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

Assignment 7 - Due date 03/20/23

John Rooney

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(forecast)
```

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

```
library(tseries)
library(lubridate)
```

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

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

```
library(Kendall)
library(sarima)
```

```
## Loading required package: stats4
```

```
##
## Attaching package: 'sarima'

## The following object is masked from 'package:stats':
##
##     spectrum
```

Importing and processing the data set

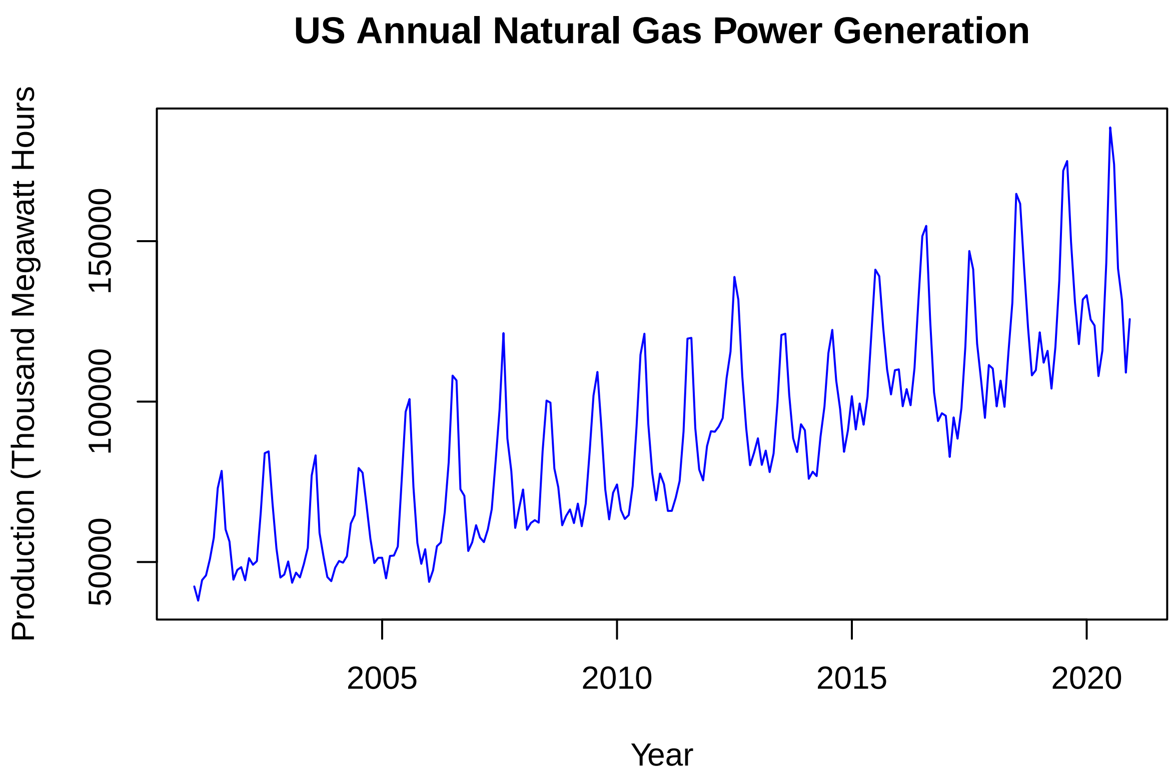
Consider the data from the file "Net_generation_United_States_all_sectors_monthly.csv". The data corresponds to the monthly net generation from January 2001 to December 2020 by source and is provided by the US Energy Information and Administration. **You will work with the natural gas column only**.

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

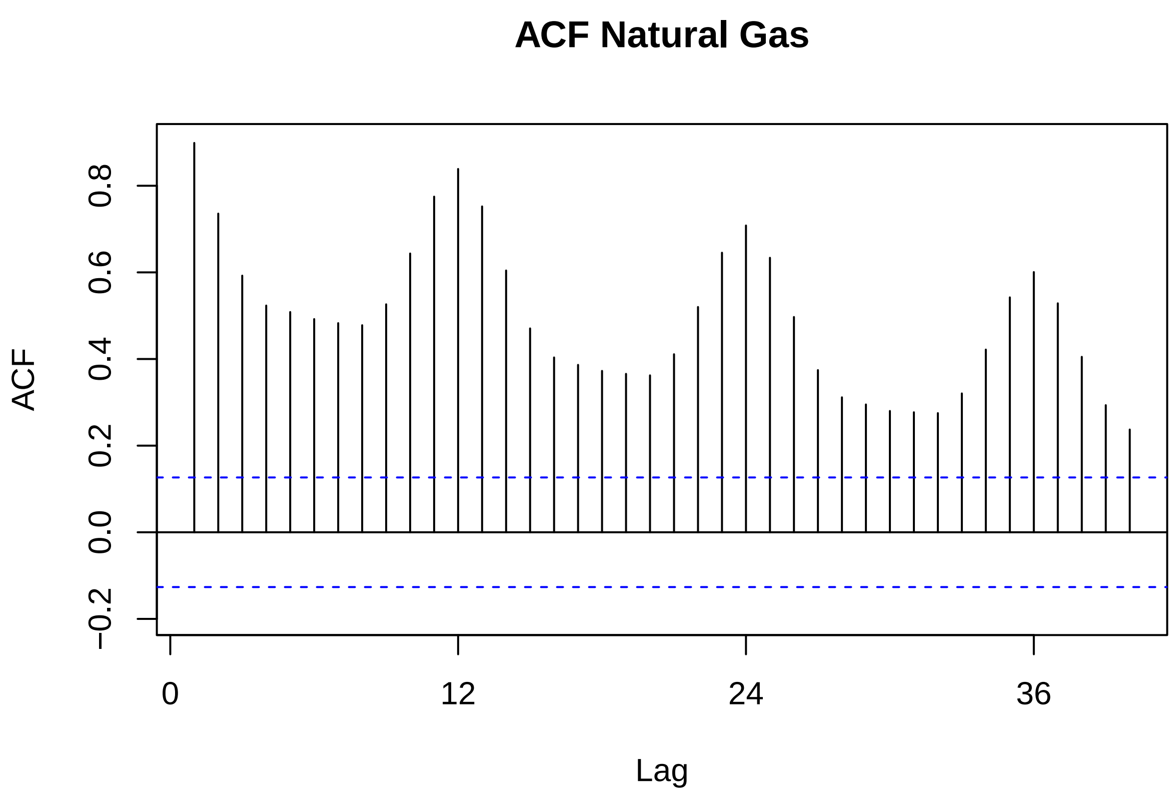
Q1

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.

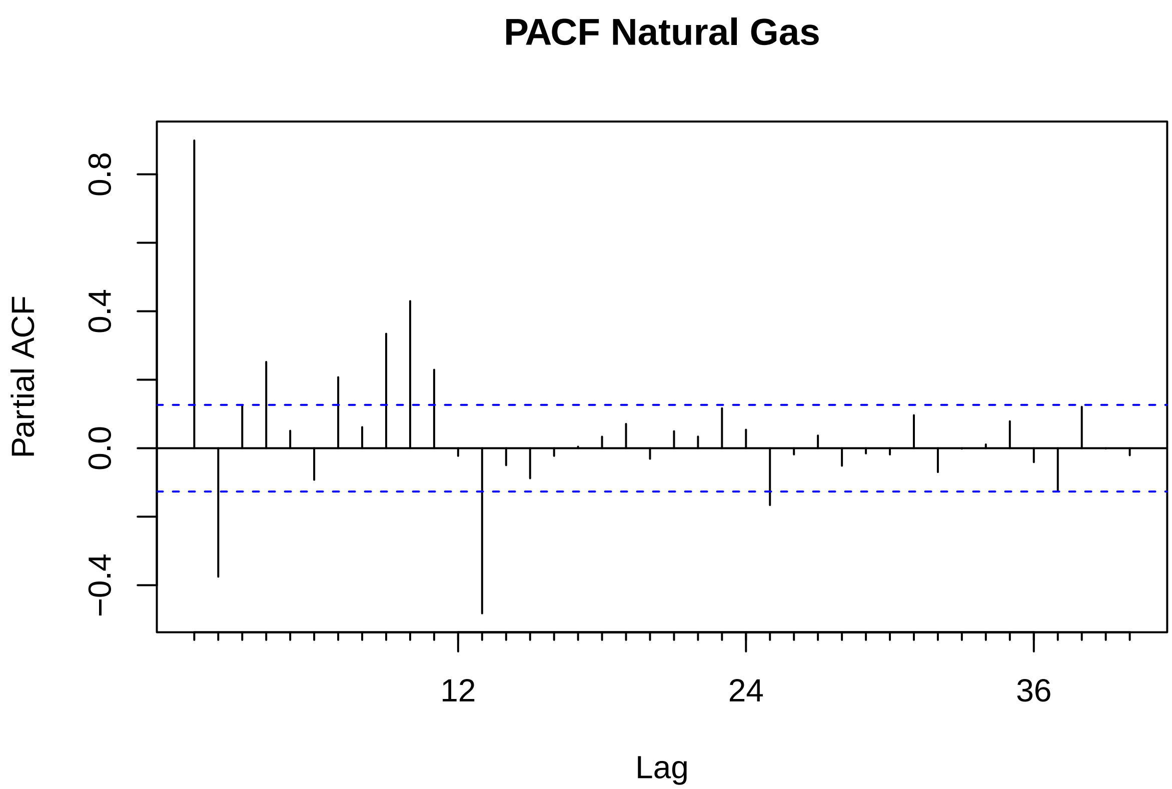
```r
#Import file
net.generation <- read.csv(file="./Data/Net_generation_United_States_all_sectors_monthly.csv",header=TRU

#Fix date
net.generation$Month <- my(net.generation$Month)

#Create time series with rev because of date order
gas.ts <- ts(rev(net.generation$natural.gas.thousand.megawatthours), start=c(2001,1), frequency=12)

#plot time series
plot(gas.ts, type="l", col="blue", xlab="Year", ylab="Production (Thousand Megawatt Hours", main="US Ann
```




```r
#ACF and PACF plots
Acf(gas.ts, lag.max=40, main="ACF Natural Gas")
```



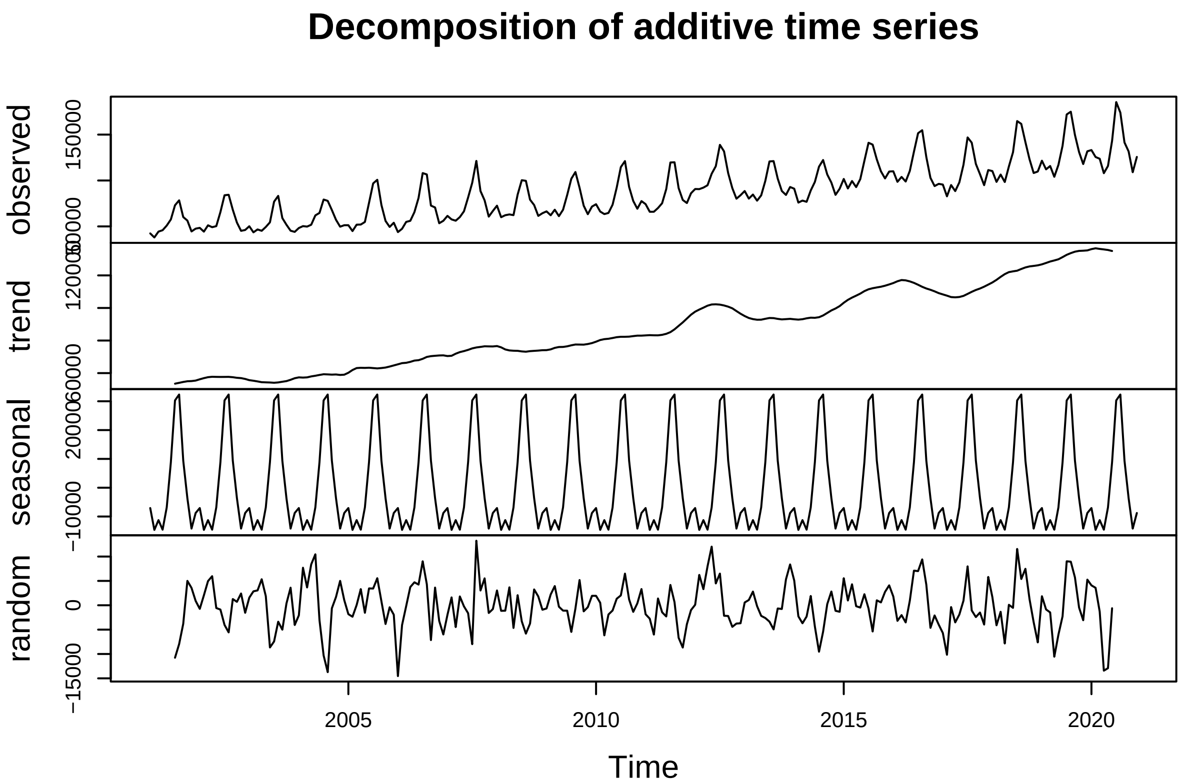
```r
Pacf(gas.ts, lag.max=40, main="PACF Natural Gas")
```



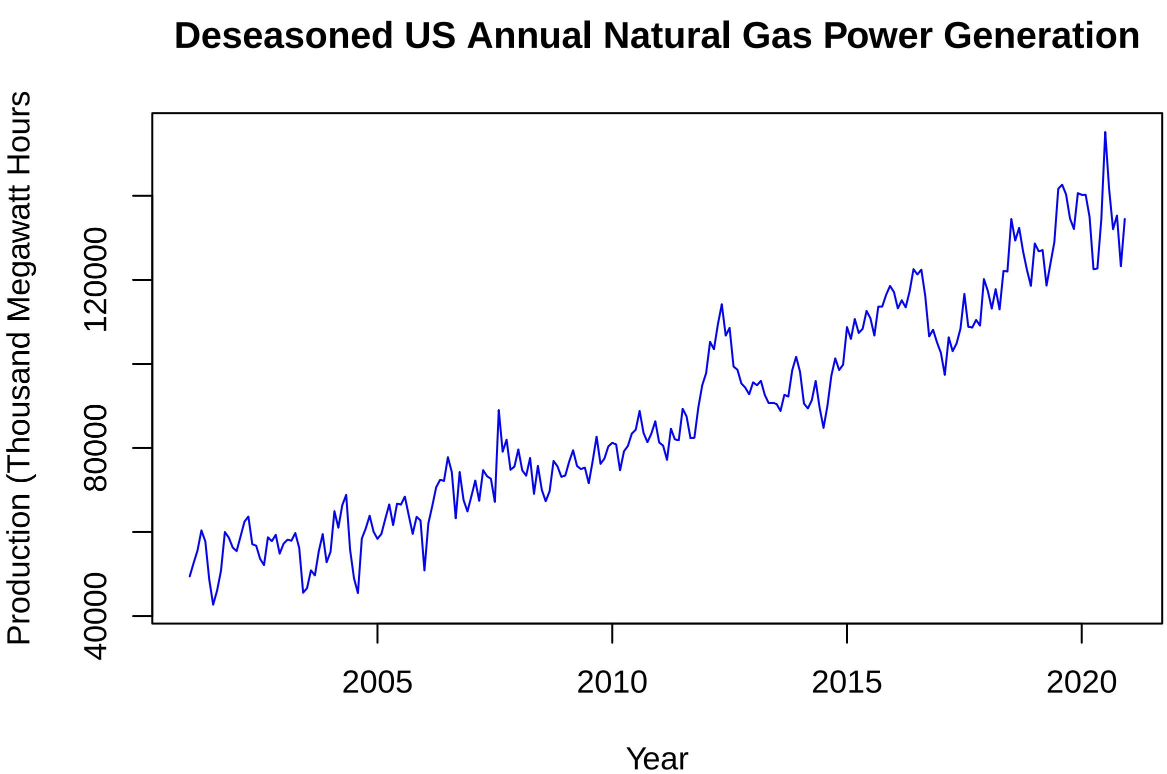
Q2

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.

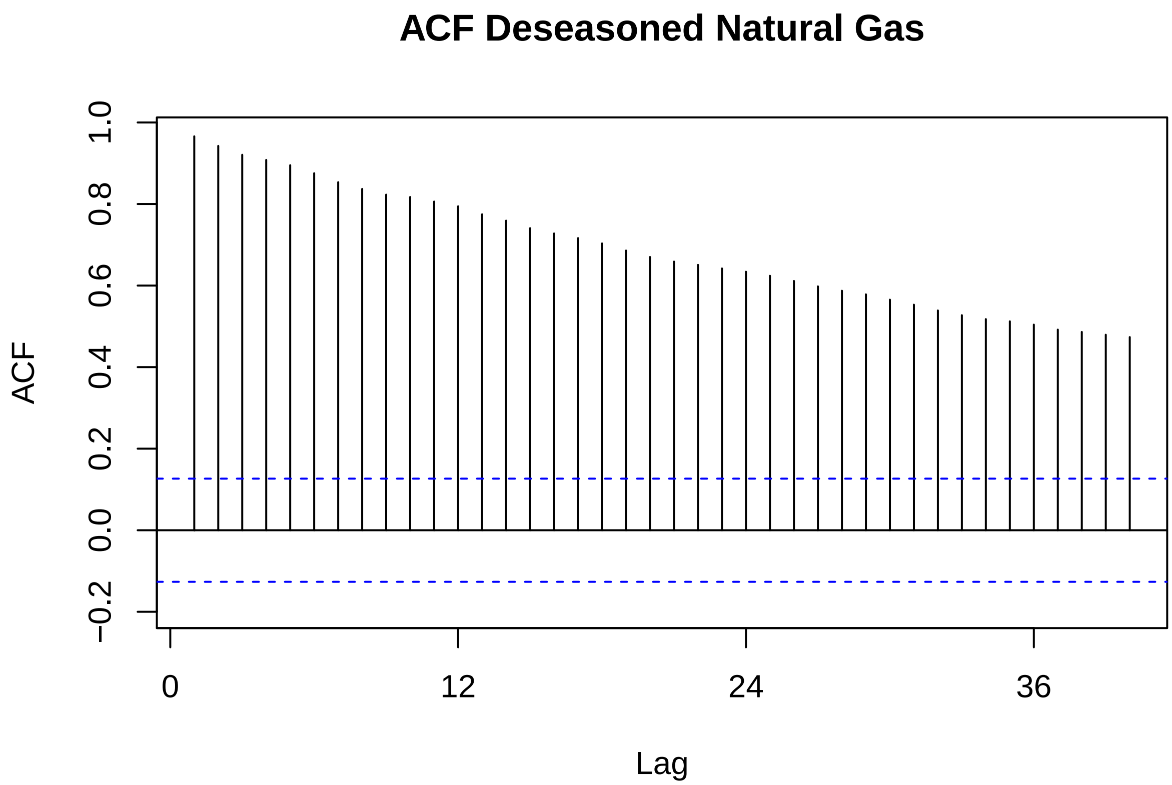
```r
#decompose time series
decompose.gas <- decompose(gas.ts, type="additive")
plot(decompose.gas)
```



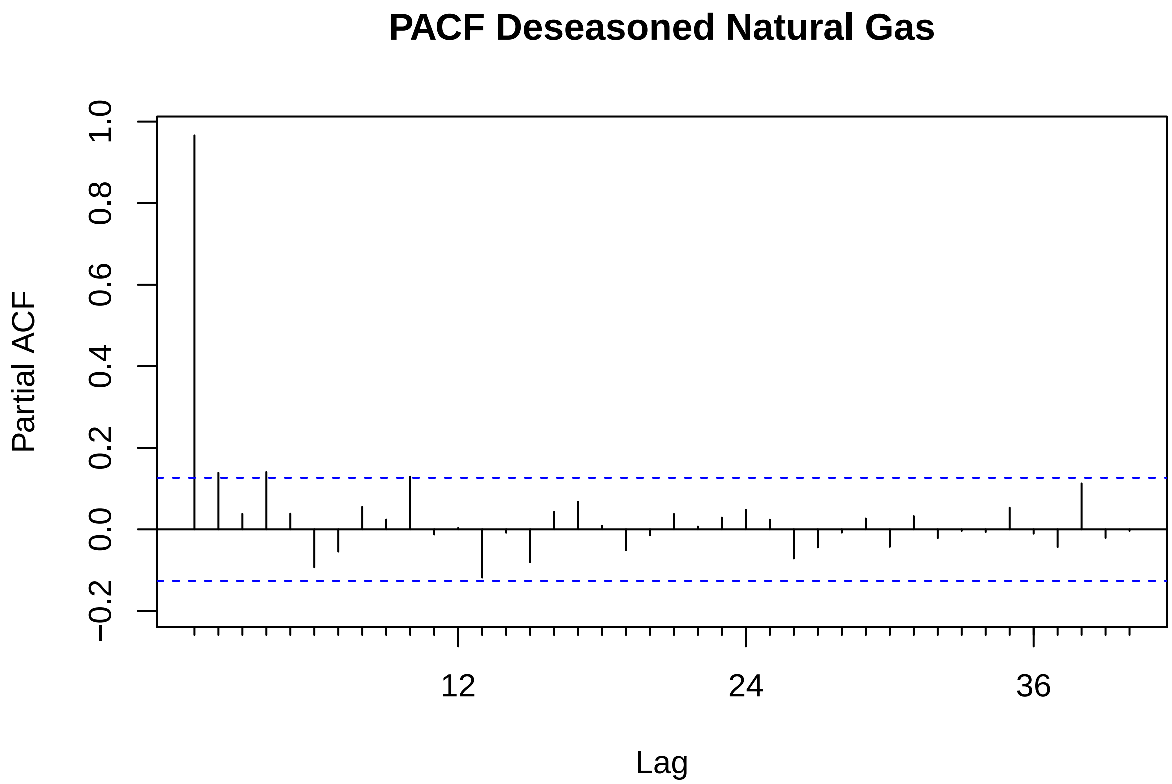

```
#deseason and plot
deseason.gas <- seasadj(decompose.gas)
plot(deseason.gas, type="l", col="blue", xlab="Year", ylab="Production (Thousand Megawatt Hours", main=
```



```r
#ACF and PACF
Acf(deseason.gas, lag.max=40, main="ACF Deseasoned Natural Gas")
```




```r
Pacf(deseason.gas, lag.max=40, main="PACF Deseasoned Natural Gas")
```



> Answer: Especially looking at the ACF plots, this clearly removed seasonality.

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(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
```

```
print(summary(MannKendall(deseason.gas)))
```

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

Answer: The ADF test returned a p-value of 0.01. Given this p-value, we can reject the null hypothesis in favor of the alternative, which tells us that our series is stationary. The Mann Kendall test returned a two-sides p-value >0.001. Given this, we know that trend is not statistically significant.

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: Based on the plots and the test results, I believe this is an (1,0,0) ARIMA model. The ACF shows exponential decay and the PACF shows 1 lag, which tells us it is an AR (1). We know from the ADF and Mann Kendall that trend is not statistically significant so we do not need to difference the series.

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.

```
Model_100 <- Arima(deseason.gas,order=c(1,0,0),include.drift=TRUE)
print(Model_100)
```

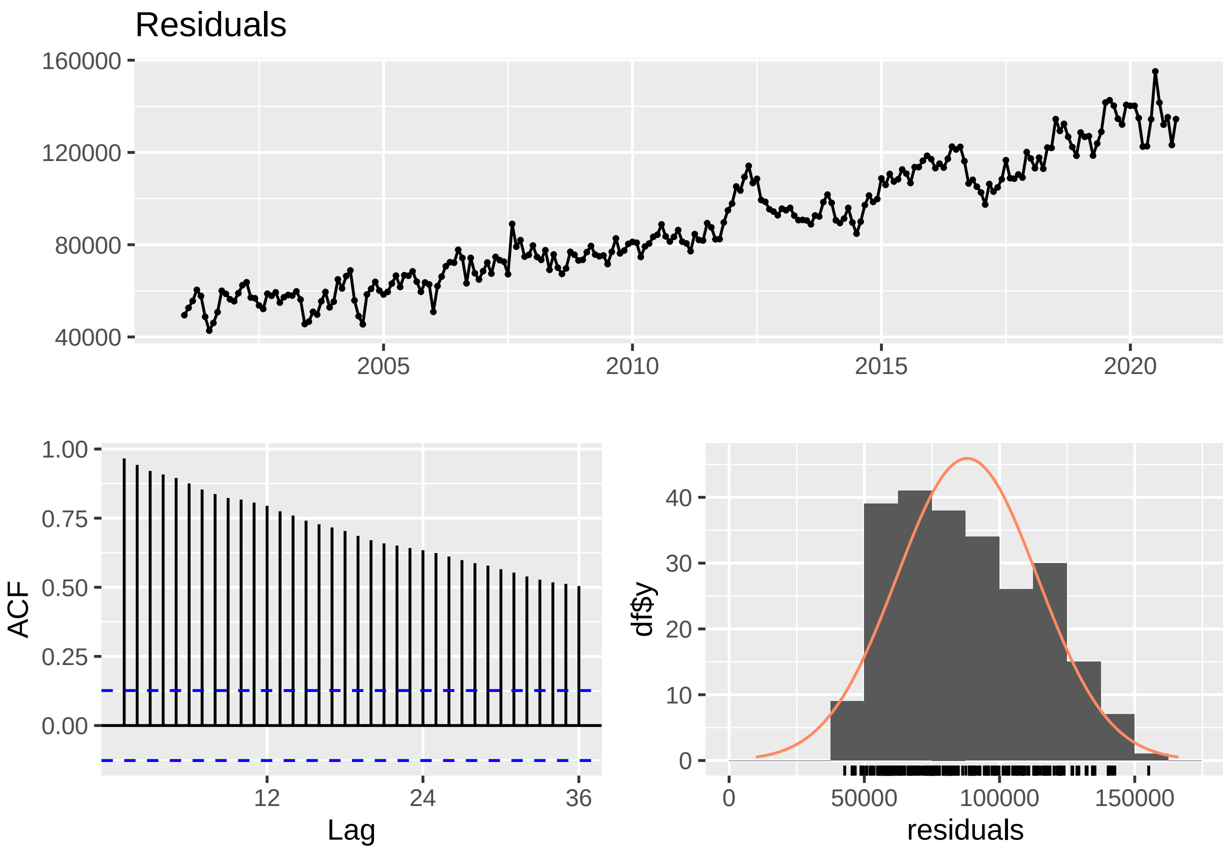
```
## Series: deseason.gas
## ARIMA(1,0,0) with drift
##
## Coefficients:
##          ar1  intercept     drift
##       0.7182   44800.49  359.3965
## s.e.  0.0445    2296.04   16.4305
##
## sigma^2 = 26630969:  log likelihood = -2391.11
## AIC=4790.22   AICc=4790.39   BIC=4804.14
```

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(deseason.gas)
```

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



>

Answer: The residuals series do look like a white noise series to me as there's no clear seasonality due to us removing it. There is still a trend visible, however.

Modeling the original series (with seasonality)

Q7

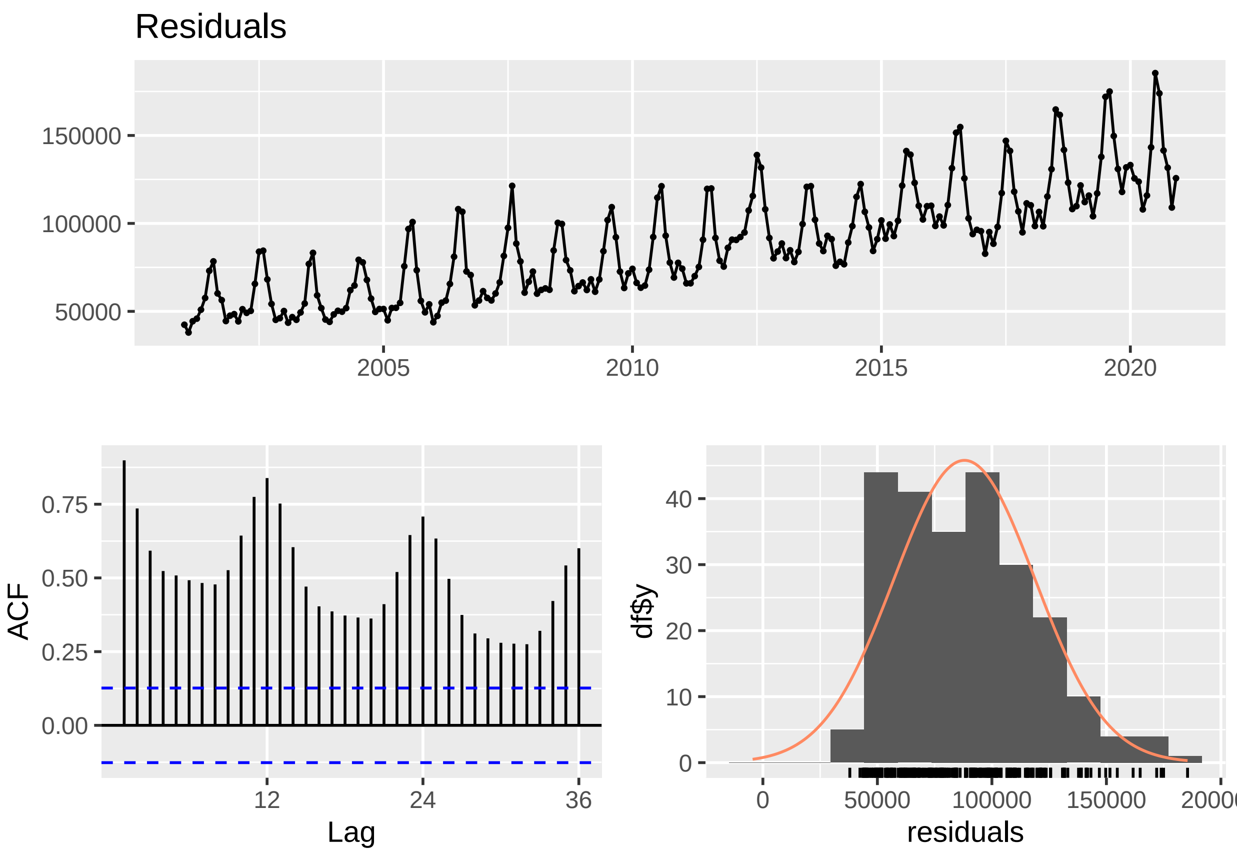
Repeat Q4-Q6 for the original series (the complete series that has the seasonal component). Note that when you model the seasonal series, you need to specify the seasonal part of the ARIMA model as well, i.e., P, D and Q.

```
#SARIMA model
Model_200 <- Arima(gas.ts,order=c(2,0,0),seasonal=c(1,0,0),include.drift=TRUE)
print(Model_200)
```

```
## Series: gas.ts
## ARIMA(2,0,0)(1,0,0)[12] with drift
##
## Coefficients:
##          ar1      ar2    sar1  intercept     drift
##       0.7392  -0.0082  0.8755   47175.59  353.4497
## s.e.  0.0667   0.0694  0.0312   13183.35   79.3085
##
## sigma^2 = 39110228:  log likelihood = -2444.98
## AIC=4901.95   AICc=4902.31   BIC=4922.84
```

```r
#Plot SARIMA residuals
checkresiduals(gas.ts)
```

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



Answer: Looking at the plots for Q1, I would guess that it is a (2,0,0)(1,0,0) model. I say this because of the decay in the ACF and the 2 lags in the PACF, and knowing there's a seasonal component. These residuals do not show a white noise series.

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.

Answer: While it looks like more residuals fall under a normal distribution for the deseasoned series, I don't believe it's fair to compare the two as the deseasoned has been manipulated.

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?

```
arima.auto<-auto.arima(deseason.gas)
print(arima.auto)
```

```
## 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 ARIMA model is (1,1,1). This model sadly does not match what I had in Q4 as it has d=1 and q=1 where I had them equalling 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?

```
sarima.auto <- auto.arima(gas.ts)
print(sarima.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
```

> Answer: This shows that the best model is (1,0,0)(0,1,1). I was pretty far off here!