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

Assignment 7 - Due date 03/07/24

Yilun Zhu

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

```
#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 core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v stringr 1.5.1
## v forcats 1.0.0     v tibble  3.2.1
## v purrr     1.0.2     v tidyr    1.3.1
## v readr     2.1.5
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
library(cowplot)
```

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

```
library(uroot)
```

Importing and processing the data set

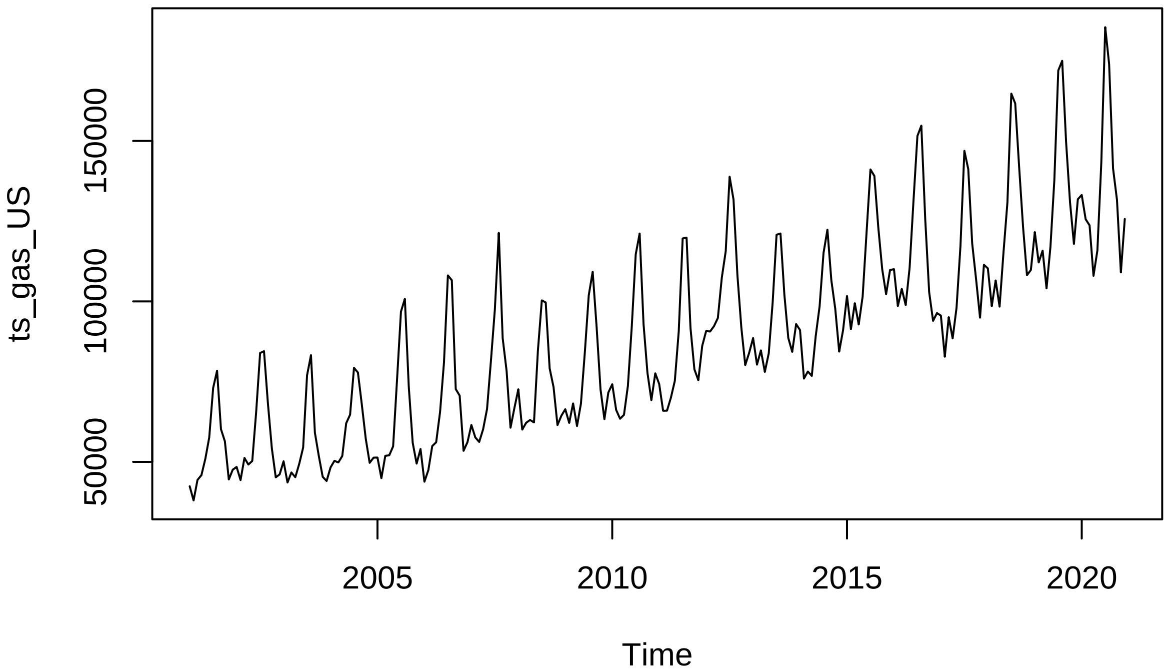
Consider the data from the file "Net_generation_United_States_all_sectors_monthly.csv". The data corresponds to the monthly net generation from January 2001 to December 2020 by source and is provided by the US Energy Information and Administration. **You will work with the natural gas column only**.

Q1

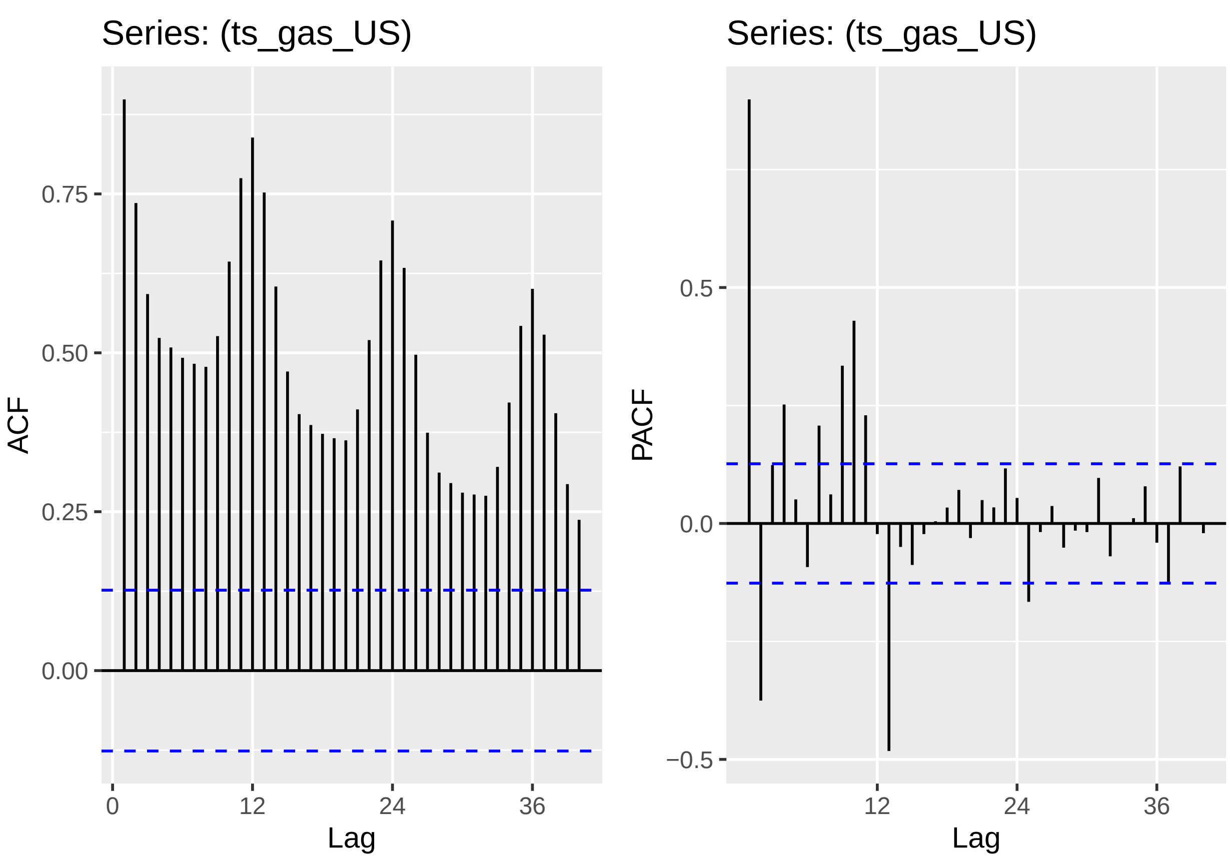
Import the csv file and create a time series object for natural gas. Make you sure you specify the **start=** and **frequency=** arguments. Plot the time series over time, ACF and PACF.

```
Net_Gen_US <- read.csv(file="./Data/Net_generation_United_States_all_sectors_monthly.csv",header=TRUE,s
Net_Gen_US <- Net_Gen_US %>%
  mutate(Month = my(Month)) %>%
  arrange(Month)
ts_gas_US <- ts(Net_Gen_US[,4], start = c(year(Net_Gen_US$Month[1]), month(Net_Gen_US$Month[1])), frequ

plot(ts_gas_US)
```



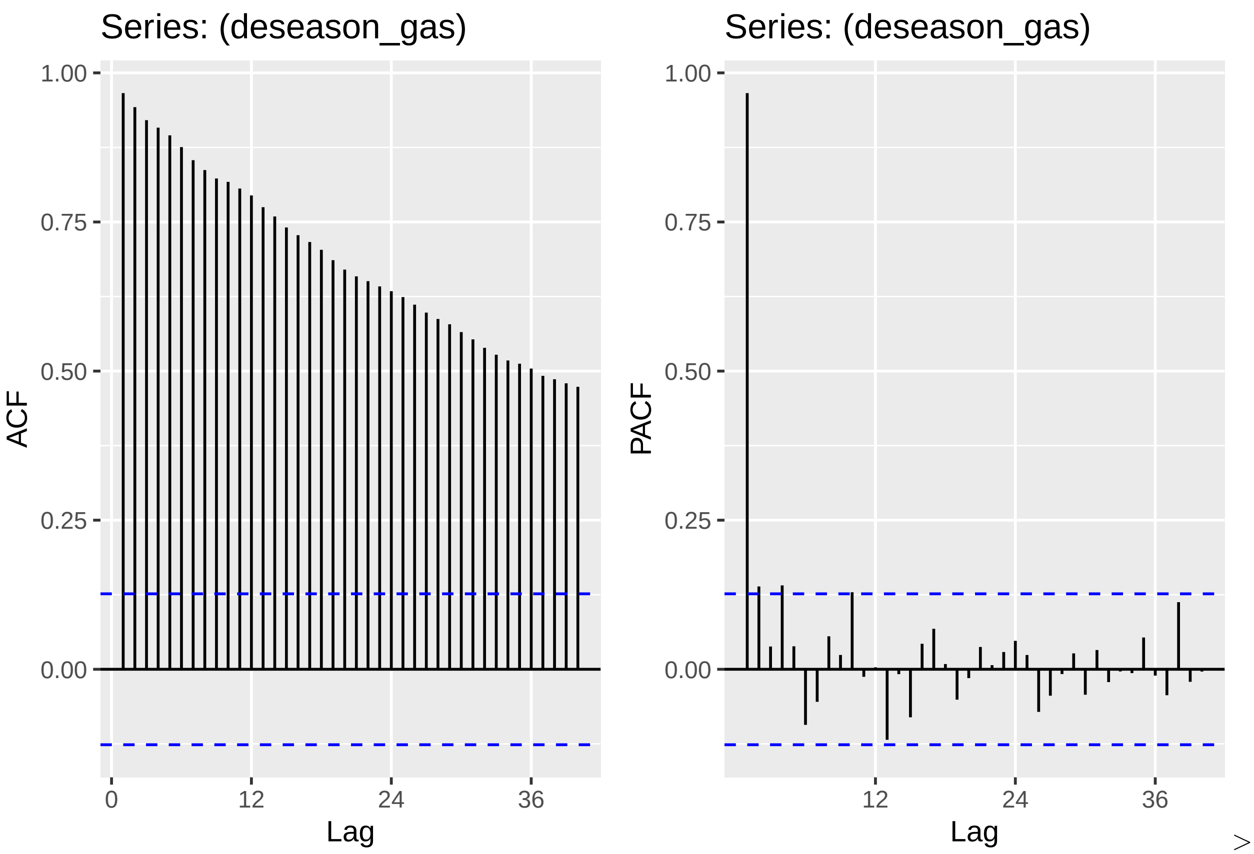

```r
Acfgas <- Acf((ts_gas_US), lag = 40, plot = FALSE)
Pacfgas <- Pacf((ts_gas_US), lag = 40, plot = FALSE)
plot_grid(autoplot(Acfgas),autoplot(Pacfgas), nrow = 1)
```



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_US,"additive")
deseason_gas <- seasadj(decompose_gas)
Acf_desea_gas <- Acf((deseason_gas), lag = 40, plot = FALSE)
Pacf_desea_gas <- Pacf((deseason_gas), lag = 40, plot = FALSE)
plot_grid(autoplot(Acf_desea_gas),autoplot(Pacf_desea_gas), nrow = 1)
```



Answer: Seasonal trend is removed.

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.

```
print("Results for ADF test/n")
```

```
## [1] "Results for ADF test/n"
```

```r
print(adf.test(deseason_gas,alternative = "stationary"))
```

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

```
##
##  Augmented Dickey-Fuller Test
##
## data:  deseason_gas
## Dickey-Fuller = -4.0271, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```r
print("Results for MK test/n")
```

```
## [1] "Results for MK test/n"
```

```r
print(summary(MannKendall(deseason_gas)))
```

```
## Score =  24186 , Var(Score) = 1545533
## denominator =  28680
## tau = 0.843, 2-sided pvalue =< 2.22e-16
## NULL
```

> Answer: The result of ADF test shows a p-value < 0.01 which means we can reject the null hypothesis that the deseasonal gas data has a unit root. And we cannot conclude that it's stationary because the MannKendall Test shows a p-value < 0.05 means there is a deterministic trend, which can also be checked in Acf plot.

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.

```r
#From the previous adf and MK test, I can get the information that it still needs to do differencing.
#Test if the ts_gas needs difference
Dif <- ndiffs(deseason_gas, alpha = 0.05, test = c("kpss", "adf", "pp"), max.d = 2)
print(Dif)
```

```
## [1] 1
```

```r
#It shows deseason_gas needs a difference. So, I did a test to confirm it
ddgas <- diff(deseason_gas,differences=1,lag=1)
print("Results for ddgas ADF test/n")
```

```
## [1] "Results for ddgas ADF test/n"
```

```r
print(adf.test(ddgas,alternative = "stationary"))
```

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

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ddgas
## Dickey-Fuller = -6.9137, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```r
print("Results for ddgas MK test/n")
```

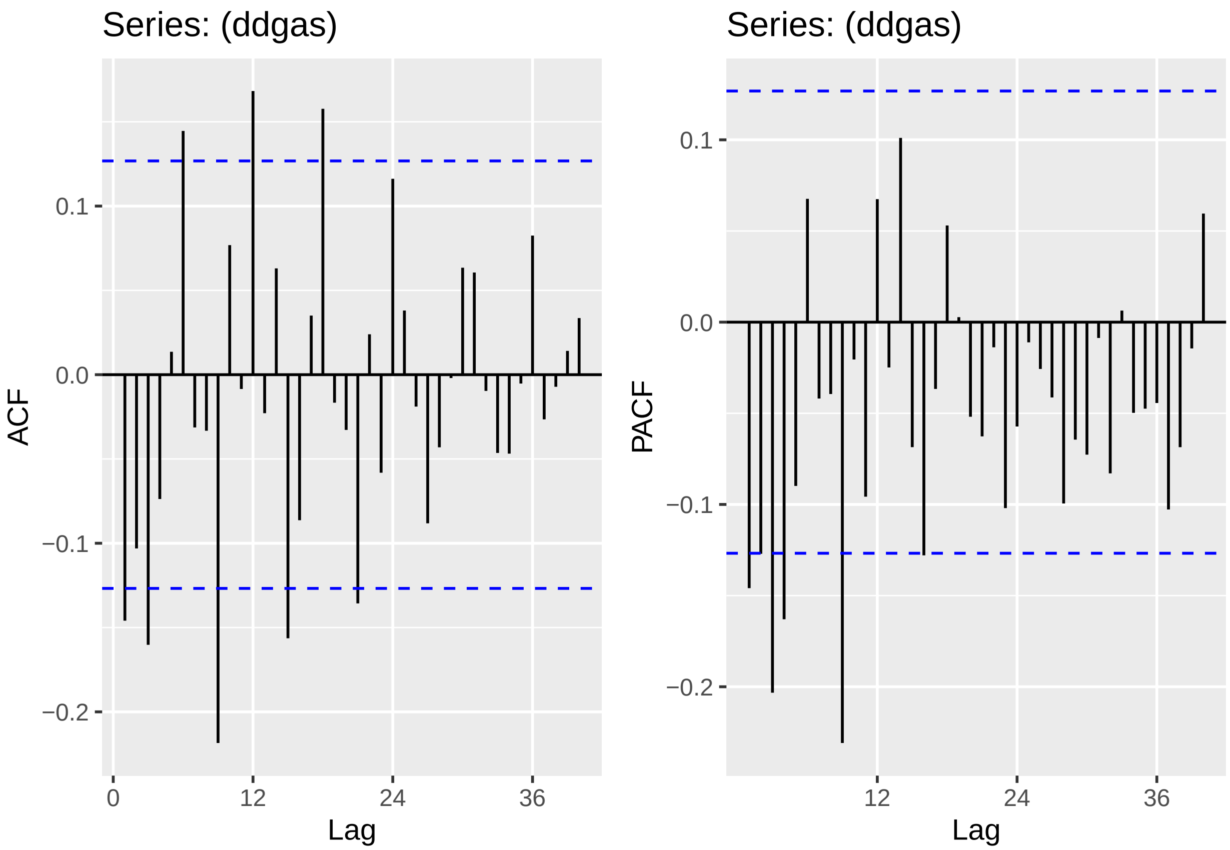
```
## [1] "Results for ddgas MK test/n"
```

```r
print(summary(MannKendall(ddgas)))
```

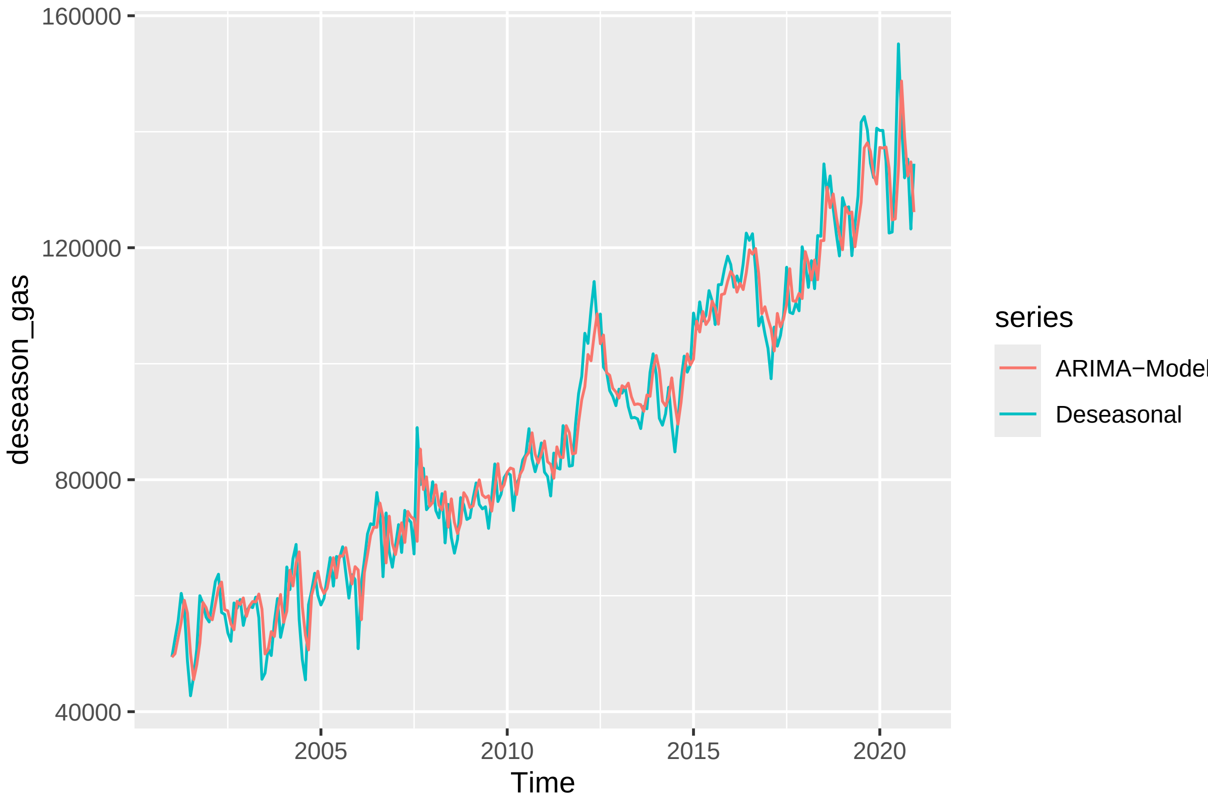
```
## Score =  -299 , Var(Score) = 1526334
## denominator =  28441
## tau = -0.0105, 2-sided pvalue =0.80939
## NULL
```

```r
# The result of two test is agreed with each other. d will be 1 in Arima

Acf_ddgas <- Acf((ddgas), lag = 40, plot = FALSE)
Pacf_ddgas <- Pacf((ddgas), lag = 40, plot = FALSE)
plot_grid(autoplot(Acf_ddgas),autoplot(Pacf_ddgas), nrow = 1)
```




```
arima_gas <- Arima(deseason_gas, order = c(1,1,1),include.drift = TRUE)
autoplot(deseason_gas, series = "Deseasonal")+
  autolayer(arima_gas$fitted, series = "ARIMA-Model")
```




```
arima_gas2 <- Arima(deseason_gas, order = c(2,1,1),include.drift = TRUE)
autoplot(deseason_gas, series = "Deseasonal")+
  autolayer(arima_gas2$fitted, series = "ARIMA-Model2")
```


```
arima_gas3 <- Arima(deseason_gas, order = c(1,1,2),include.drift = TRUE)
autoplot(deseason_gas, series = "Deseasonal")+
  autolayer(arima_gas3$fitted, series = "ARIMA-Model3")
```


```r
arima_gas4 <- Arima(deseason_gas, order = c(2,1,2),include.drift = TRUE)
autoplot(deseason_gas, series = "Deseasonal")+
  autolayer(arima_gas4$fitted, series = "ARIMA-Model4")
```


```r
print(arima_gas)
```

```
## Series: deseason_gas
## ARIMA(1,1,1) with drift
##
## Coefficients:
##          ar1      ma1     drift
##       0.7065  -0.9795  359.5052
## s.e.  0.0633   0.0326   29.5277
##
## sigma^2 = 26980609:  log likelihood = -2383.11
## AIC=4774.21   AICc=4774.38   BIC=4788.12
```

```r
print(arima_gas2)
```

```
## Series: deseason_gas
## ARIMA(2,1,1) with drift
##
## Coefficients:
##          ar1     ar2      ma1     drift
##       0.7057  0.0017  -0.9798  359.4921
## s.e.  0.0710  0.0707   0.0360   29.3046
##
## sigma^2 = 27094287:  log likelihood = -2383.11
## AIC=4776.21   AICc=4776.47   BIC=4793.59
```

```r
print(arima_gas3)
```

```
## Series: deseason_gas
## ARIMA(1,1,2) with drift
##
## Coefficients:
##          ar1      ma1     ma2     drift
##       0.7081  -0.9823  0.0025  359.4980
## s.e.  0.0939   0.1189  0.0982   29.3122
##
## sigma^2 = 27094333:  log likelihood = -2383.11
## AIC=4776.21   AICc=4776.47   BIC=4793.59
```

```r
print(arima_gas4)
```

```
## Series: deseason_gas
## ARIMA(2,1,2) with drift
##
## Coefficients:
##           ar1     ar2      ma1      ma2     drift
##       -0.2336  0.7179  -0.0576  -0.9424  359.3693
## s.e.   0.0542  0.0478   0.0494   0.0484   17.2660
##
## sigma^2 = 26504472:  log likelihood = -2381.07
## AIC=4774.13   AICc=4774.49   BIC=4794.99
```

> Answer: From the adf test and MK test, I get the information that deseason_gas need a difference. After differencing, from the new ACF and Pacf, I observe that Acf tails off and Pacf cuts off which indicate ARMA process.Then I test 4 plots and found that (2,1,2) is the best since the log likeihood is the highest(-2381.07). So finally the best Arima model for deseaon_gas is ARIMA$(2, 1, 2)$.

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.

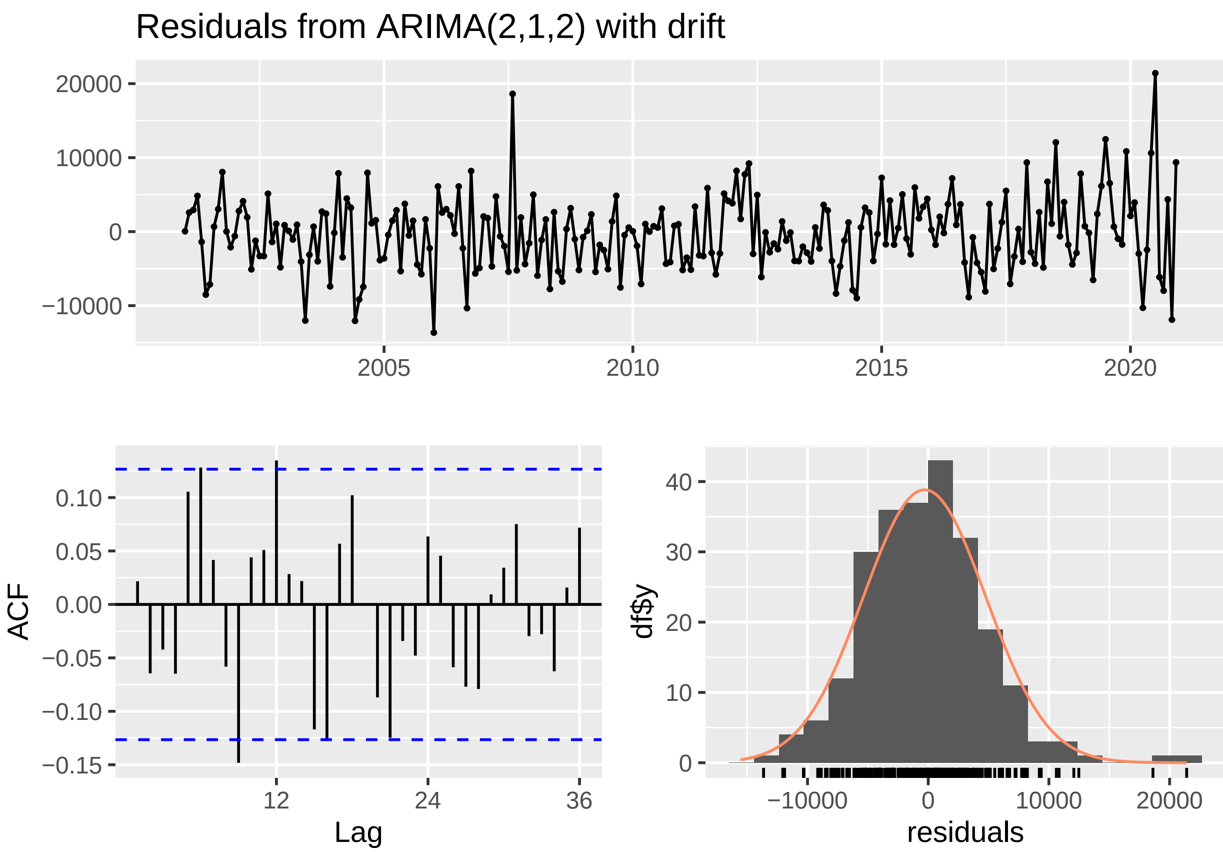
```r
arima_gas <- Arima(deseason_gas, order = c(2,1,2), include.drift = TRUE)
print(arima_gas)
```

```
## Series: deseason_gas
## ARIMA(2,1,2) with drift
##
## Coefficients:
##           ar1     ar2      ma1      ma2     drift
##       -0.2336  0.7179  -0.0576  -0.9424  359.3693
## s.e.   0.0542  0.0478   0.0494   0.0484   17.2660
##
## sigma^2 = 26504472:  log likelihood = -2381.07
## AIC=4774.13   AICc=4774.49   BIC=4794.99
```

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?

```
checkresiduals(arima_gas, lag = 48)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,1,2) with drift
## Q* = 61.317, df = 44, p-value = 0.04302
##
## Model df: 4.   Total lags used: 48
```

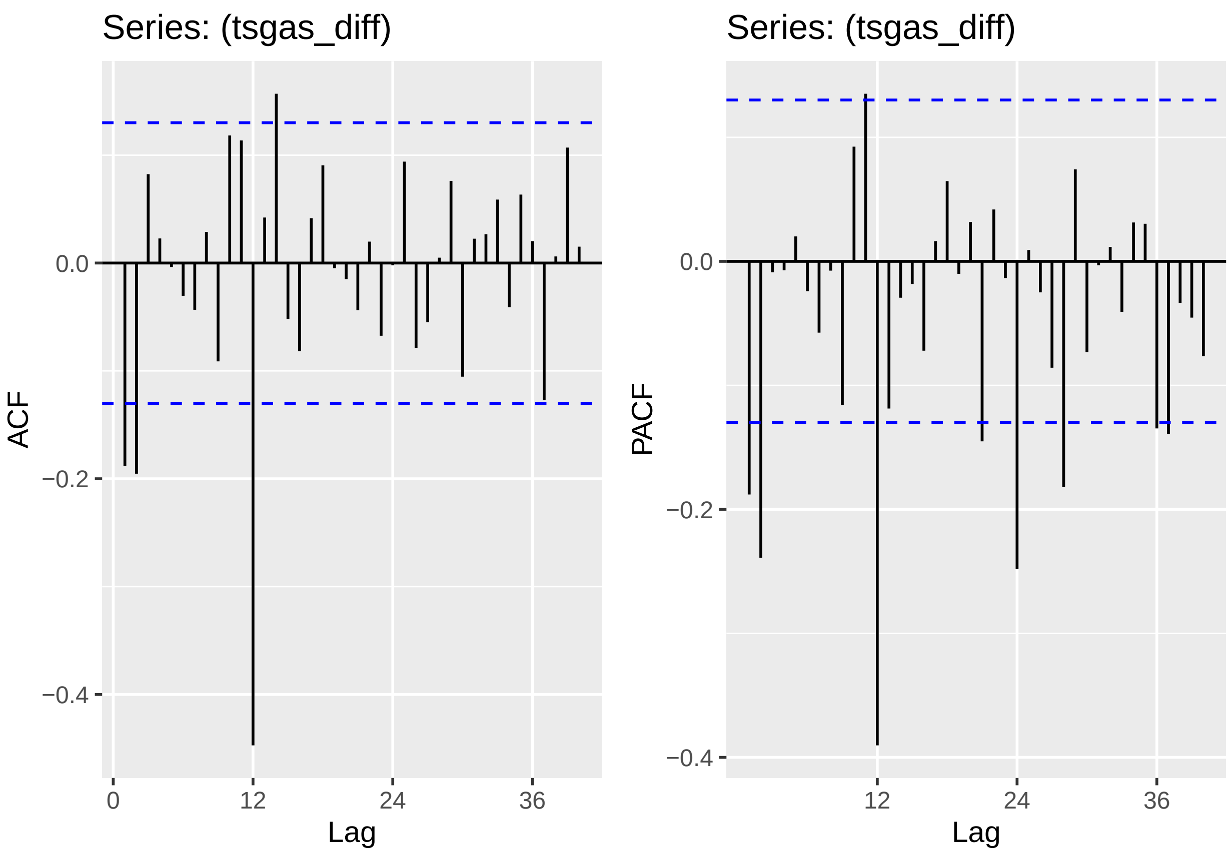
Answer: The outcome looks like a white noise series but has some extreme high points.

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.

```r
gas_trenddiff <- diff(ts_gas_US,differences=1,lag=1)
tsgas_diff <- diff(gas_trenddiff, differences=1, lag = 12)
Acf_dtsgas <- Acf((tsgas_diff), lag = 40, plot = FALSE)
Pacf_dtsgas <- Pacf((tsgas_diff), lag = 40, plot = FALSE)
plot_grid(autoplot(Acf_dtsgas),autoplot(Pacf_dtsgas), nrow = 1)
```




```r
seasonDifch <- nsdiffs(gas_trenddiff, alpha = 0.05, test = "ch", max.D = 1)
seasonDifocsb <- nsdiffs(gas_trenddiff, alpha = 0.05, test = "ocsb", max.D = 1)
print(seasonDifch)
```

```
## [1] 0
```

```r
print(seasonDifocsb)
```

```
## [1] 0
```

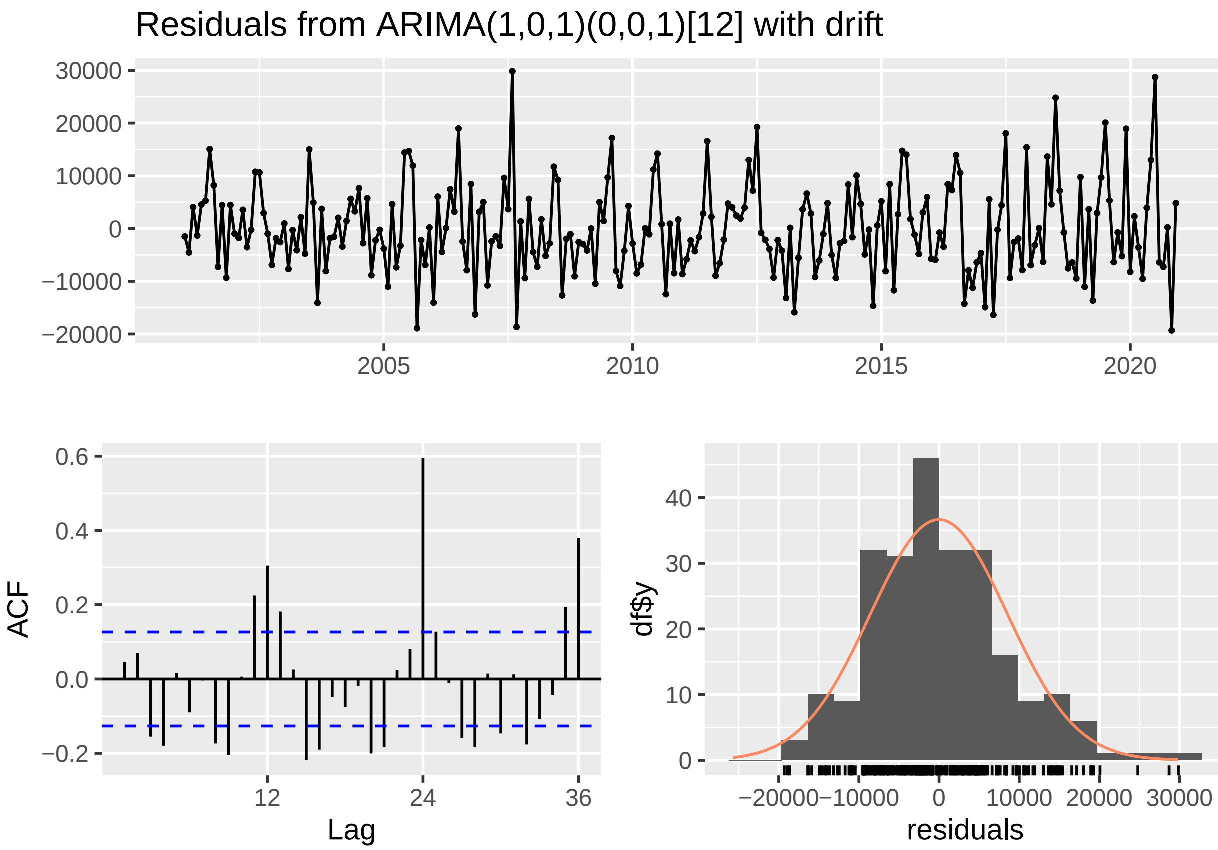
```r
# seems that no need for D

sarima_gas <- Arima(ts_gas_US, order = c(1,0,1), seasonal = c(0,0,1), include.drift = TRUE)
autoplot(ts_gas_US, series = "seasonal")+
  autolayer(sarima_gas$fitted, series = "SARIMA-Model")
```


```r
print(sarima_gas)
```

```
## Series: ts_gas_US
## ARIMA(1,0,1)(0,0,1)[12] with drift
##
## Coefficients:
##          ar1     ma1    sma1   intercept     drift
##       0.5372  0.4302  0.5816   44668.746  360.6864
## s.e.  0.0670  0.0688  0.0433    5097.854   36.2314
##
## sigma^2 = 74601026:  log likelihood = -2516.3
## AIC=5044.59   AICc=5044.95   BIC=5065.48
```

```r
checkresiduals(sarima_gas, lag = 48)
```



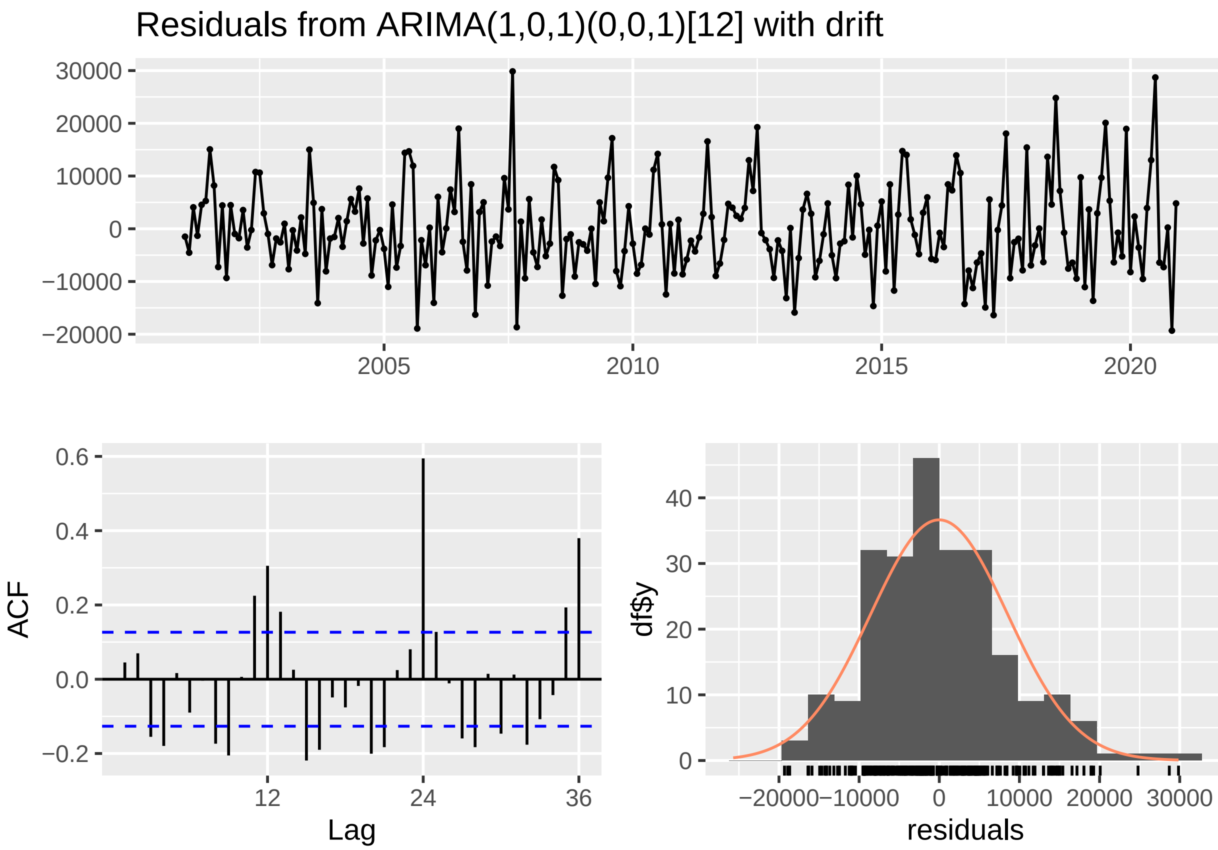

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,1)(0,0,1)[12] with drift
## Q* = 419.53, df = 45, p-value < 2.2e-16
##
## Model df: 3.   Total lags used: 48
```

> Answer:From non-seasonal part, Acf and Pacf all cuts off, so it should be a ARMA. From the Acf and Pacf seasonal part, the Acf has single spikes and Pacf has multiple spikes, so there is more likely a SMA process than SAR process. From the nsdiffs function, it shows there is no need for a seasonal differencing. So model should be $\text{SARIMA}(1,0,1)(0,0,1)_1 2$

Q8

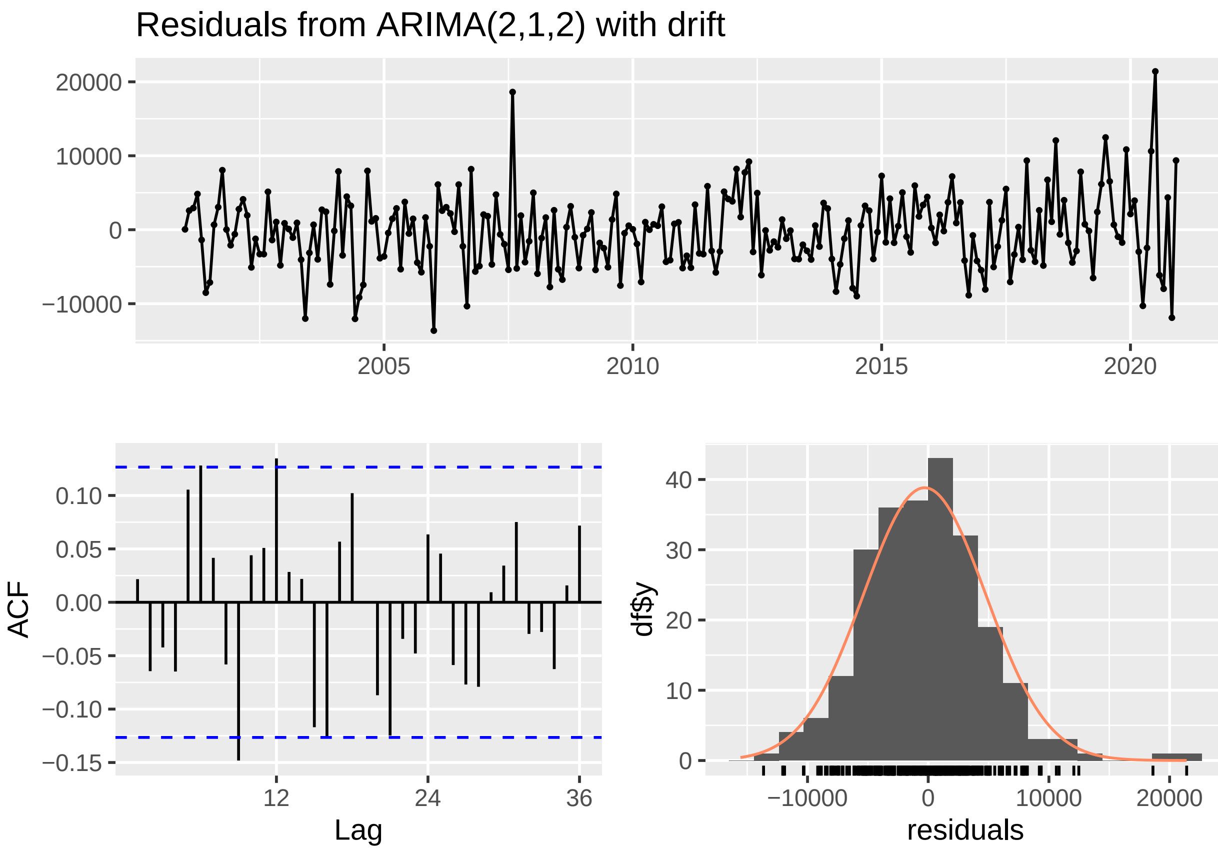
Compare the residual series for Q7 and Q6. Can you tell which ARIMA model is better representing the Natural Gas Series? Is that a fair comparison? Explain your response.

```r
checkresiduals(sarima_gas, lag = 48)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,1)(0,0,1)[12] with drift
## Q* = 419.53, df = 45, p-value < 2.2e-16
##
## Model df: 3.   Total lags used: 48
```

```r
checkresiduals(arima_gas, lag = 48)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,1,2) with drift
## Q* = 61.317, df = 44, p-value = 0.04302
##
## Model df: 4.   Total lags used: 48
```

Answer: According to Acf and residual series, sarima shows more spikes and more large residuals than arima. This comparation seems unfair because the sarima model get a overall combine of seasonality and trend which is more complex.

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?

```
fitarima_gas <- auto.arima(deseason_gas, seasonal = FALSE)
print(fitarima_gas)
```

```
## Series: deseason_gas
## ARIMA(1,1,1) with drift
##
## Coefficients:
##          ar1      ma1     drift
##       0.7065  -0.9795  359.5052
## s.e.  0.0633   0.0326   29.5277
##
## sigma^2 = 26980609:  log likelihood = -2383.11
## AIC=4774.21   AICc=4774.38   BIC=4788.12
```

Answer: The best fit is ARIMA(1,1,1). I wonder the log liki of (2,1,2) is higher than this (1,1,1), why it is not the best model?

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
fitsarima_gas <- auto.arima(ts_gas_US)
print(fitsarima_gas)
```

```
## Series: ts_gas_US
## ARIMA(1,0,0)(0,1,1)[12] with drift
##
## Coefficients:
##          ar1     sma1     drift
##       0.7416  -0.7026  358.7988
## s.e.  0.0442   0.0557   37.5875
##
## sigma^2 = 27569124:  log likelihood = -2279.54
## AIC=4567.08   AICc=4567.26   BIC=4580.8
```

Answer: The best fit is ARIMA(1,0,0),(0,1,1). Far away from my model.