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

Assignment 7 - Due date 03/07/24

Student Name

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

```r
#Load/install required package here
library(forecast)
library(tseries)
library(sarima)
library(ggplot2)
library(cowplot)
library(tidyverse)
library(Kendall)
```

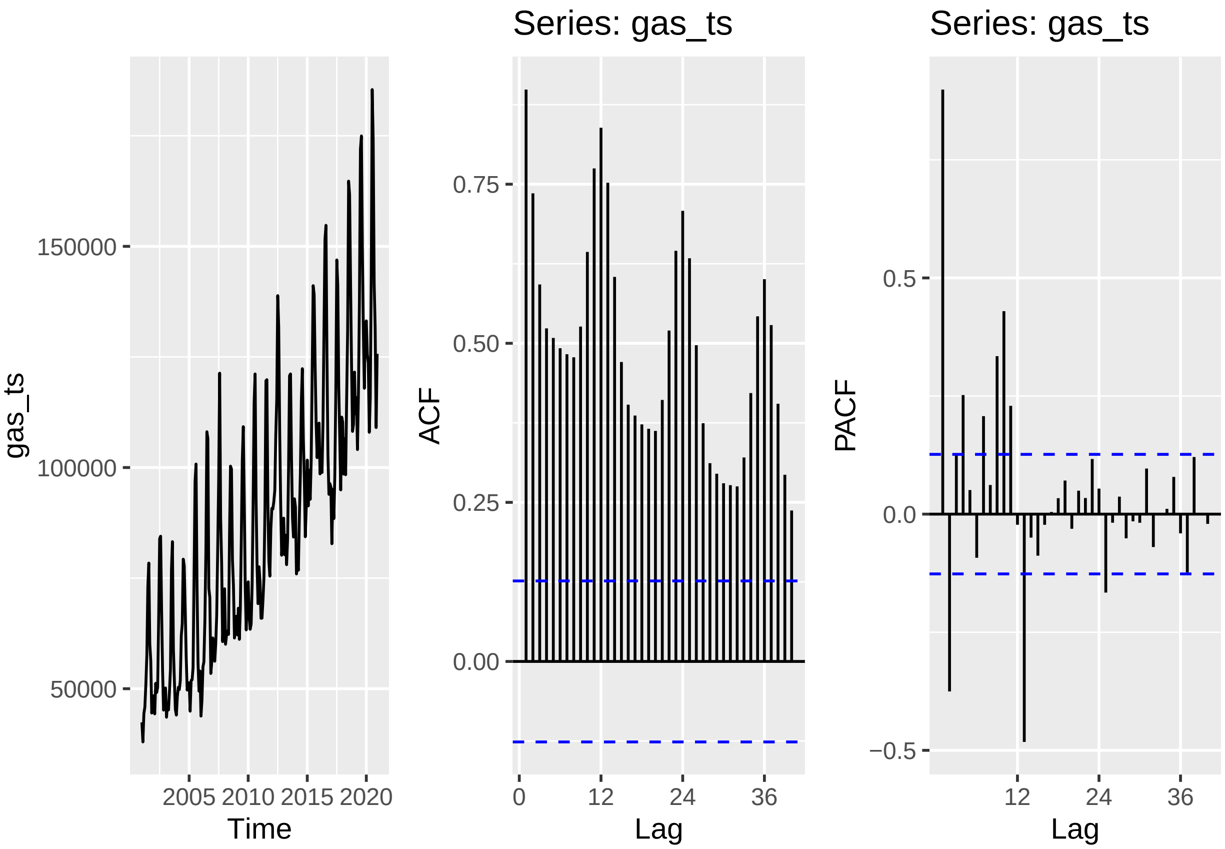
Importing and processing the data set

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.

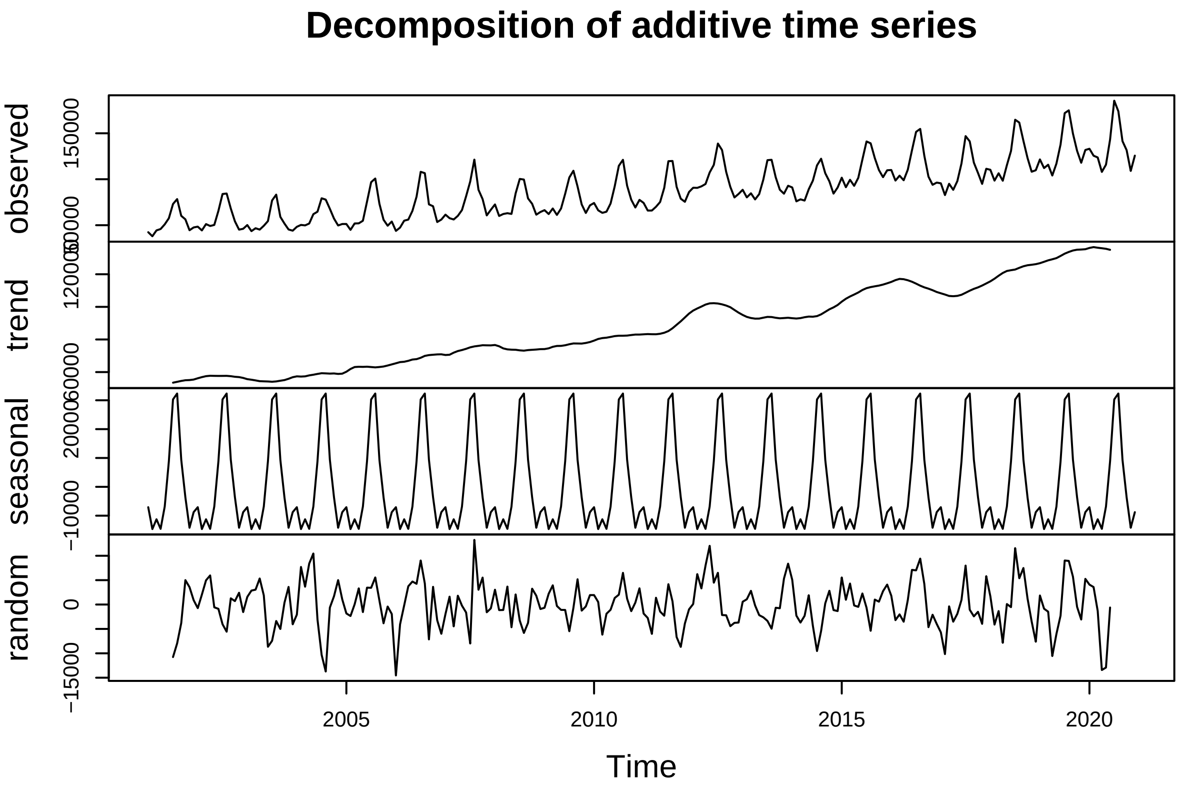
```r
data <- read.csv(
  file="./Data/Net_generation_United_States_all_sectors_monthly.csv",
  header=TRUE,
  skip=4)

gas_data <- data %>%
  mutate(Date = my(Month)) %>%
  arrange(Date) %>%
  mutate(Nat.gas = natural.gas.thousand.megawatthours) %>%
  select(Date,Nat.gas)

fmonth <- month(first(gas_data$Date))
fyear <- year(first(gas_data$Date))

seqDate <- seq.Date(first(gas_data$Date),last(gas_data$Date),by="month")
#240 rows as gas_data so no NAs

gas_ts <- ts(gas_data$Nat.gas,
             start=c(fyear,fmonth),
             frequency=12)

plot_grid(
  autoplot(gas_ts),
  autoplot(Acf(gas_ts,plot=FALSE,lag.max=40)),
  autoplot(Pacf(gas_ts,plot=FALSE,lag.max=40)),
  nrow=1
)
```



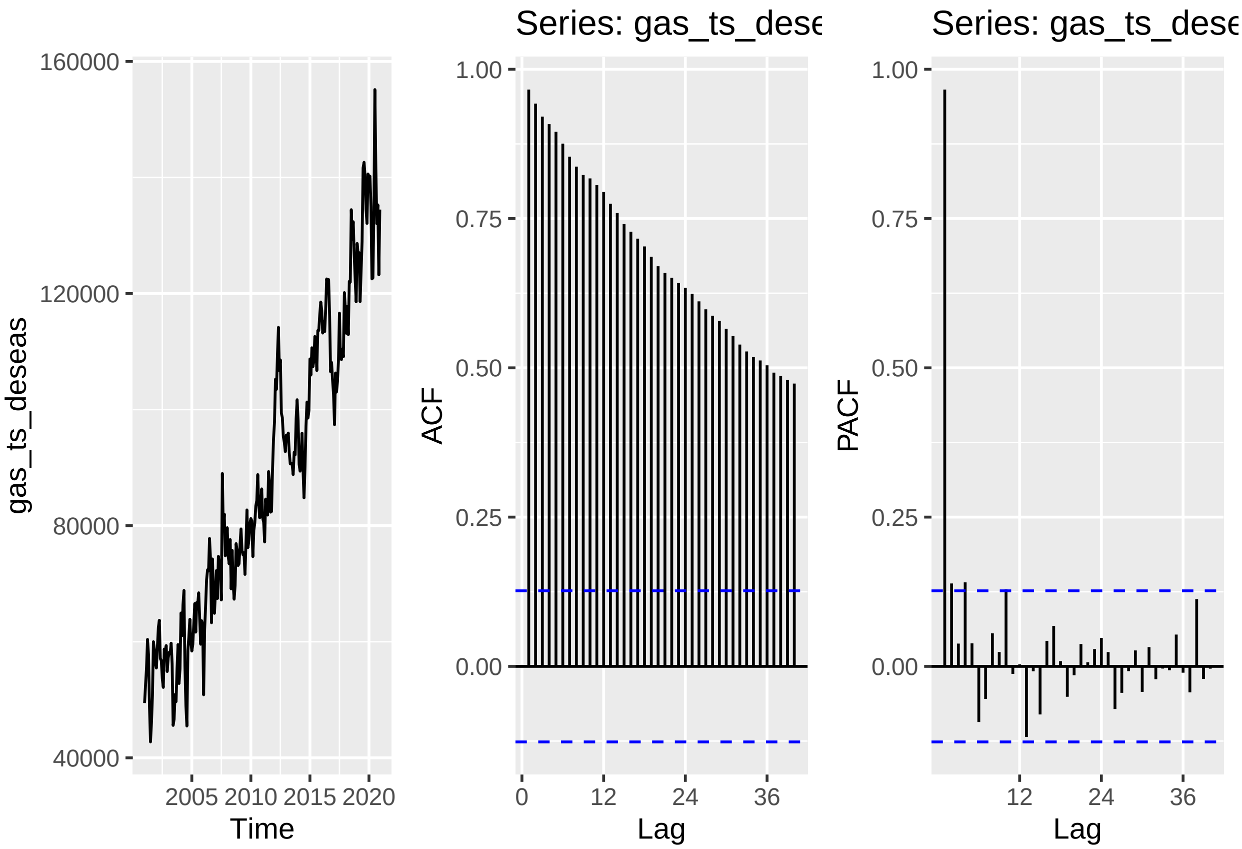
Q2

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.

```r
gas_decomp <- decompose(gas_ts)
plot(gas_decomp)
```



```
gas_ts_deseas <- seasadj(gas_decomp)

plot_grid(
  autoplot(gas_ts_deseas),
  autoplot(Acf(gas_ts_deseas,plot=FALSE,lag.max=40)),
  autoplot(Pacf(gas_ts_deseas,plot=FALSE,lag.max=40)),
  nrow=1
)
```



Answer: Seasonal component seems to have been eliminated with the seasonal adjustment. Series still exibits a trend from the ACF plot since we have a slow decay of the autocorrelation coeficients.

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.

```
#ADF first
print(adf.test(gas_ts_deseas))
```

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

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

```
#Mann Kendall
summary(MannKendall(gas_ts_deseas))
```

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

> Answer: ADF test shows a p-value of 0.01. Therefore we reject the null hypothesis, i.e., the deseason series does not have a stochastic trend. Mann Kendall test p-value indicates we should reject the null hypothesis, in other words, data has a deterministic trend.

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: From Q2 plots and Q3 test results, the deseason series is an AR model of order 1 (p =1 and q = 0) and there is no need to differentiate the serie (d=0). Therefore ARIMA(1,0,0).

> Answer: However if you use the ndiffs function you would have specified d = 1 and then after differencing you would need to look at ACF/PACF plots again and revisit your initial assumption for p and q.

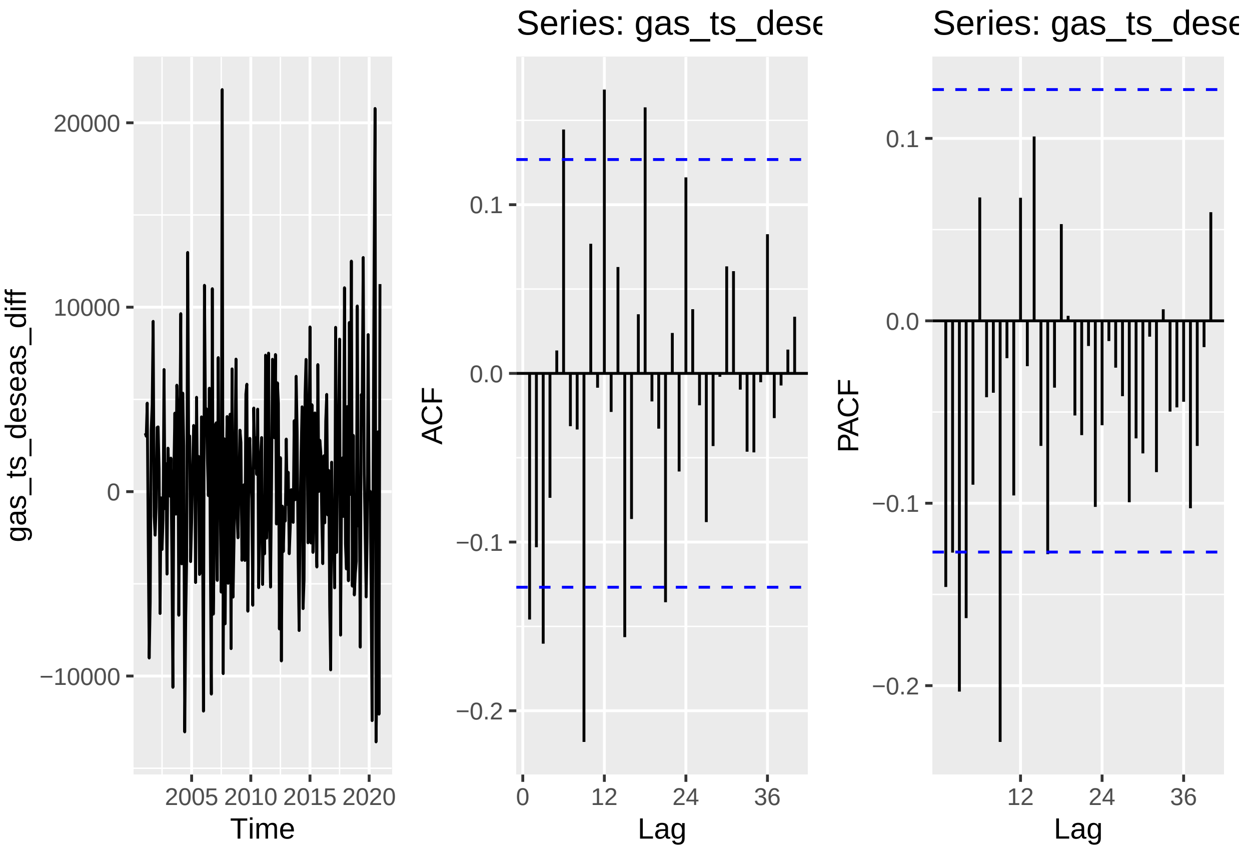
```r
print("Number of lag 1 differencing needed:")
```

```
## [1] "Number of lag 1 differencing needed:"
```

```r
print(ndiffs(gas_ts_deseas))
```

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

```r
gas_ts_deseas_diff <- diff(gas_ts_deseas, lag = 1, differences = 1)

plot_grid(
  autoplot(gas_ts_deseas_diff),
  autoplot(Acf(gas_ts_deseas_diff,plot=FALSE,lag.max=40)),
  autoplot(Pacf(gas_ts_deseas_diff,plot=FALSE,lag.max=40)),
  nrow=1
)
```



> Answer: After differencing the trend component is gone meaning we do not need an AR component to model this series, i.e., p = 0. And it also does not seem to need a moving average term since we can't see slow decay on PACF, i.e., q = 0. Remember that whenever we can't see AR or MA characteristics on ACF and PACF it could be that we have both going on meaning we have an ARMA model. Thefore ARIMA(1,1,1) is also a good candidate for the deseasonal series.

> Answer: Any specification will be accepted for full credit here. Just make sure you justify your choices!

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.

```
#Fitting the Arima(1,0,0)
Model_AR1 <- Arima(gas_ts_deseas,
                      order=c(1,0,0),
                      include.constant = TRUE)
summary(Model_AR1)
```

```
## Series: gas_ts_deseas
## ARIMA(1,0,0) with non-zero mean
##
## Coefficients:
##          ar1      mean
##       0.9825  90230.35
## s.e.  0.0120  16957.97
## 
## sigma^2 = 30851494:  log likelihood = -2410.59
## AIC=4827.17   AICc=4827.27   BIC=4837.61
## 
## Training set error measures:
##                    ME    RMSE      MAE         MPE     MAPE      MASE
## Training set 281.0308 5531.22 4289.962 -0.09287026 5.240518 0.5244823
##                    ACF1
## Training set -0.1387296
```

```r
print(Model_AR1$coef)
```

```
##          ar1    intercept 
## 9.825447e-01 9.023035e+04
```

```r
#Fitting the Arima(1,1,1)
Model_ARIMA111 <- Arima(gas_ts_deseas,
                      order=c(1,1,1),
                      include.constant = TRUE)
summary(Model_ARIMA111)
```

```
## Series: gas_ts_deseas 
## 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
## 
## Training set error measures:
##                     ME     RMSE     MAE        MPE     MAPE      MASE
## Training set -141.3123 5150.819 3984.38 -0.7171368 4.850437 0.4871225
##                      ACF1
## Training set -0.003014461
```

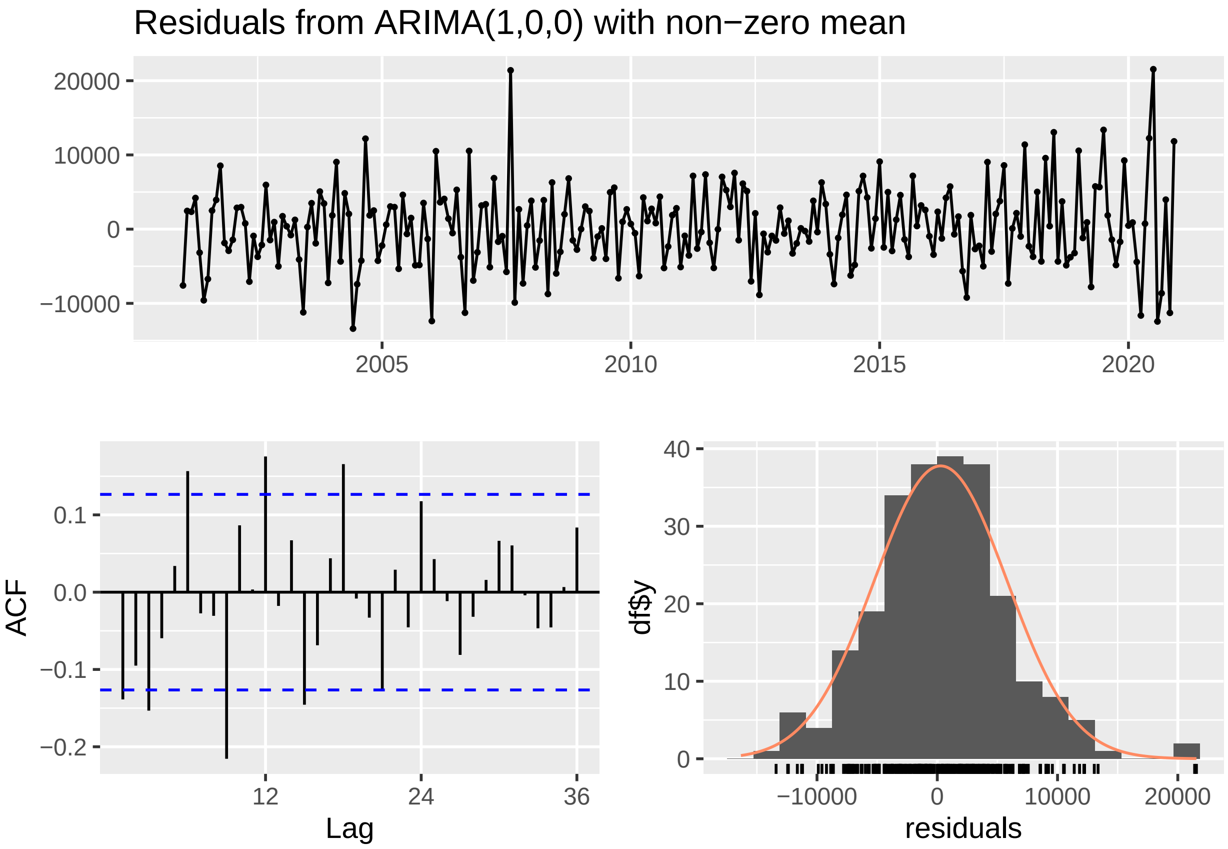
```r
print(Model_ARIMA111$coef)
```

```
##         ar1         ma1       drift 
##   0.7065237  -0.9794655 359.5051902
```

Q6

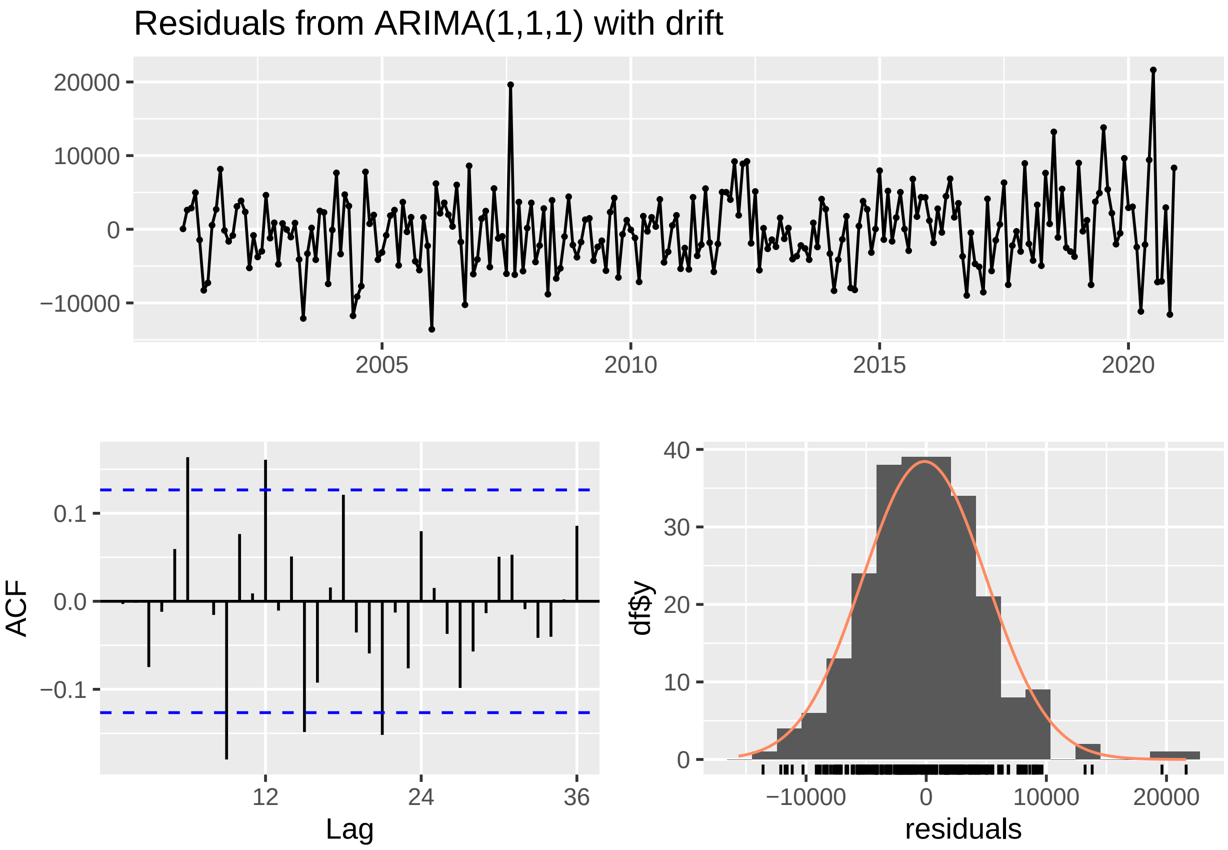
Now plot the residuals of the ARIMA fit from Q5 along with residuals ACF and PACF on the same window. You may use the *checkresiduals*() function to automatically generate the three plots. Do the residual series look like a white noise series? Why?

```r
checkresiduals(Model_AR1)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0) with non-zero mean
## Q* = 66.317, df = 23, p-value = 4.443e-06
##
## Model df: 1.   Total lags used: 24
```

```r
checkresiduals(Model_ARIMA111)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,1) with drift
## Q* = 48.356, df = 22, p-value = 0.0009736
##
## Model df: 2.   Total lags used: 24
```

Answer: For both Arima fits, the residuals seems to be following a normal distribution with no significant temporal correlation, so yes, a white noise series. Both models lead to a few ACF values outside the blue dashed lines. The AIC results also indicate lower AIC for ARIMA(1,1,1). The Ljung–Box test is a type of statistical test of whether any of a group of autocorrelations of a time series are different from zero. Null hypothesis state that data are independently distributed and alternative is data is NOT independently distributed. Note from test results for both models we reject the null hypothesis and conclude there IS temporal correlation. In other words the ARIMA fits were not able to fully model the temporal correlation of the series.

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: From Q1 plots, we have a strong trend component and seasonal component. Let's start with seasonal part by checking if we need seasonal differencing with nsdiffs().

```r
print("Number of seasonal differencing needed:")
```

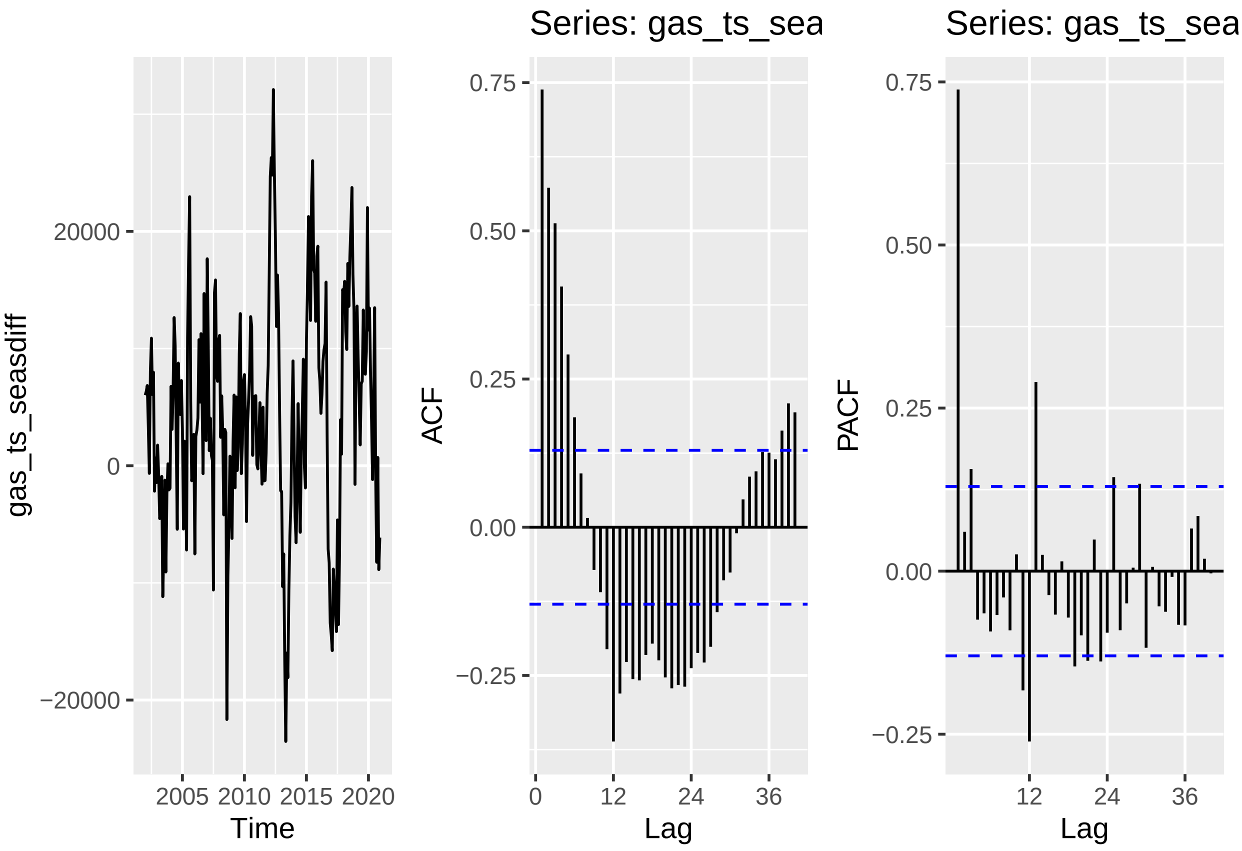
```
## [1] "Number of seasonal differencing needed:"
```

```r
print(nsdiffs(gas_ts))
```

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

> Answer: We need one differencia at the seasonal lag, so let's use the diff() function and plot ACF and PACF for differenced series.

```r
gas_ts_seasdiff <- diff(gas_ts, lag = 12, differences = 1)

plot_grid(
  autoplot(gas_ts_seasdiff),
  autoplot(Acf(gas_ts_seasdiff,plot=FALSE,lag.max=40)),
  autoplot(Pacf(gas_ts_seasdiff,plot=FALSE,lag.max=40)),
  nrow=1
)
```



> Answer: From the new ACF/PACF plots wave pattern is gone, but we still have a negative significant seasonal lags in ACF 12 and PACF 12 meaning we might still need a MA seasonal term. Remember seasonal term is either AR or MA, never both! Therefore P = 0, Q = 1 and D = 1.

Answer: Now let's focus on initial nonseasonal part - lags 1 through 12. The ACF has a exponential decay meaning we do not need any differencing and an AR term is needed. From PACF order of AR component should be 1 because we a have a cut off from lag 1 to 2. Note that lag 2 is already falling withing the blue dashed lines. It seems like we do not need a moving average component, therefore p = 1, q = 0 and d = 0.

Answer: You could also run the ndiffs() on the seasonal differenced series to check if additional differencing is needed.

```r
print("Number of lag 1 differencing needed after seasonal differencing:")
```

```
## [1] "Number of lag 1 differencing needed after seasonal differencing:"
```

```r
print(nsdiffs(gas_ts_seasdiff))
```

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

Answer: The ndiffs() is reinforcing our obserbation from ACF that no additional differencing is needed. Now we need to fit the SARIMA model we identifued and look at residuals.

```r
#Fitting the Arima(1,1,1)
Model_SARIMA <- Arima(gas_ts,
                      order=c(1,0,0),
                      seasonal=c(0,1,1),
                      include.constant = TRUE)
summary(Model_SARIMA)
```

```
## Series: gas_ts
## 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
##
## Training set error measures:
##                      ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -97.32578 5083.901 3950.295 -0.7114711 4.673706 0.4829553
##                     ACF1
## Training set -0.04171074
```

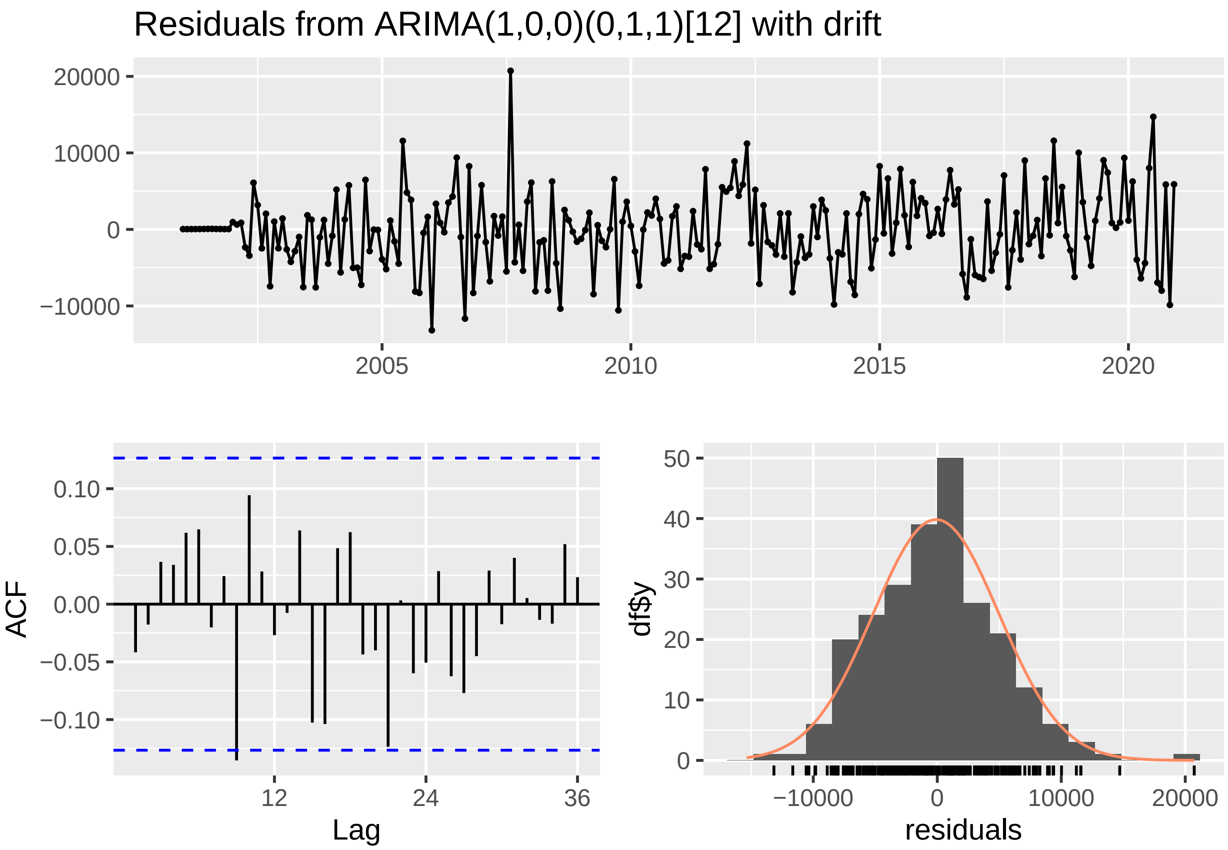
```r
print(Model_SARIMA$coef)
```

```
##         ar1        sma1       drift
##   0.7415607  -0.7025578 358.7988398
```

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.

```
checkresiduals(Model_SARIMA)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0)(0,1,1)[12] with drift
## Q* = 25.414, df = 22, p-value = 0.2777
##
## Model df: 2.   Total lags used: 24
```

> Answer: From the ACF plot, the SARIMA model seems to be doing a better job modeling the temporal dependency of the series. Note that Ljung test now concludes that the residual are independently distribute, i.e, no temporal correlation. But remember from the class this is not a fair comparison because SARIMA is modeling the seasonal component rather than assuming it is constant and removing from the series. So SARIMA results tend to be better than ARIMA.

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_deseas_auto <- auto.arima(gas_ts_deseas,seasonal=FALSE)
summary(Model_deseas_auto)
```

```
## Series: gas_ts_deseas
## 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
##
## Training set error measures:
##                      ME     RMSE     MAE        MPE     MAPE      MASE
## Training set -141.3123 5150.819 3984.38 -0.7171368 4.850437 0.4871225
##                      ACF1
## Training set -0.003014461
```

Answer: Note that the auto.arima() fitted an ARIMA(1,1,1) to the data set. It does match one of our model specifications from Q4. All students will get ful credit even if the auto.arima() does not match your Q4 specification.

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?

```
Model_original_auto <- auto.arima(gas_ts)
summary(Model_original_auto)
```

```
## Series: gas_ts
## 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
##
## Training set error measures:
##                      ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -97.32578 5083.901 3950.295 -0.7114711 4.673706 0.4829553
##                     ACF1
## Training set -0.04171074
```

Answer: Note this is the exactly the same model we identified in Q7. All students will get ful credit even if the auto.arima() does not match your Q7 specification.