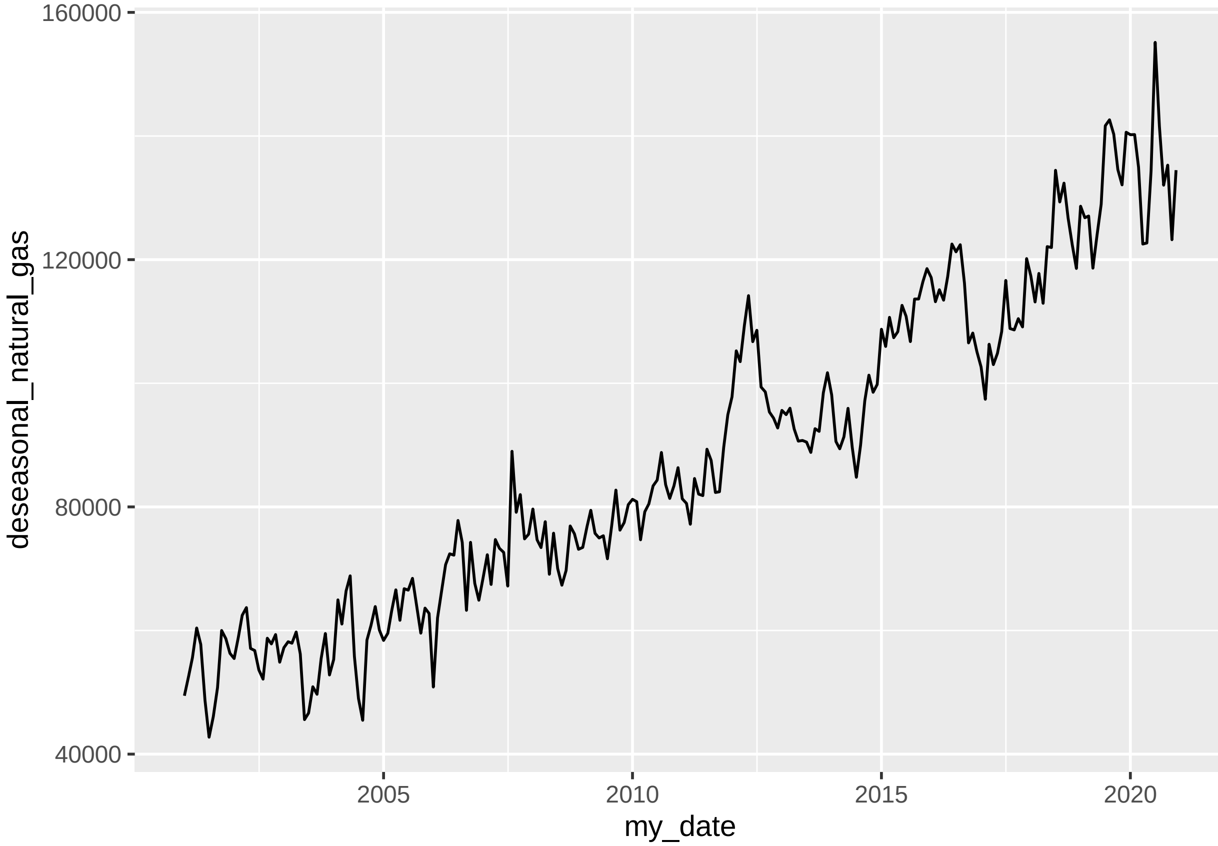
Using the *decompose*() or *stl*() and the *seasadj*() functions create a series without the seasonal component, i.e., a deseasonalized natural gas series. Plot the deseasonalized series over time and corresponding ACF and PACF. Compare with the plots obtained in Q1.

```
#decompose the time series
decompose_natural_gas <- decompose(ts_natural_gas,type="additive")
plot(decompose_natural_gas)
```

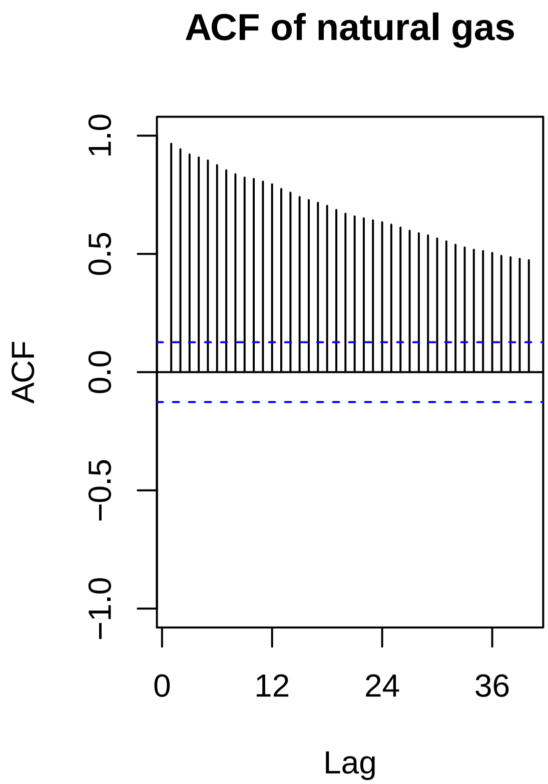


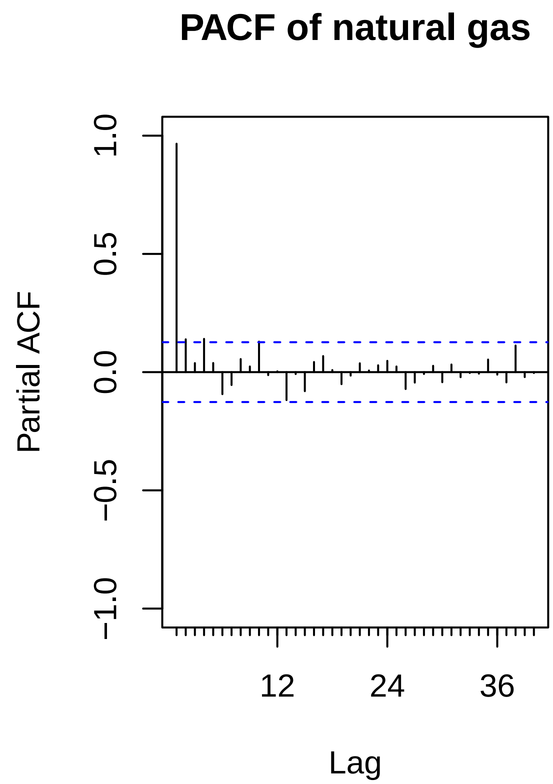

```r
#deseason the time series
deseasonal_natural_gas <- seasadj(decompose_natural_gas)

#plot the deseasoned series and its ACF, PACF
plot(ggplot(deseasonal_natural_gas,aes(x=my_date,y=deseasonal_natural_gas))+
       geom_line()
)
```

```
## Don't know how to automatically pick scale for object of type <ts>. Defaulting
## to continuous.
```




```
par(mfrow=c(1,2))
Acf(deseasonal_natural_gas,lag.max = 40, ylim=c(-1,1),main="ACF of natural gas")
Pacf(deseasonal_natural_gas,lag.max = 40, ylim=c(-1,1),main="PACF of natural gas")
```





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.

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 can read the plots and interpret the test results.

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 $cat()$ function to print.

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?

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.

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.

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?

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?