ENV 797 - Time Series Analysis for Energy and Environment Applications | Spring 2025

Assignment 7 - Due date 03/06/25

Justin Maynard

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_Sp25.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

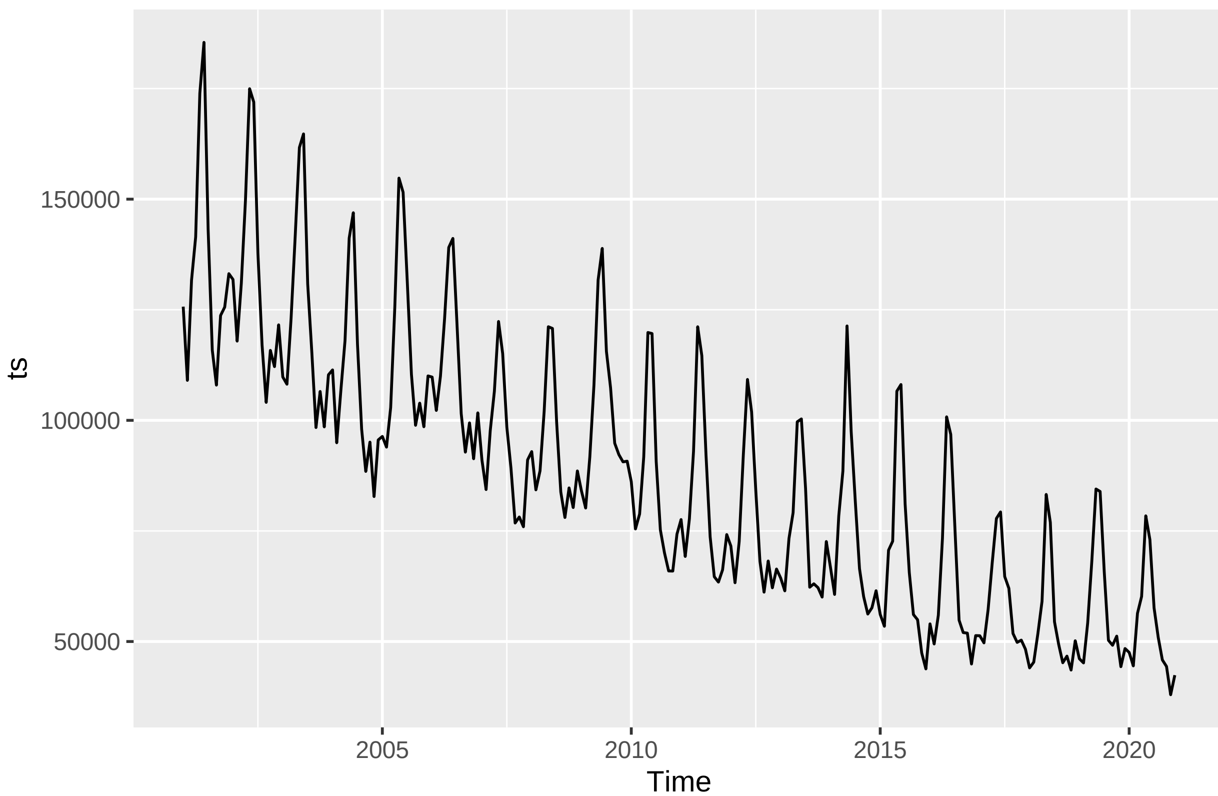
Importing and processing the data set

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**.

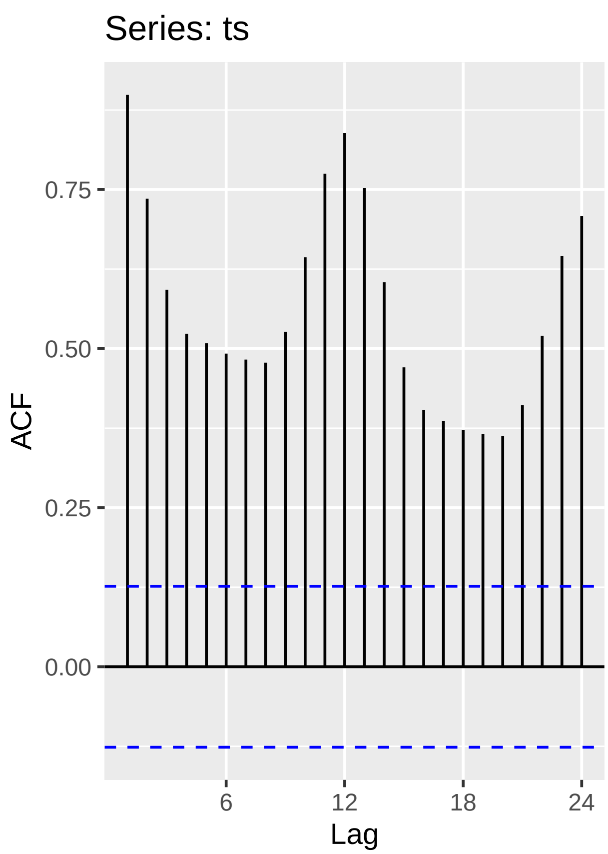
Q1

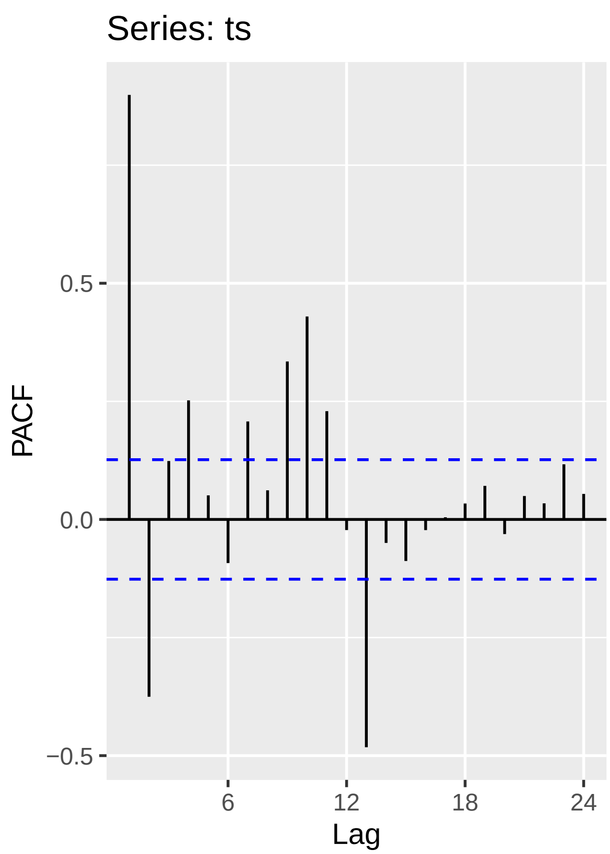
Import the csv file and create a time series object for natural gas. Make you sure you specify the **start=** and **frequency=** arguments. Plot the time series over time, ACF and PACF.

```
autoplot(ts)
```




```
plot_grid(q1_acf, q1_pacf)
```



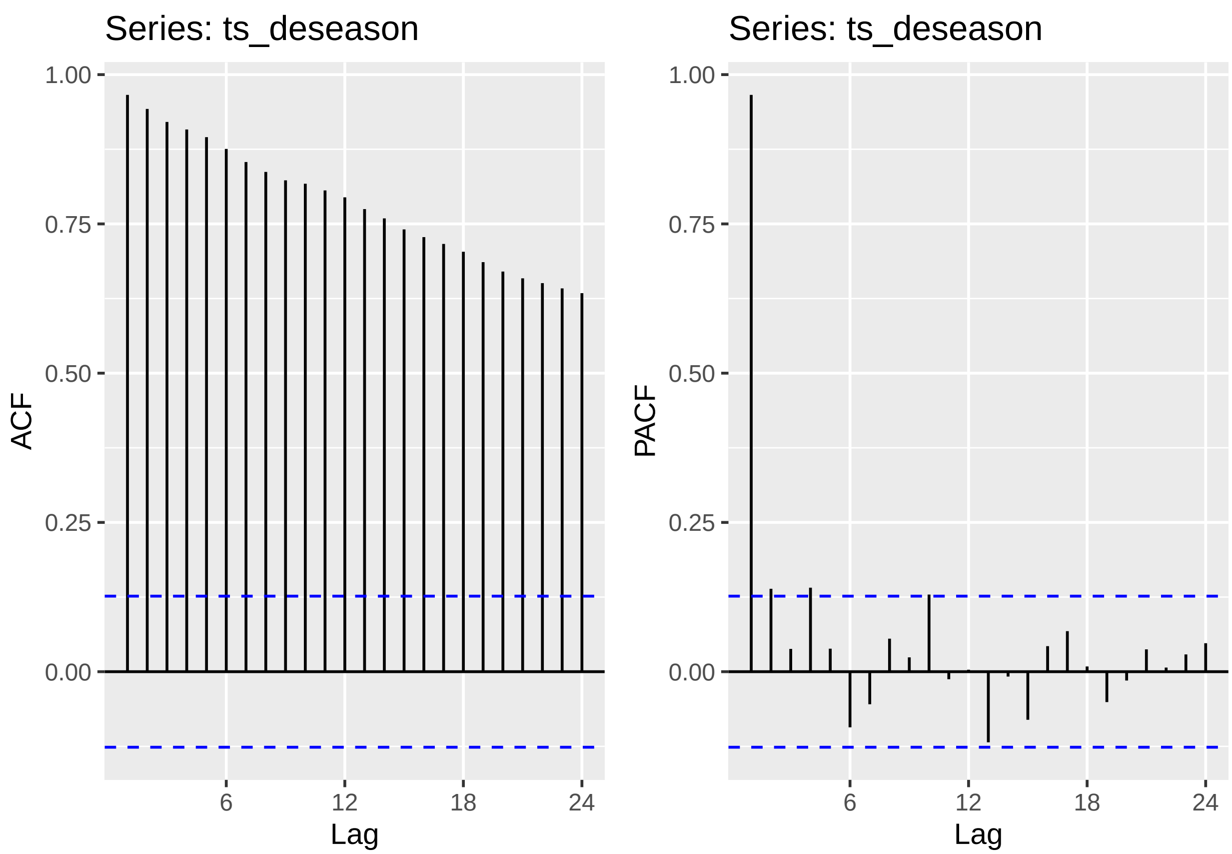


Q2

Using the *decompose*() 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.

```
autoplot(ts_deseason)
```


```r
plot_grid(q2_acf, q2_pacf)
```



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.

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

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

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

```r
print(summary(MannKendall(ts_deseason)))
```

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

The P value of the ADF test (p value < .05) tell us that we reject the null hypothesis, indicating a unit root is not present. The p value from the MK test is < .05, meaning we reject the null hypothesis, and the time series follows a 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.

p: 1 (the PACF cutoff is 1 in AR model) d: 1 (deterministic trend present from MK test) q: 0 (no cutoff for MA)

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()` or `print()` function to print.

```
arima_deseason <- Arima(ts_deseason, c(1,0,0), include.mean = TRUE, include.drift = TRUE)
arima_deseason$coef
```

```
##           ar1     intercept         drift
##  7.181903e-01  1.314198e+05 -3.594317e+02
```

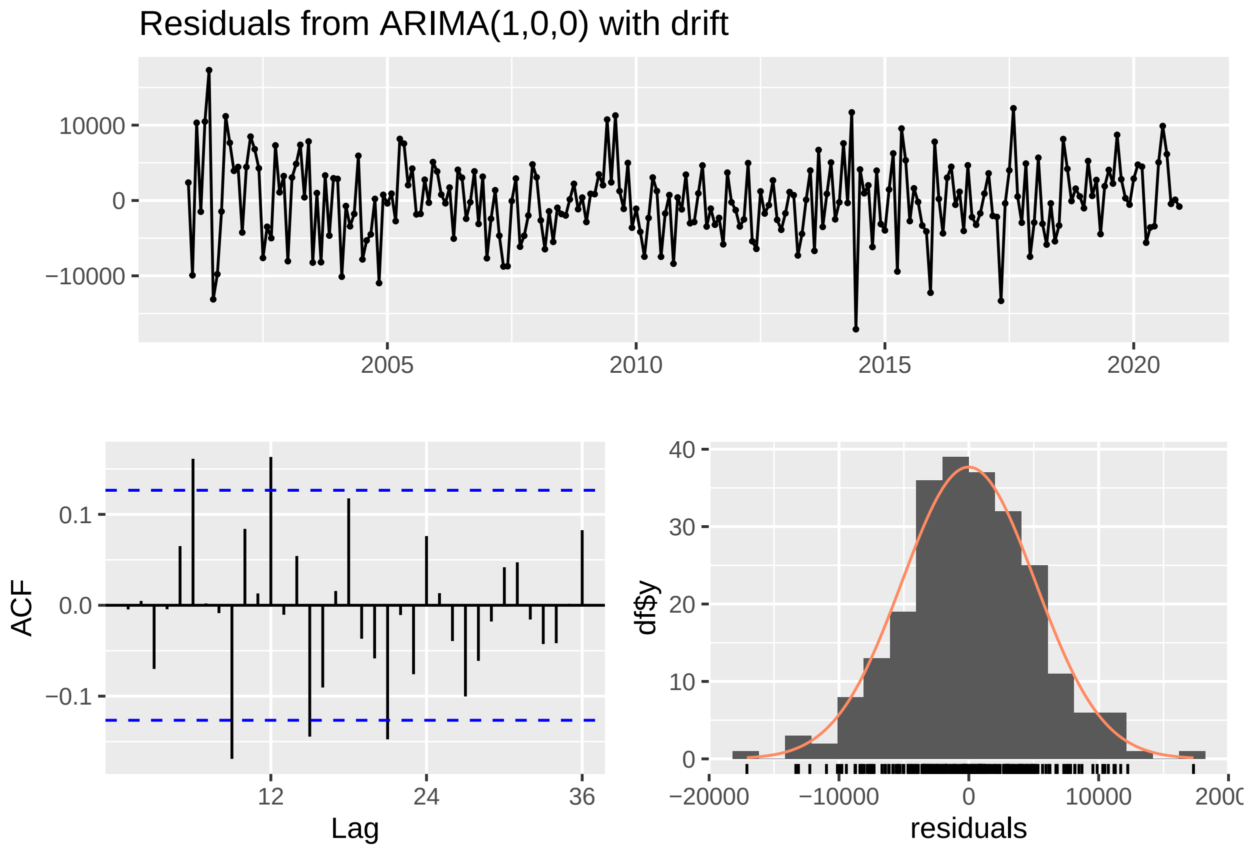
```
print(cat("Autoregressive:", arima_deseason$coef[1], ", Intercept:", arima_deseason$coef[2], ", Drift:"
```

```
## Autoregressive: 0.7181903 , Intercept: 131419.8 , Drift: -359.4317NULL
```

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_deseason)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0) with drift
## Q* = 46.658, df = 23, p-value = 0.002475
##
## Model df: 1.   Total lags used: 24
```

The residuals do look like a white noise series as they do not seem to follow any patterns. The residuals are also normally distributed.

Modeling the original series (with seasonality)

Q7

Repeat Q3-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.

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

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

```
##
##  Augmented Dickey-Fuller Test
##
## data:  ts
## Dickey-Fuller = -8.8795, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```
print(summary(MannKendall(ts)))
```

```
## Score =  -18658 , Var(Score) = 1545533
## denominator =  28680
## tau = -0.651, 2-sided pvalue =< 2.22e-16
## NULL
```

The P value of the ADF test (p value $< .05$) tell us that we reject the null hypothesis, indicating a unit root is not present. The p value from the MK test is $< .05$, meaning we reject the null hypothesis, and the time series follows a trend. The results are the same for the decomposoed and deseasoned series.

p: 1 (the PACF cutoff exists) d: 1 (deterministic trend present from MK test) q: 0 (no cutoff for MA)

P: 1 (Autocorrelation at seasonal periods is positive, multiple spikes at seasonal lag in ACF represents SAR process) D: 1 (series has positive autocorrelations out to a high number of lags) Q: 1 (Multiple spikes at seasonal lag in PACF represents SMA process)

```
#plot_grid(q1_acf, q1_pacf)
```

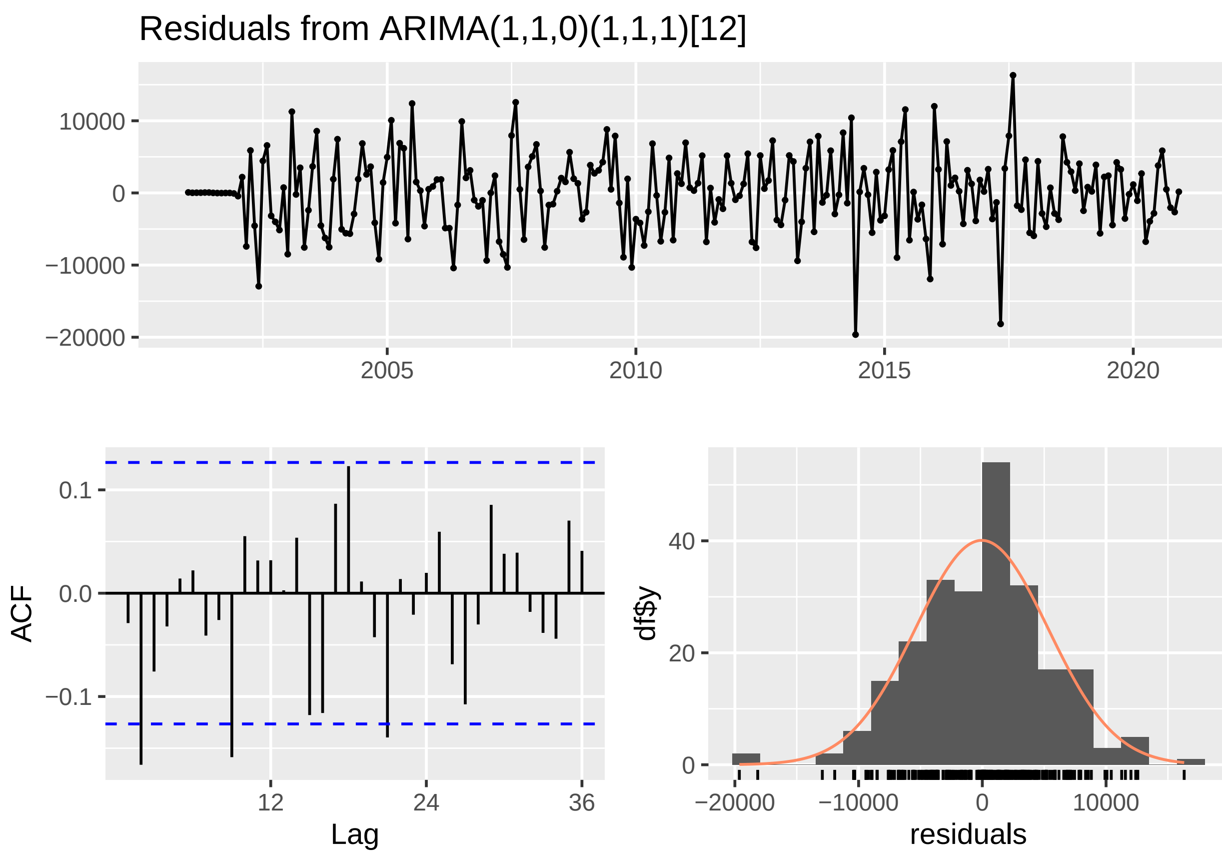
```
arima_ts <- Arima(ts, order = c(1,1,0), seasonal = c(1,1,1), include.mean = TRUE)
arima_ts
```

```
## Series: ts
## 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
```

```
print(cat("AR1:", arima_ts$coef[1], ", SAR1:", arima_ts$coef[2], ", SMA1:",arima_ts$coef[3]))
```

```
## AR1: -0.1818705 , SAR1: -0.03997979 , SMA1: -0.6673795NULL
```

```
checkresiduals(arima_ts)
```



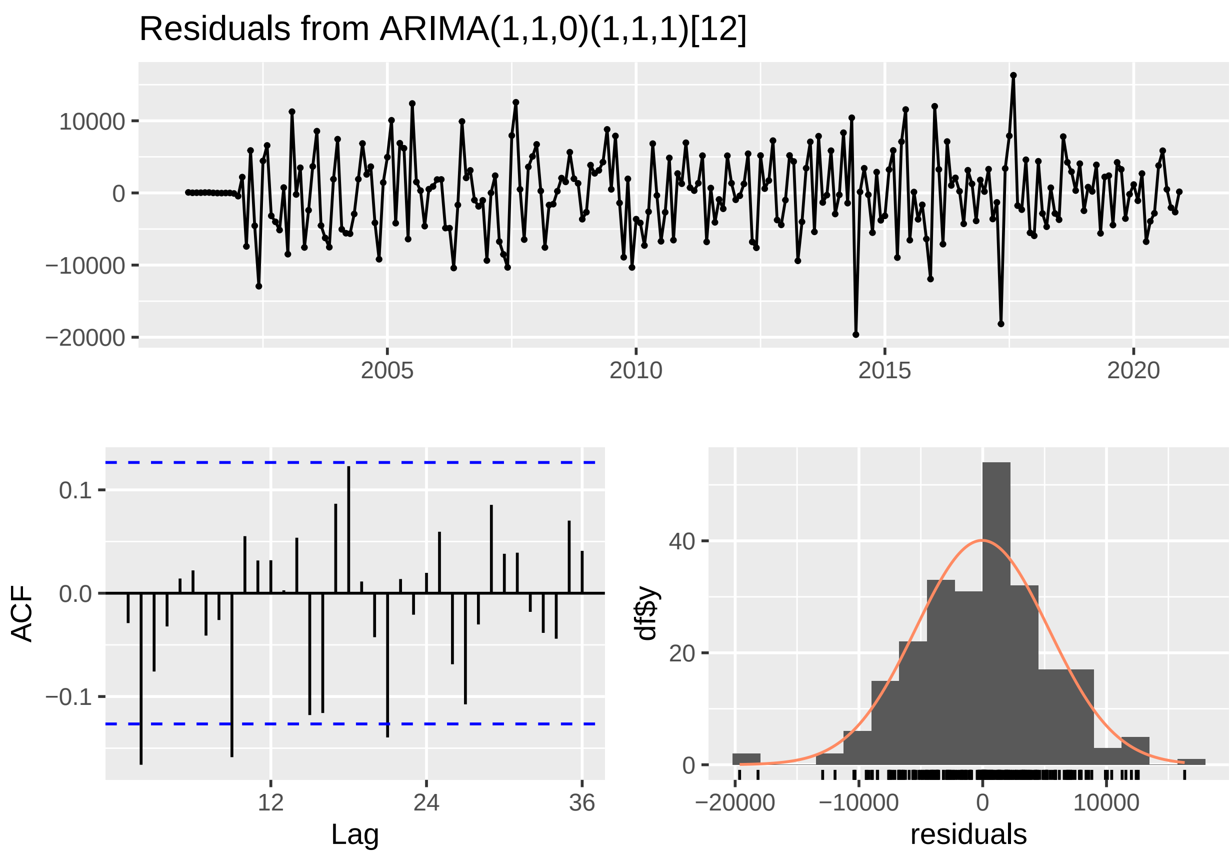

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,0)(1,1,1)[12]
## Q* = 36.636, df = 21, p-value = 0.01853
##
## Model df: 3.   Total lags used: 24
```

The residuals appear to follow a white noise, as there are no clear patterns. They also loosely follow a normal distribution.

Q8

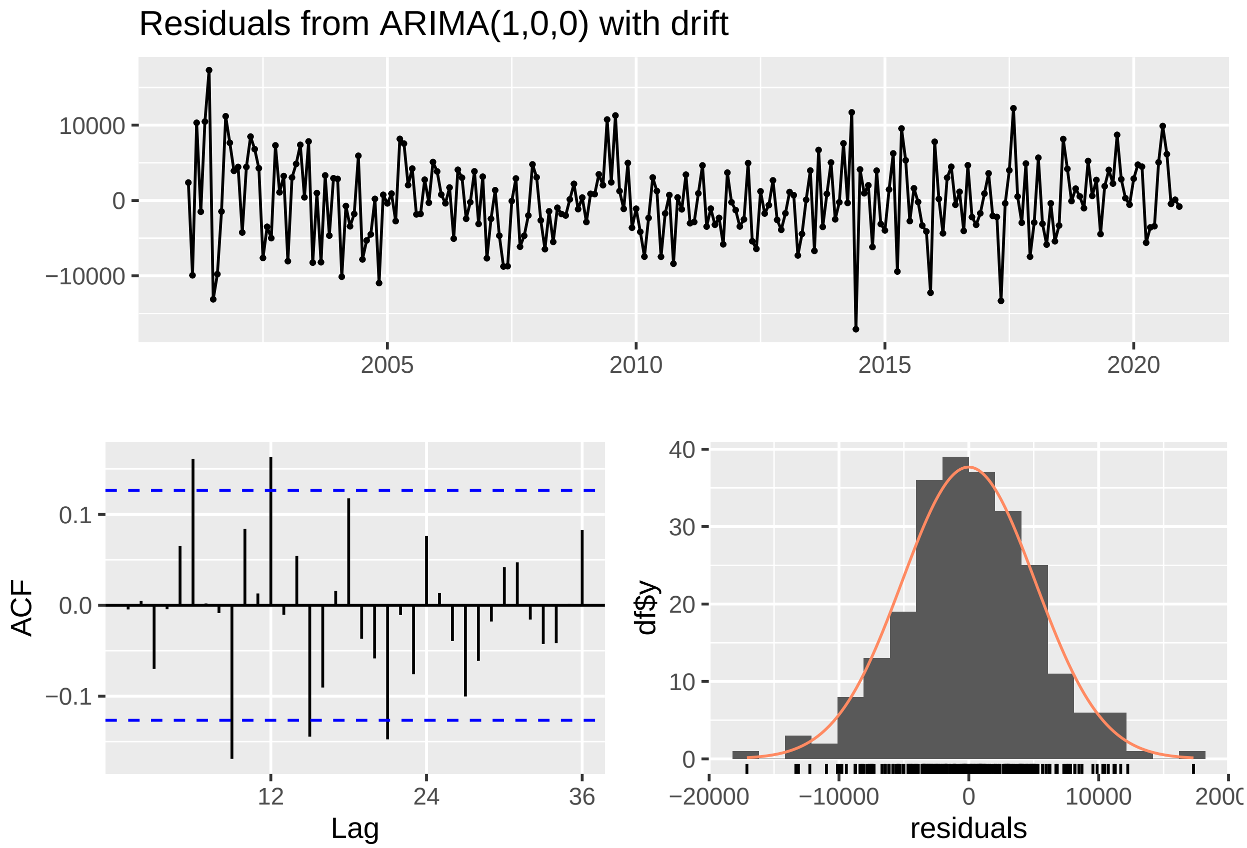
Compare the residual series for Q7 and Q6. Can you tell which ARIMA model is better representing the Natural Gas Series? Is that a fair comparison? Explain your response.

```
checkresiduals(arima_ts)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,0)(1,1,1)[12]
## Q* = 36.636, df = 21, p-value = 0.01853
##
## Model df: 3.   Total lags used: 24
```

```
checkresiduals(arima_deseason)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0) with drift
## Q* = 46.658, df = 23, p-value = 0.002475
##
## Model df: 1.   Total lags used: 24
```

When viewing both residual series above, the deseasoned appears to be better representative as the residuals are more normally distributed. This is not a fair comparison as the data is deseasoned and decomposed, so it is easier to fit a model.

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?

```r
auto.arima(ts_deseason)
```

```
## Series: ts_deseason
## 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    328.8248
##
## sigma^2 = 27791547:  log likelihood = -2384.89
## AIC=4783.79   AICc=4784.27   BIC=4808.12
```

The order of the best ARIMA model is ARIMA(3,1,0)(1,0,1)[12] with drift. This does not match what I specified, as I had guessed ARIMA(1,1,0), and had not considered the seasonal terms.

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
auto.arima(ts)
```

```
## Series: ts
## 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   1.3222   0.1014   1.3215  0.9215    44.1285
##
## sigma^2 = 27958166:  log likelihood = -2278.46
## AIC=4574.91   AICc=4575.74   BIC=4605.77
```

The order of the best ARIMA model is ARIMA(2,0,1)(2,1,2)[12] with drift. I improperly identified ARIMA(1,1,0)(1,1,1) to be the best match.