ENV 790.30 - Time Series Analysis for Energy Data | Spring 2021

Assignment 6 - Due date 03/26/21

Traian Nirca

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 project open the first thing you will do is change "Student Name" on line 3 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. Rename the pdf file such that it includes your first and last name (e.g., "LuanaLima_TSA_A06_Sp21.Rmd"). Submit this pdf using Sakai.

Set up

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

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

```
library(tseries)
```

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.\

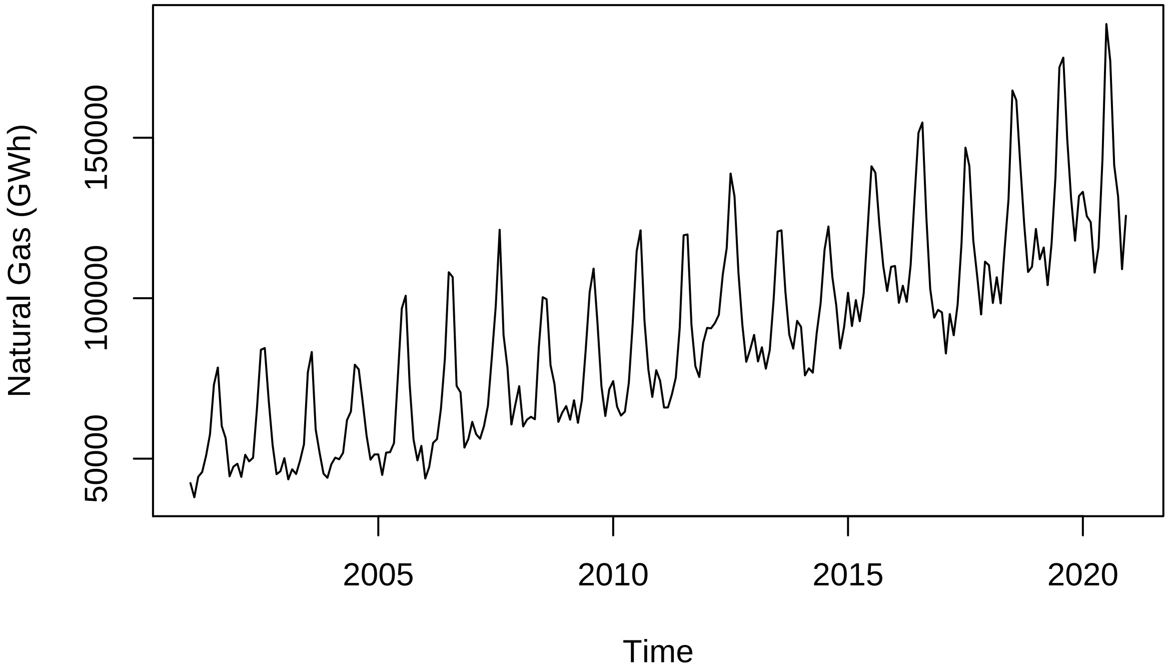
Q1

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.

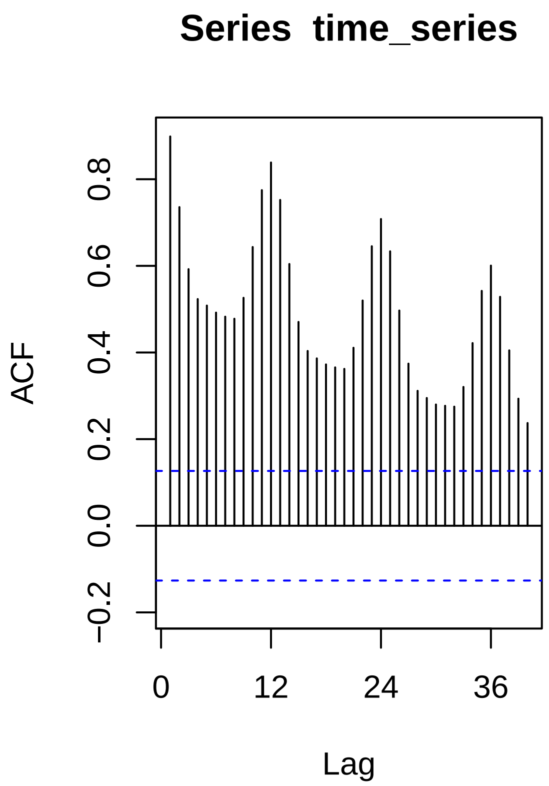
```
raw_data <- read.csv(file="../Data/Net_generation_United_States_all_sectors_monthly.csv",header=FALSE,s

work_data <- data.frame("Natural Gas"=raw_data[,4])
work_data <- work_data[nrow(raw_data):1,]

time_series <- ts(work_data, frequency = 12, start = c(2001,1))
head(time_series)
```

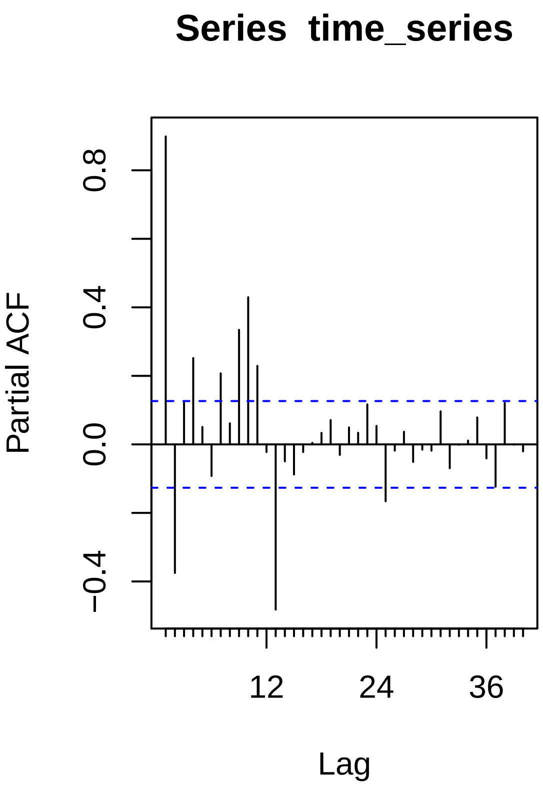
```
##           Jan      Feb      Mar      Apr      May      Jun
## 2001 42388.66 37966.93 44364.41 45842.75 50934.21 57603.15
```

```
plot(time_series, ylab="Natural Gas (GWh)")
```



```
par(mfrow=c(1,2))
Acf(time_series, lag.max = 40)
Pacf(time_series, lag.max = 40)
```

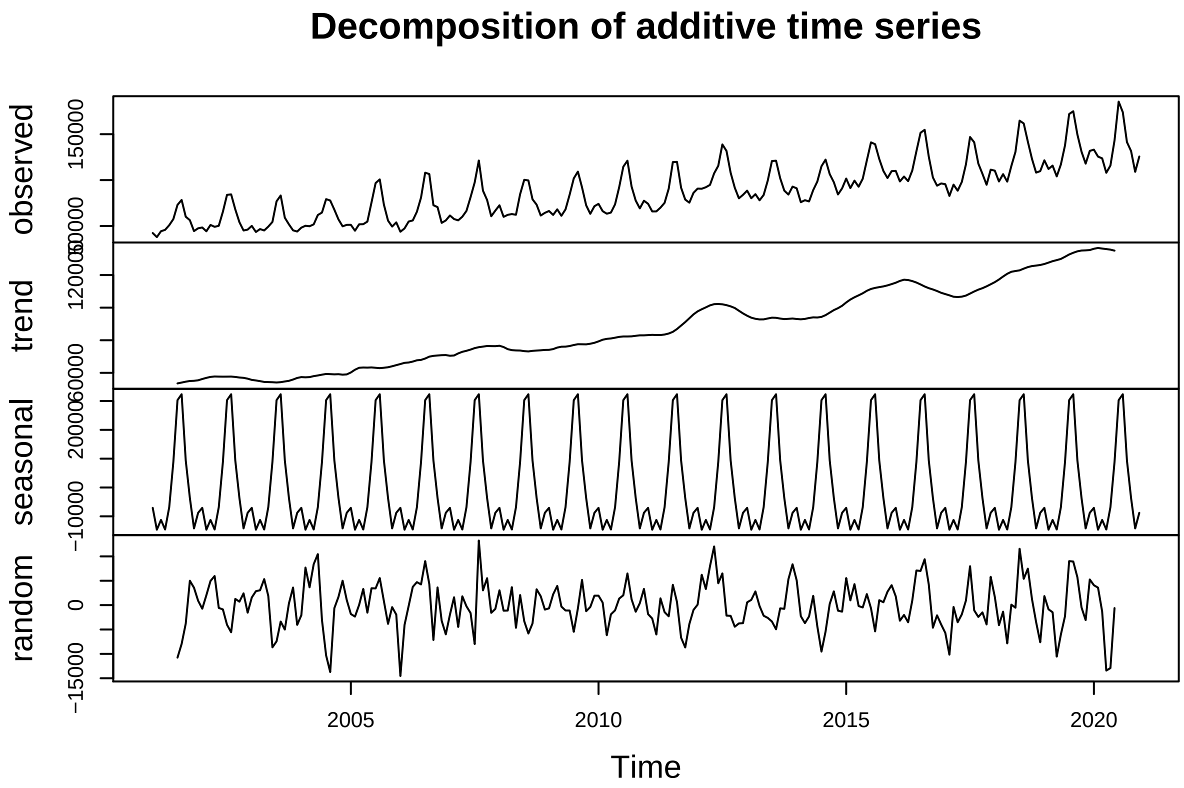




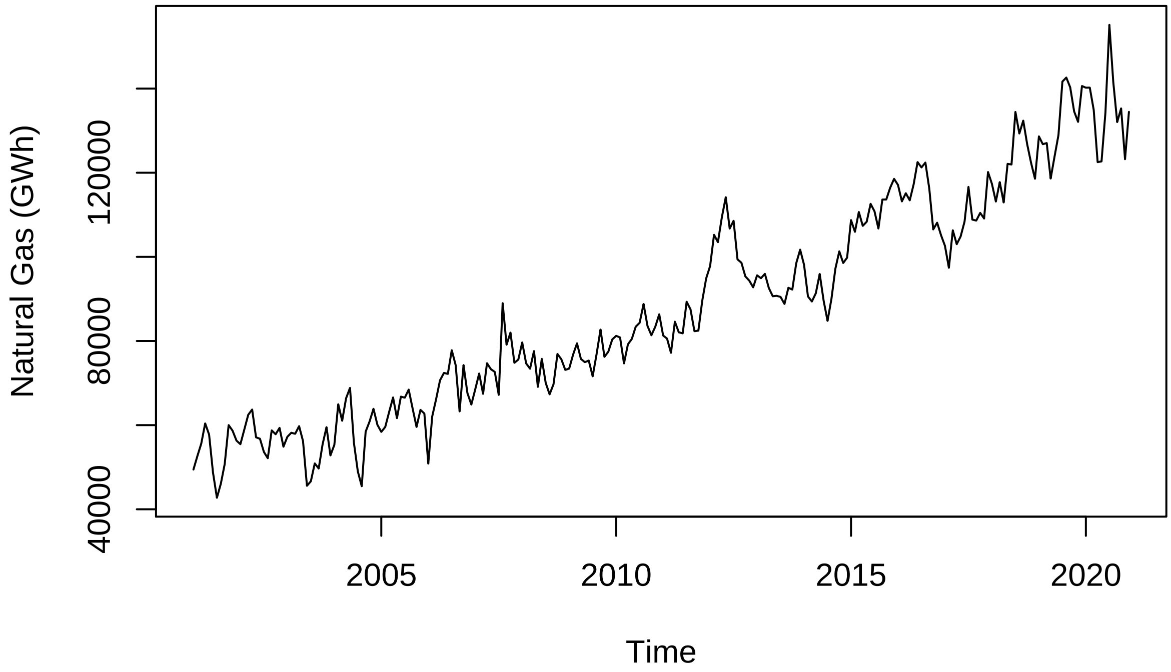
Q2

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.

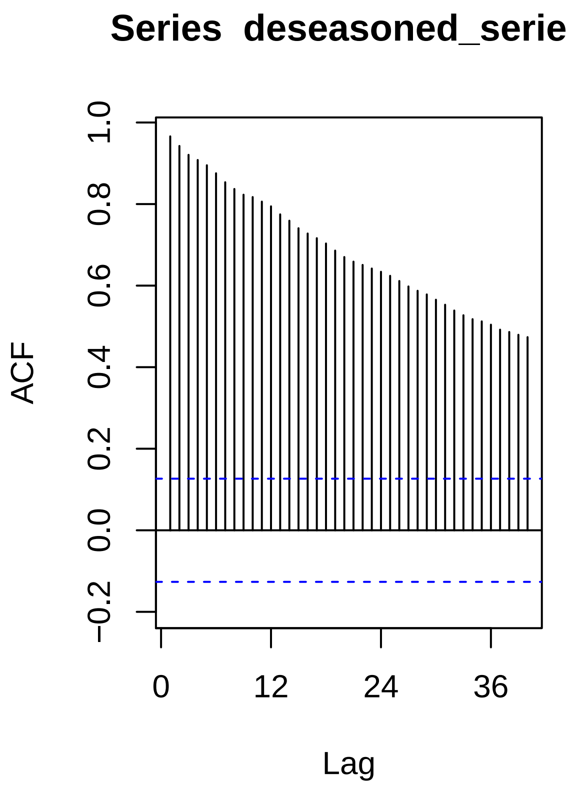
```
decomposed_series <-decompose(time_series,"additive")
plot(decomposed_series)
```

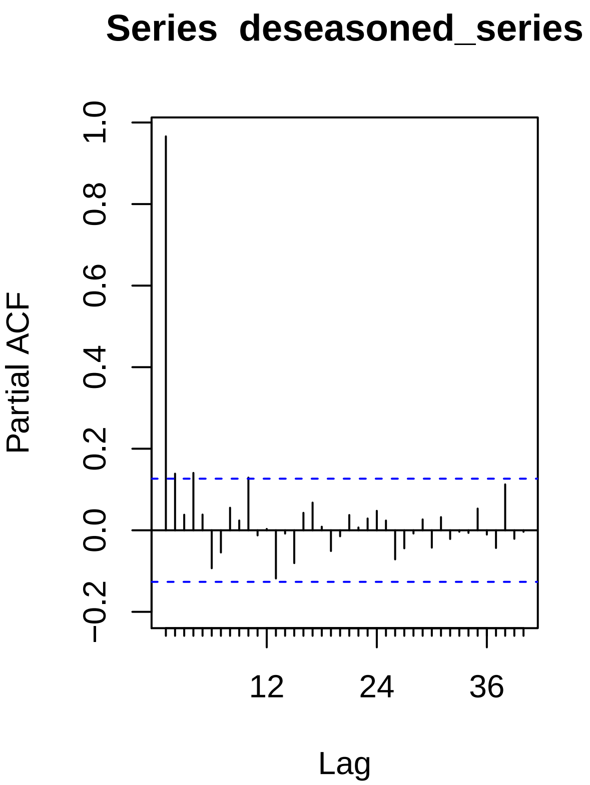


```
deseasoned_series <- decomposed_series[["x"]]- decomposed_series[["seasonal"]]

plot(deseasoned_series, ylab="Natural Gas (GWh)")
```



```
par(mfrow=c(1,2))
Acf(deseasoned_series, lag.max = 40)
Pacf(deseasoned_series, lag.max = 40)
```





We no longer observe the seasonality in the new plot, as well as in the ACF. We only see a rapid decrease.

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 Test
adf.test(deseasoned_series)
```

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

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

```
#MannKendall Test
#MannKendall(deseasoned_series)
```

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.

> Answer: Our value of p in the Dickey-Fuller test is lower than 0.05, which means the series is stationary. No differencing is needed, so d =0. The ACF has a slow decay and a clear cut off at lag 1 in the PACF plot. It is an auto-regressive model, ARIMA(1,0,0), with p=1 and q =0.

Q5

Use *Arima*() from package "forecast" to fit an ARIMA model to your series considering the order estimated in Q4. Should you allow for 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.

```
arima_fit <- Arima(deseasoned_series, order=c(1,0,0),include.mean = TRUE, include.drift=TRUE)

#print(cat(arima_fit$coef, arima_fit$sigma2))
print(arima_fit$coef)
```

```
##          ar1    intercept        drift
## 7.182166e-01 4.480049e+04 3.593965e+02
```

```
print(cat("sigma2 =", arima_fit$sigma2))
```

```
## sigma2 = 26630969NULL
```

```
print(cat("Log likelihood = ", arima_fit$loglik))
```

```
## Log likelihood =  -2391.109NULL
```

```
print(cat("AIC=",arima_fit$aic))
```

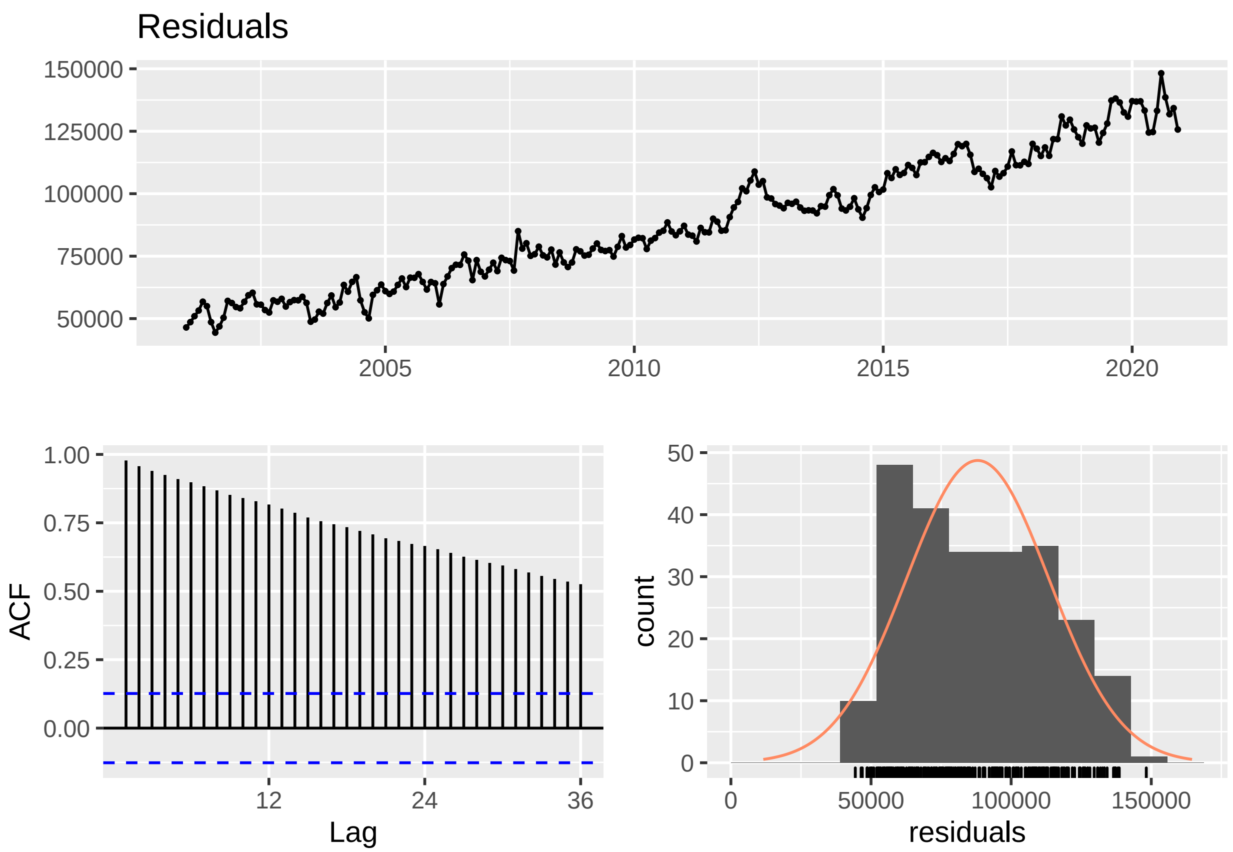
```
## AIC= 4790.217NULL
```

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_fit$fitted,test="LB", plot=TRUE)
```

```
## Warning in modeldf.default(object): Could not find appropriate degrees of
## freedom for this model.
```



The function *checkresiduals* performs the Ljung-Box test which tests if the residuals are indistinguishable from white noise. If they are not, it means that the model is not adequate.

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.

```
#ADF Test
adf.test(time_series)
```

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

```
##
##  Augmented Dickey-Fuller Test
##
## data:  time_series
## Dickey-Fuller = -8.9602, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```
#MannKendall Test
#MannKendall(time_series)
```

Answer: p is still smaller than 0.05, so D=0 For the seasonal component, we are only interested in the seasonal lags 12, 24, 36, etc. We have multiple negative spikes in the PACF at lag 12, 24, 36… We assume it is a MA process, with Q=1. Thus we have an ARIMA(1,0,0)x(0,0,1) model

```
arima_fit1 <- Arima(time_series, order=c(1,0,0), seasonal= c(0,0,1),include.mean = TRUE, include.drift=T


print(arima_fit1$coef)
```

```
##           ar1          sma1    intercept        drift
## 7.079057e-01 6.227830e-01 4.480294e+04 3.580466e+02
```

```
print(cat("sigma2 =", arima_fit1$sigma2))
```

```
## sigma2 = 84948680NULL
```

```
print(cat("Log likelihood = ", arima_fit1$loglik))
```

```
## Log likelihood =  -2532.738NULL
```

```
print(cat("AIC=",arima_fit1$aic))
```

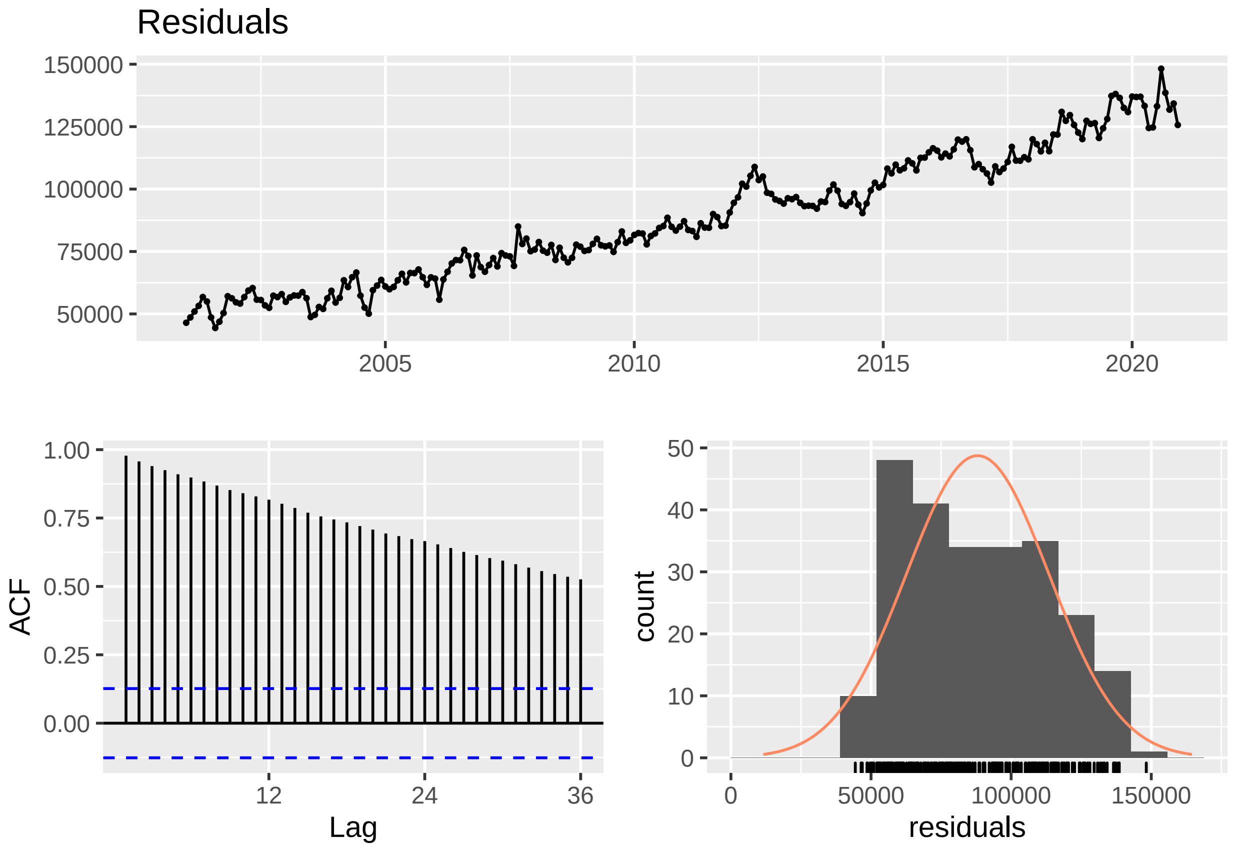
```
## AIC= 5075.475NULL
```

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.

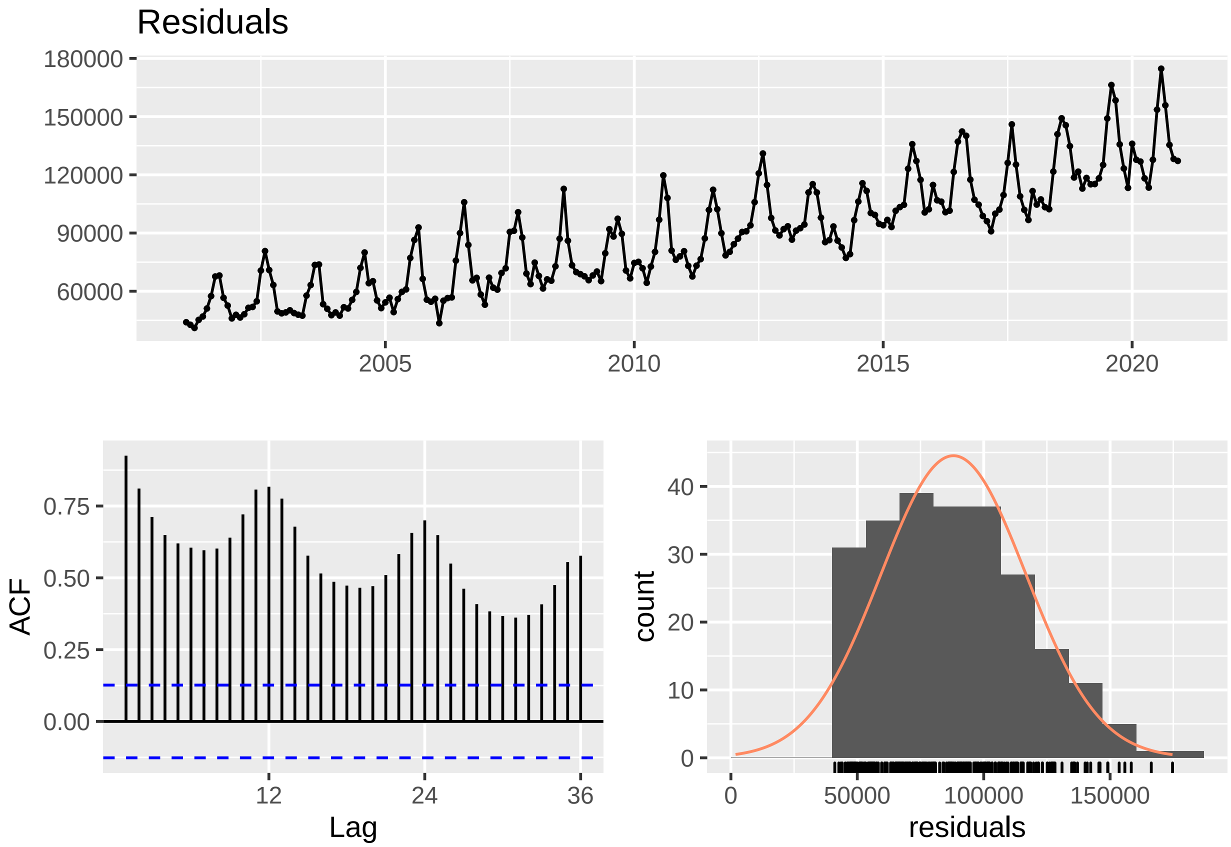
```
checkresiduals(arima_fit$fitted,test="LB", plot=TRUE)
```

```
## Warning in modeldf.default(object): Could not find appropriate degrees of
## freedom for this model.
```



```
checkresiduals(arima_fit1$fitted,test="LB", plot=TRUE)
```

```
## Warning in modeldf.default(object): Could not find appropriate degrees of
## freedom for this model.
```



For the new model, we still have a seasonal component in the residuals. This means that the selected model was not the right one. It is not a fair comparison however, as the first model has been deseasoned.

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 correct orders. The intention of the assignment is to walk you to the process and help you figure out what you did wrong (if you did anything wrong!).

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?

```
auto.arima(deseasoned_series)
```

```
## Series: deseasoned_series
## 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 estimated as 26980609:  log likelihood=-2383.11
## AIC=4774.21   AICc=4774.38   BIC=4788.12
```

Auto arima estimates that ARIMA(1,1,1) is the best model, which is not at all what I specified in Q4. It seems strange to me, as I had $p<0.05$, d should be 0.

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?

```
auto.arima(time_series)
```

```
## Series: time_series
## 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 estimated as 27569124:  log likelihood=-2279.54
## AIC=4567.08   AICc=4567.26   BIC=4580.8
```

Again, auto arima indicates a different model: ARIMA(1,0,0)(0,1,1), which is not what I specified in Q7, but it seems strange, as the non seasonal component does not match the one in Q9.