ENV 790.30 - Time Series Analysis for Energy Data | Spring 2022

Assignment 6 - Due date 03/25/22

Alex Baad

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_A07_Sp22.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)
library(cowplot)
library(lubridate)
```

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

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

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

```
library(ggplot2)
library(Kendall)
```

Importing and processing the data set

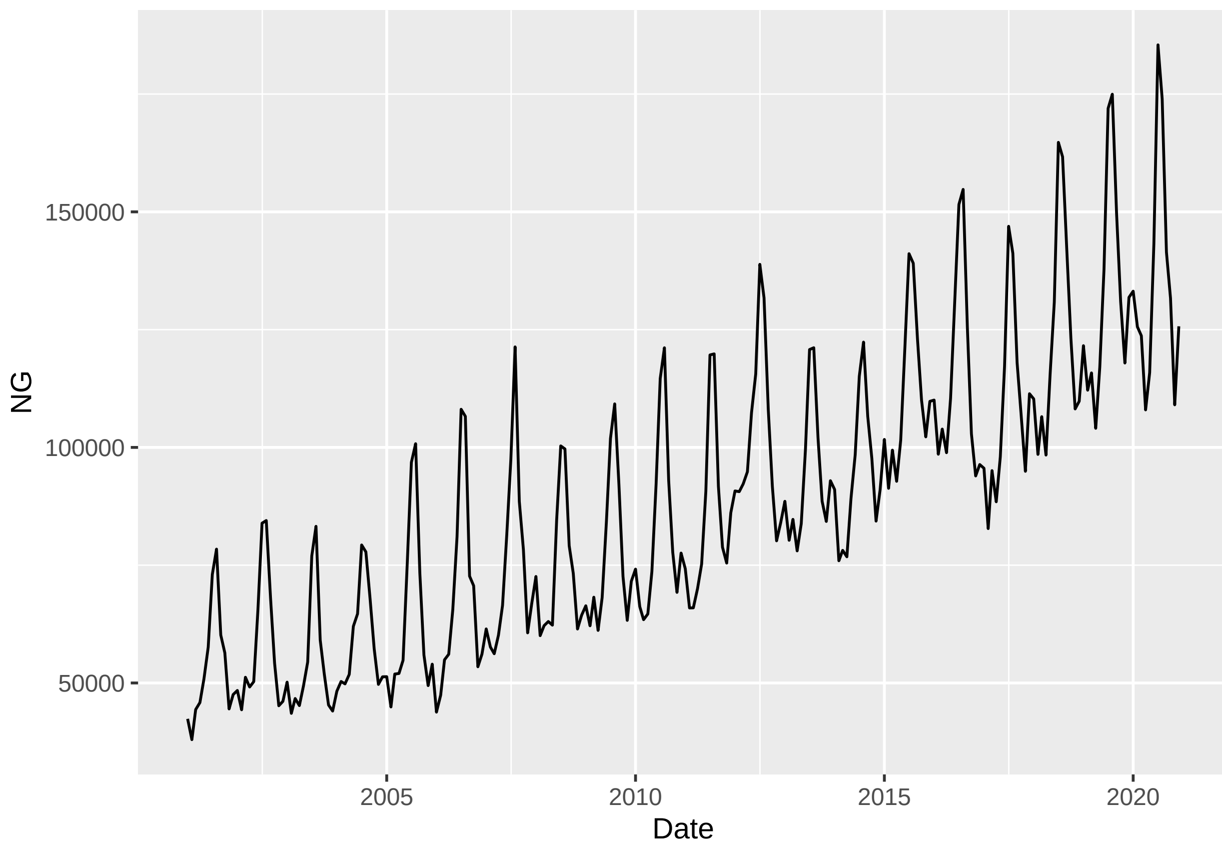
Consider the data from the file "Net_generation_United_States_all_sectors_monthly.csv". The data corresponds to the monthly net generation from January 2001 to December 2020 by source and is provided by the US Energy Information and Administration. **You will work with the natural gas column only**.

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

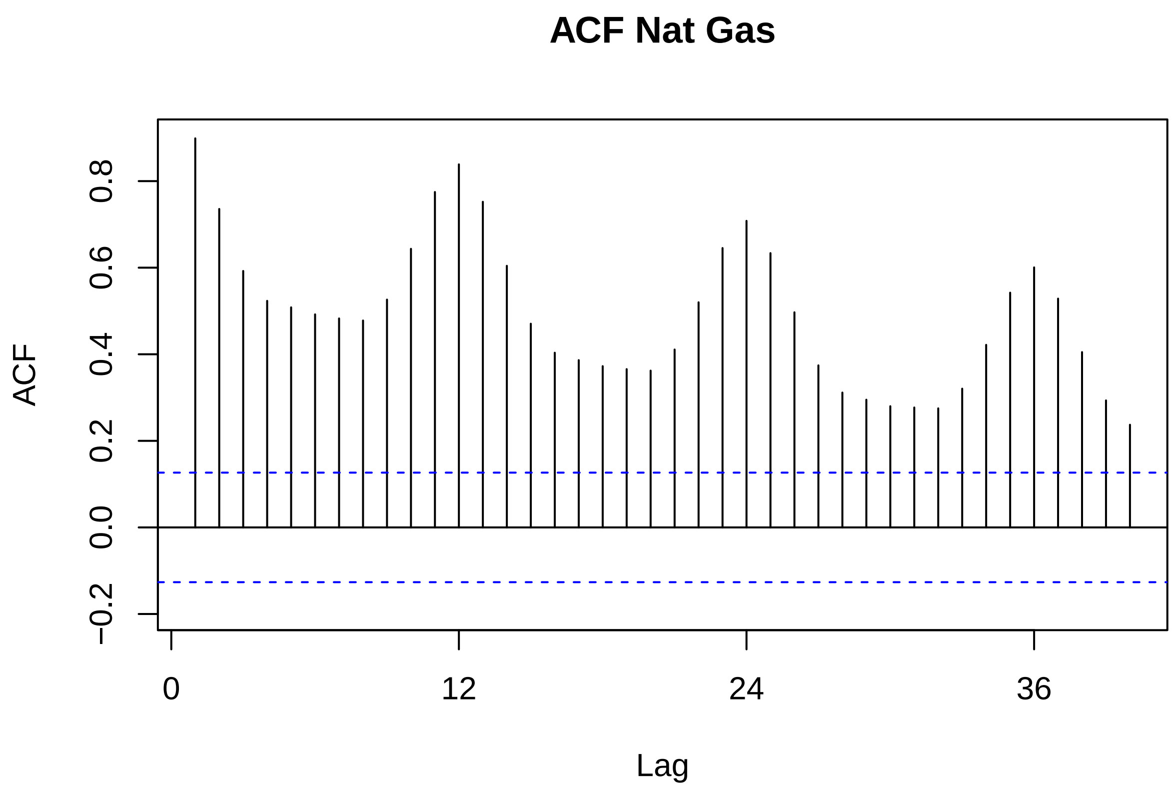
Q1

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.

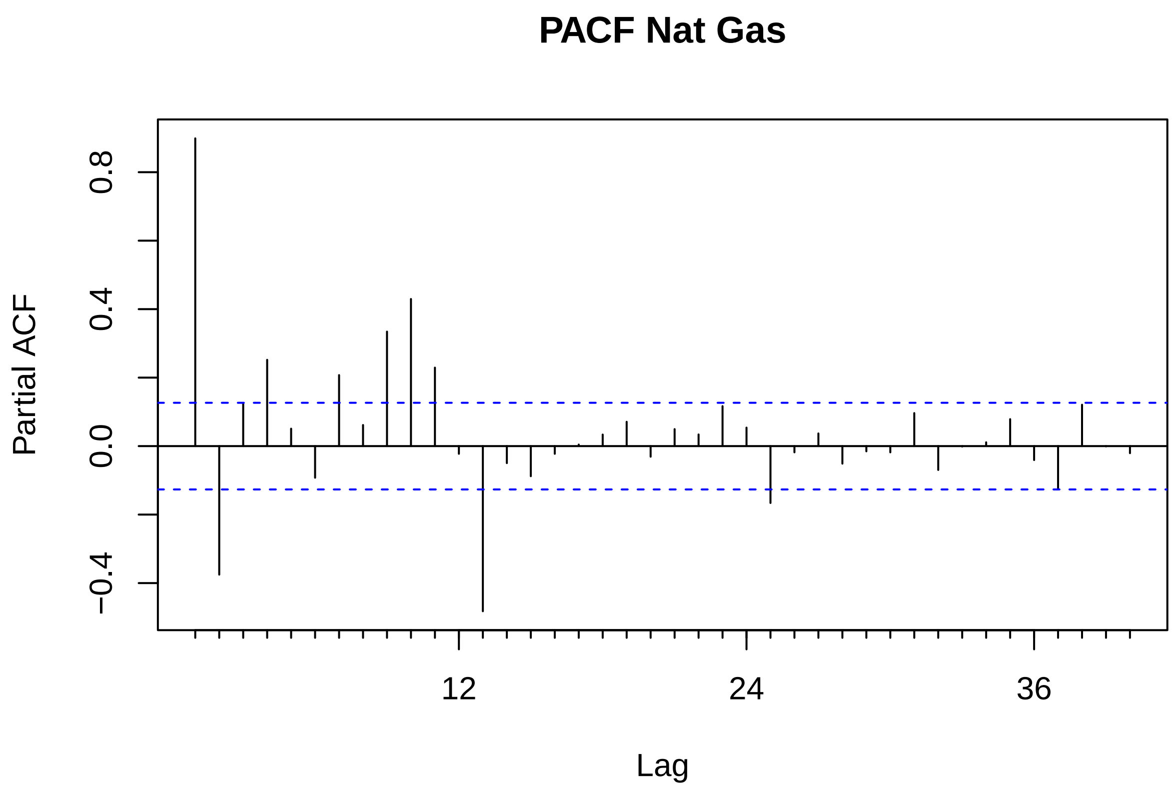
```
nat_gas <- read.csv("./Data/Net_generation_United_States_all_sectors_monthly.csv",skip = 5,header=FALSE
nat_gas <- nat_gas[,-c(2:3)]
nat_gas <- nat_gas[,-c(3:4)]
colnames(nat_gas)=c("Date","NG")
nat_gas <- data.frame("Date"=nat_gas$Date,"NG"=as.numeric(nat_gas$NG))
nat_gas$Date <- my(nat_gas$Date)
nvar <- ncol(nat_gas) - 1
nobs <- nrow(nat_gas)
ts_nat_gas <- ts(nat_gas[,2],frequency=12)
TS_Plot <-
  ggplot(nat_gas, aes(x=Date, y=NG)) +
      geom_line()
plot(TS_Plot)
```




```
Acf(ts_nat_gas,lag.max=40,main=paste("ACF Nat Gas"))
```




```
Pacf(ts_nat_gas,lag.max=40,main=paste("PACF Nat Gas"))
```

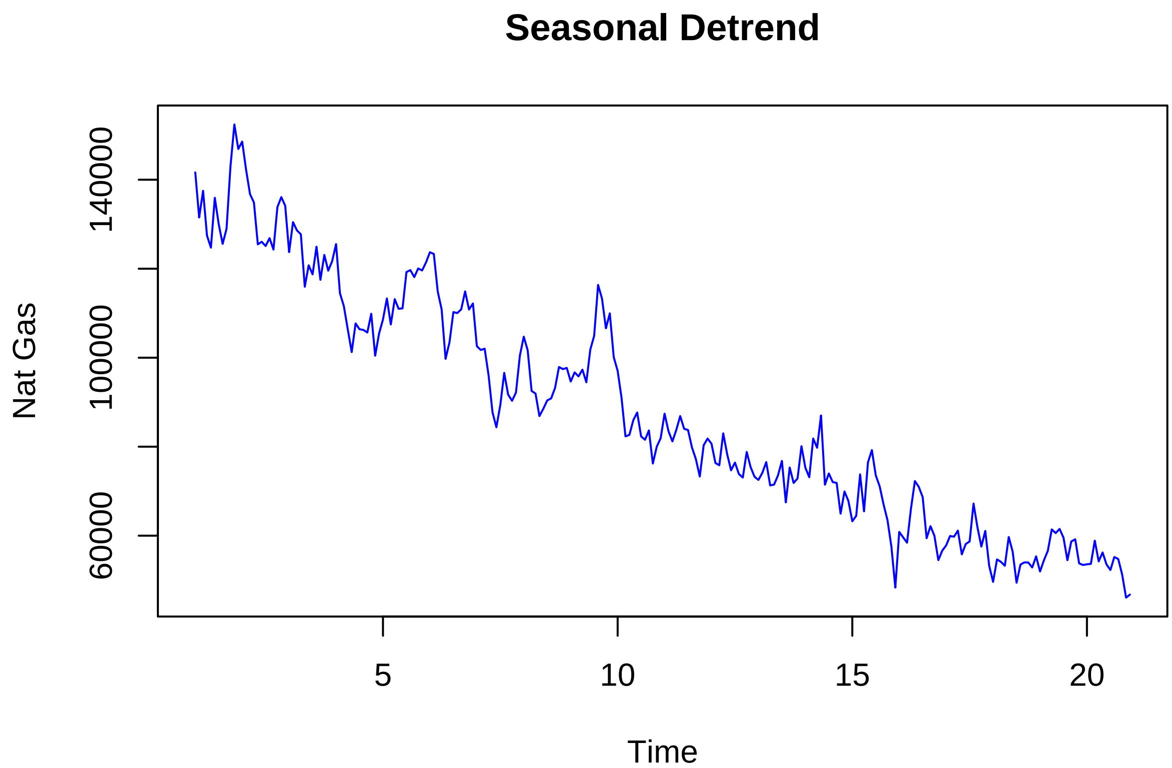


Q2

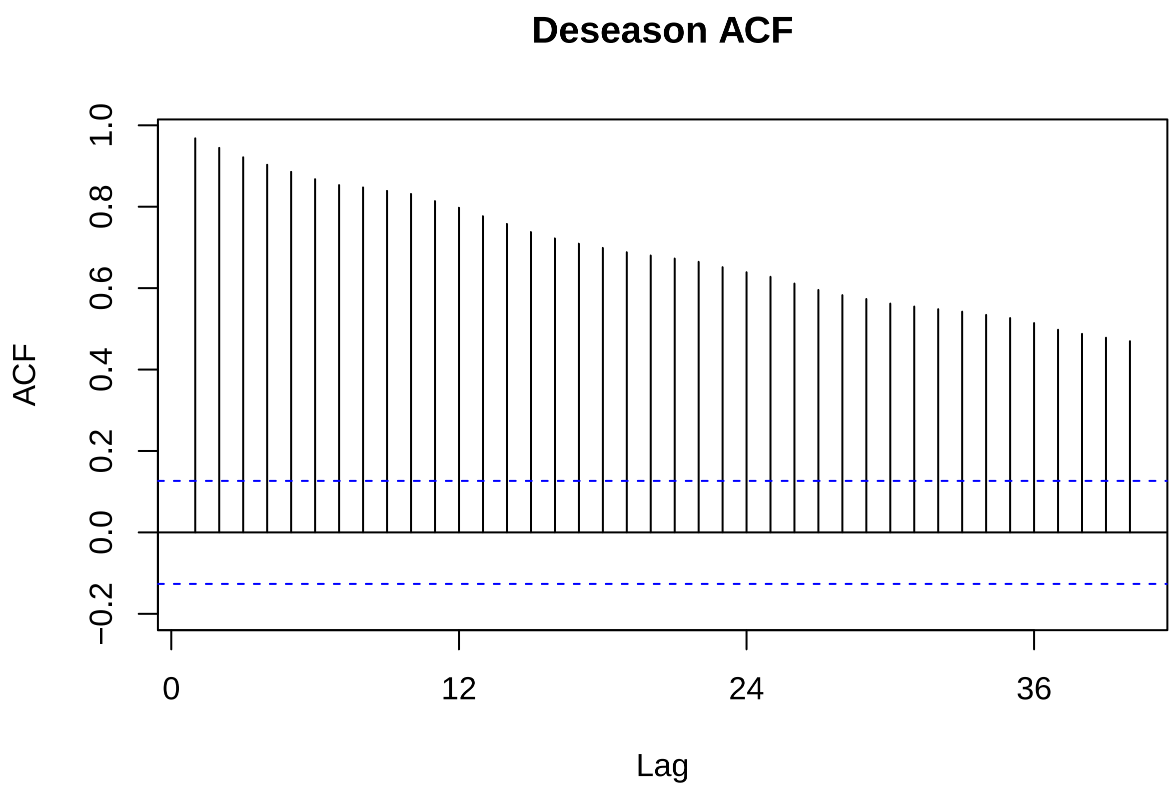
Using the *decompose*() or *stl*() and the *seasadj*() functions create a series without the seasonal component, i.e., a deseasonalized natural gas series. Plot the deseasonalized series over time and corresponding ACF and PACF. Compare with the plots obtained in Q1.

Compared to the plots in Q1, the trend now appears to be more negative than positive. Besides that, it is apparent that the seasonality in the TS, ACF, and PACF are all gone.

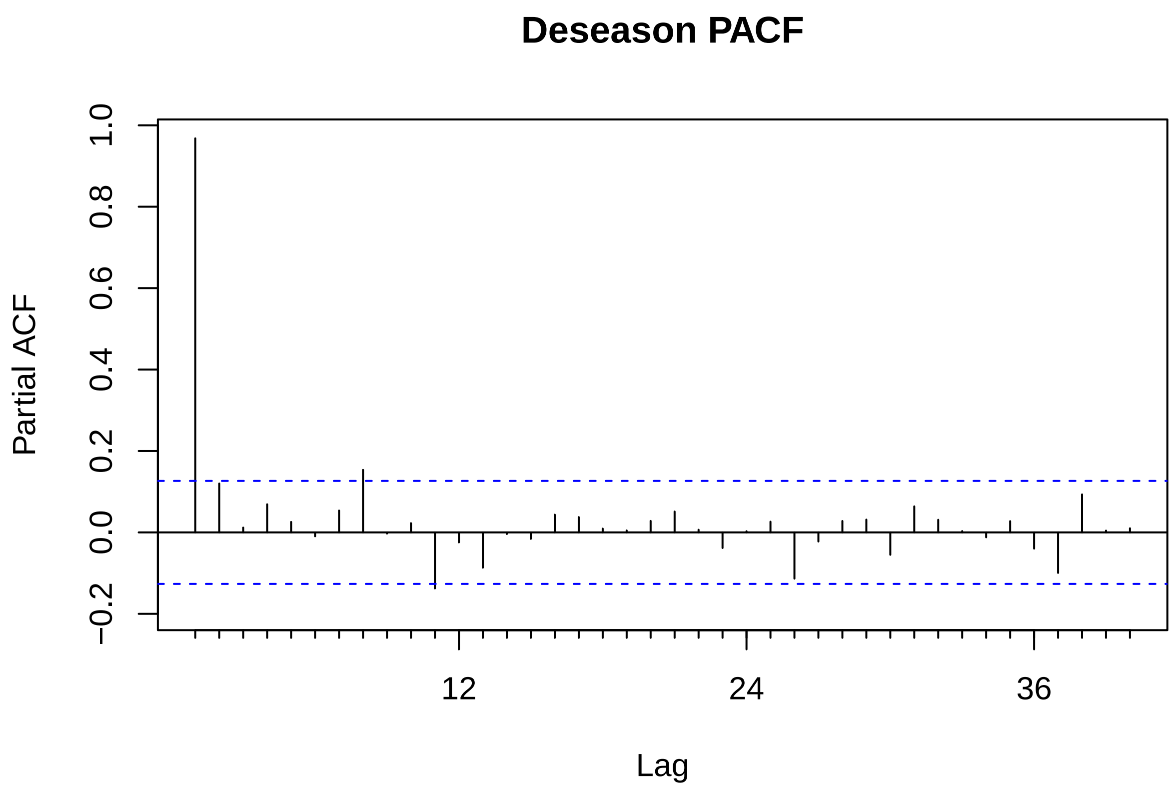
```
decompose_nat_gas=decompose(ts_nat_gas,"multiplicative")
deseasonal_nat_gas <- seasadj(decompose_nat_gas)
plot(deseasonal_nat_gas,col="blue",ylab="Nat Gas",main="Seasonal Detrend")
```




```r
Acf(deseasonal_nat_gas,lag.max=40,main=paste("Deseason ACF"))
```




```
Pacf(deseasonal_nat_gas,lag.max=40,main=paste("Deseason 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.

For the ADF test, since the p-value is less than 0.05, we reject the null hypothesis that the data has a trend. However, for the Mann-Kendall test, since the p-value is less than 0.05, we reject the null hypothesis that the data is stationary, which would mean that there is a trend. This is obviously odd, but since there appeared to be an obvious trend in the deseasoned series, I am going to trust the Mann-Kendall test that there is some trend to the series.

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

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

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

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

```
##
##  Augmented Dickey-Fuller Test
```

```
##
## data:  deseasonal_nat_gas
## Dickey-Fuller = -4.4282, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```r
print("Results of Mann Kendall on Deseasoned series")
```

```
## [1] "Results of Mann Kendall on Deseasoned series"
```

```r
print(summary(MannKendall(deseasonal_nat_gas)))
```

```
## Score =  -24106 , Var(Score) = 1545533
## denominator =  28680
## tau = -0.841, 2-sided pvalue =< 2.22e-16
## NULL
```

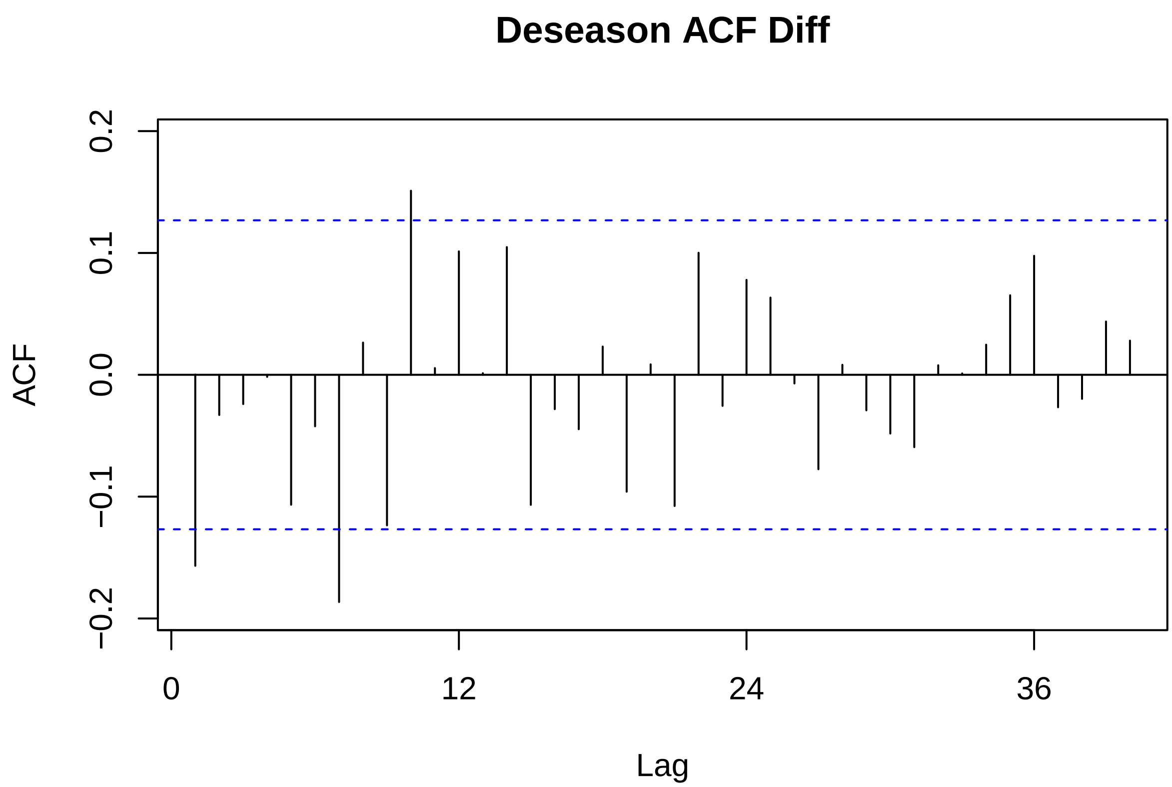
Q4

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. Since there did appear to be a trend, I wanted to difference the series. To do this I decided to set d = 1. I then plotted the new differened and deseasoned ACF and PACF plots. Looking at the ACF plot, it appears that p = 1 and looking at the PACF plot, it appears that q = 0.

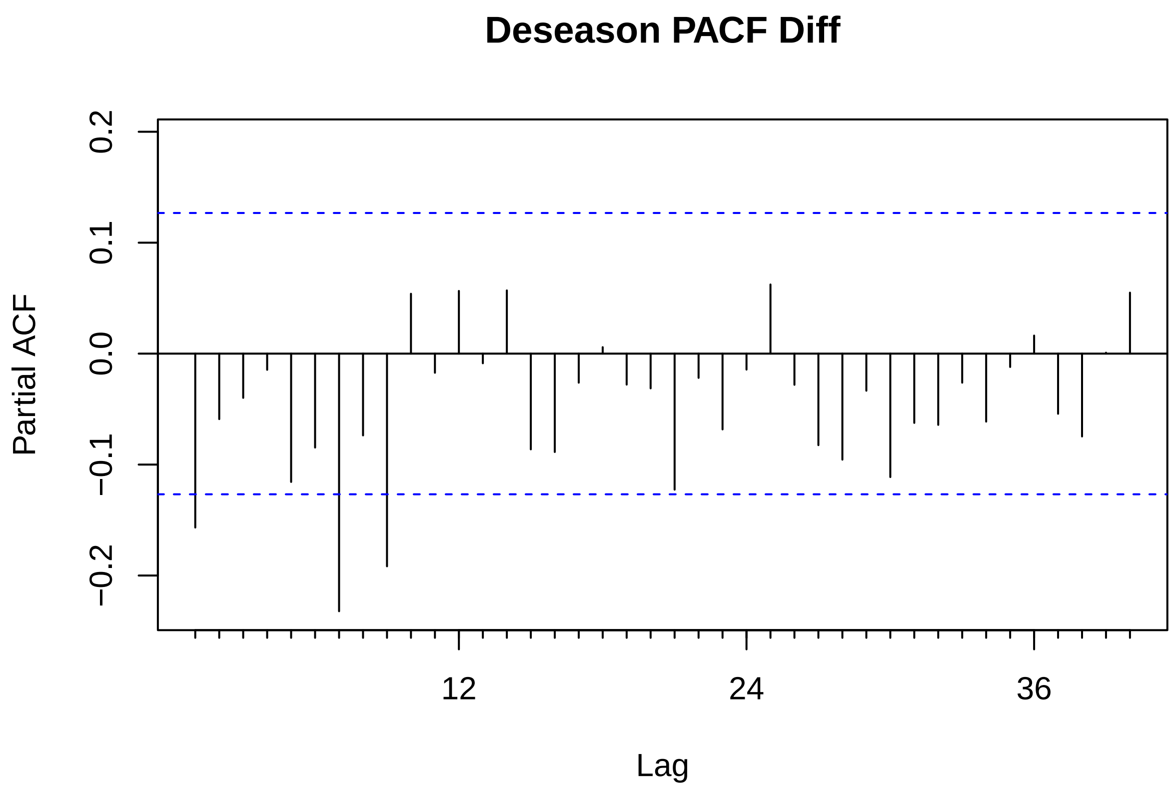
```r
n_diff <- ndiffs(deseasonal_nat_gas)
cat("Number of differencing needed: ",n_diff)
```

```
## Number of differencing needed:  1
```

```r
deseasonal_nat_gas_diff <- diff(deseasonal_nat_gas,differences=1)
Acf(deseasonal_nat_gas_diff,lag.max=40,main=paste("Deseason ACF Diff"))
```




```
Pacf(deseasonal_nat_gas_diff,lag.max=40,main=paste("Deseason PACF Diff"))
```



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. Since d = 1, I am setting $include.mean = FALSE$ and since $include.drift = TRUE$ might lead to better fits, I included it as well.

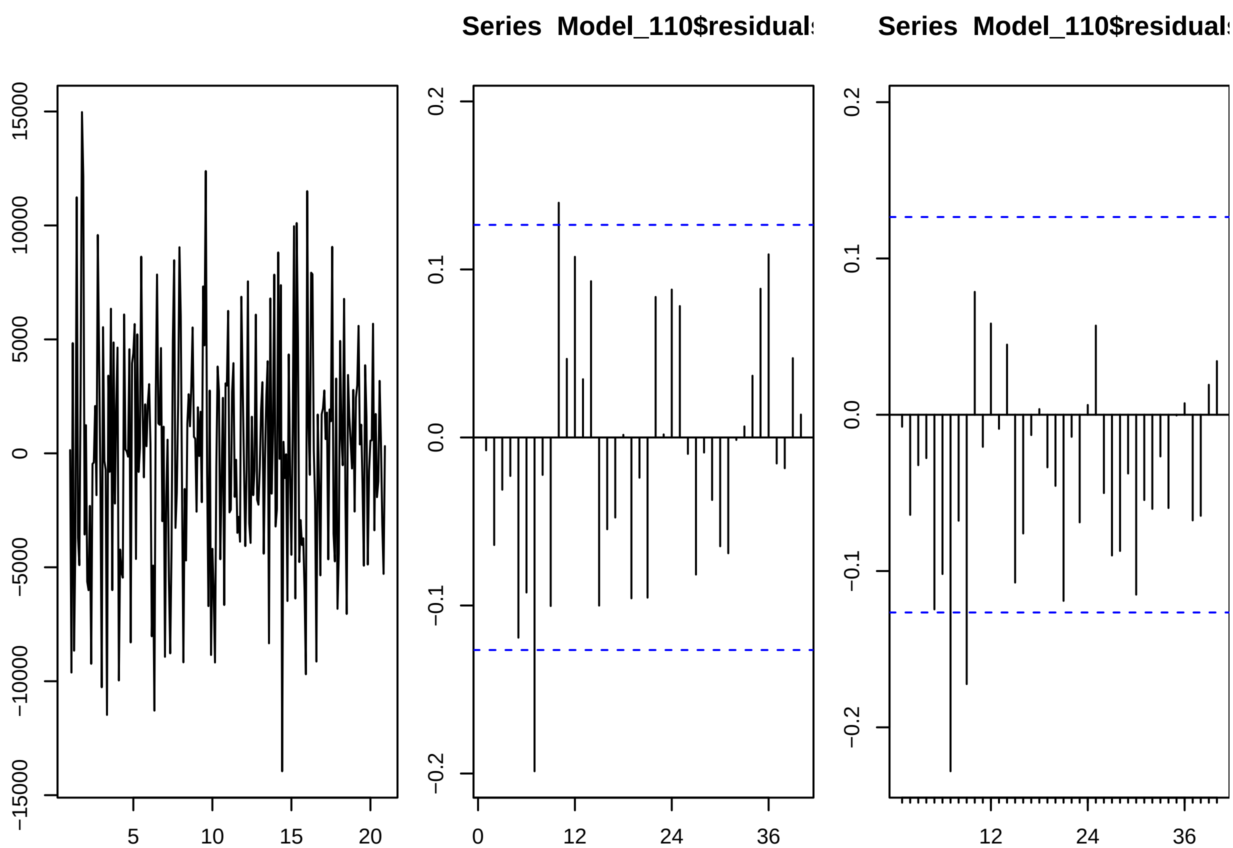
```
Model_110 <- Arima(deseasonal_nat_gas,order=c(1,1,0),include.mean = FALSE,include.drift=TRUE)
print(Model_110)
```

```
## Series: deseasonal_nat_gas
## ARIMA(1,1,0) with drift
##
## Coefficients:
##           ar1      drift
##       -0.1586  -392.0379
## s.e.   0.0642   278.9583
##
## sigma^2 = 25146651:  log likelihood = -2374.44
## AIC=4754.89   AICc=4754.99   BIC=4765.31
```

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? The three series do look like white noise series since there appears to be no real trend or seasonality to the TS, ACF, nor PACF series. This is promising and means that my model could be a good fit.

```
compare_aic <- data.frame(Model_110$aic)
par(mar=c(3,3,3,0));par(mfrow=c(1,3))
ts.plot(Model_110$residuals)
Acf(Model_110$residuals,lag.max=40)
Pacf(Model_110$residuals,lag.max=40)
```



Modeling the original series (with seasonality)

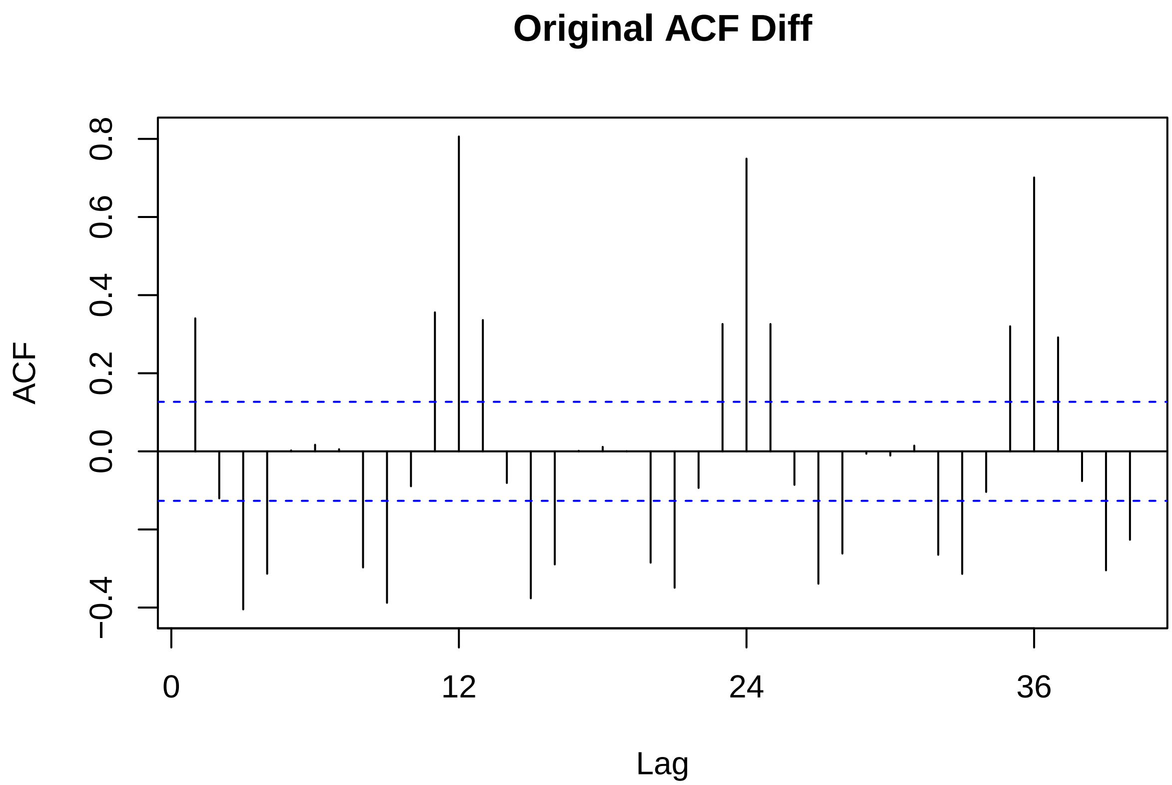
Q7

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.

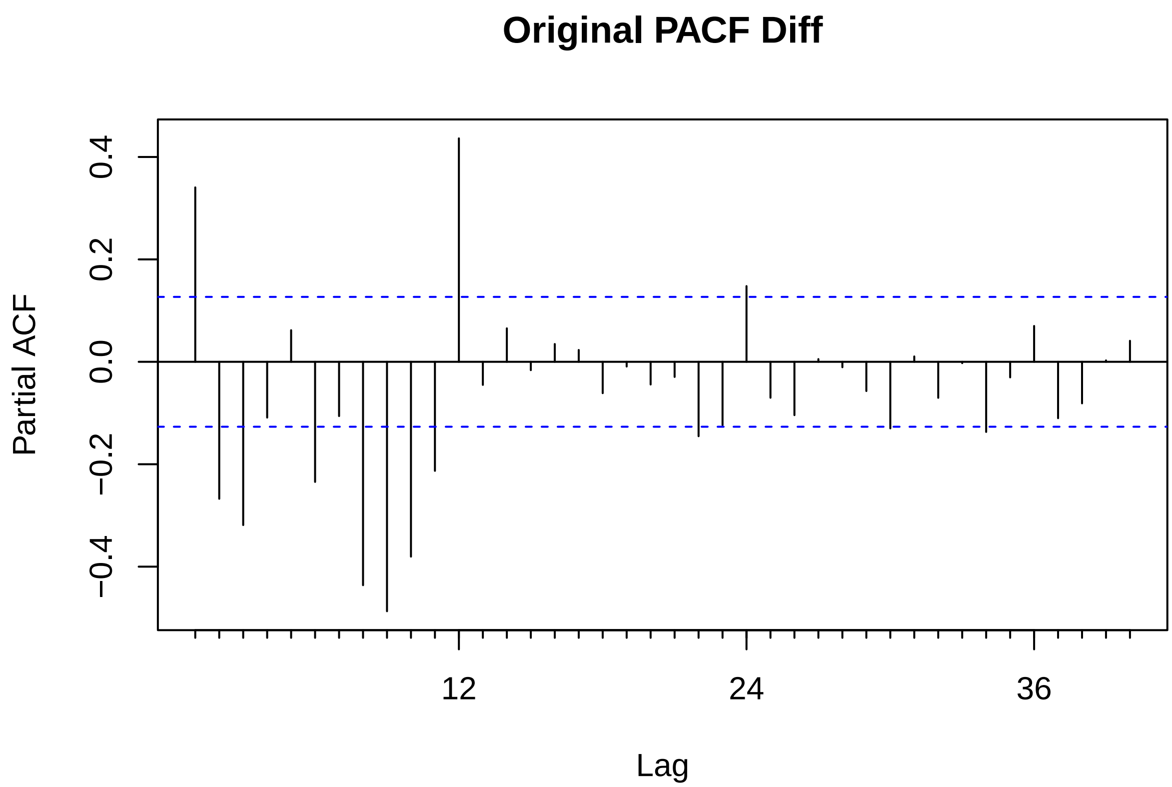
```
n_diff <- ndiffs(ts_nat_gas)
cat("Number of differencing needed: ",n_diff)
```

```
## Number of differencing needed:  1
```

```
ts_nat_gas_diff <- diff(ts_nat_gas,differences=1)
Acf(ts_nat_gas_diff,lag.max=40,main=paste("Original ACF Diff"))
```




```
Pacf(ts_nat_gas_diff,lag.max=40,main=paste("Original PACF Diff"))
```



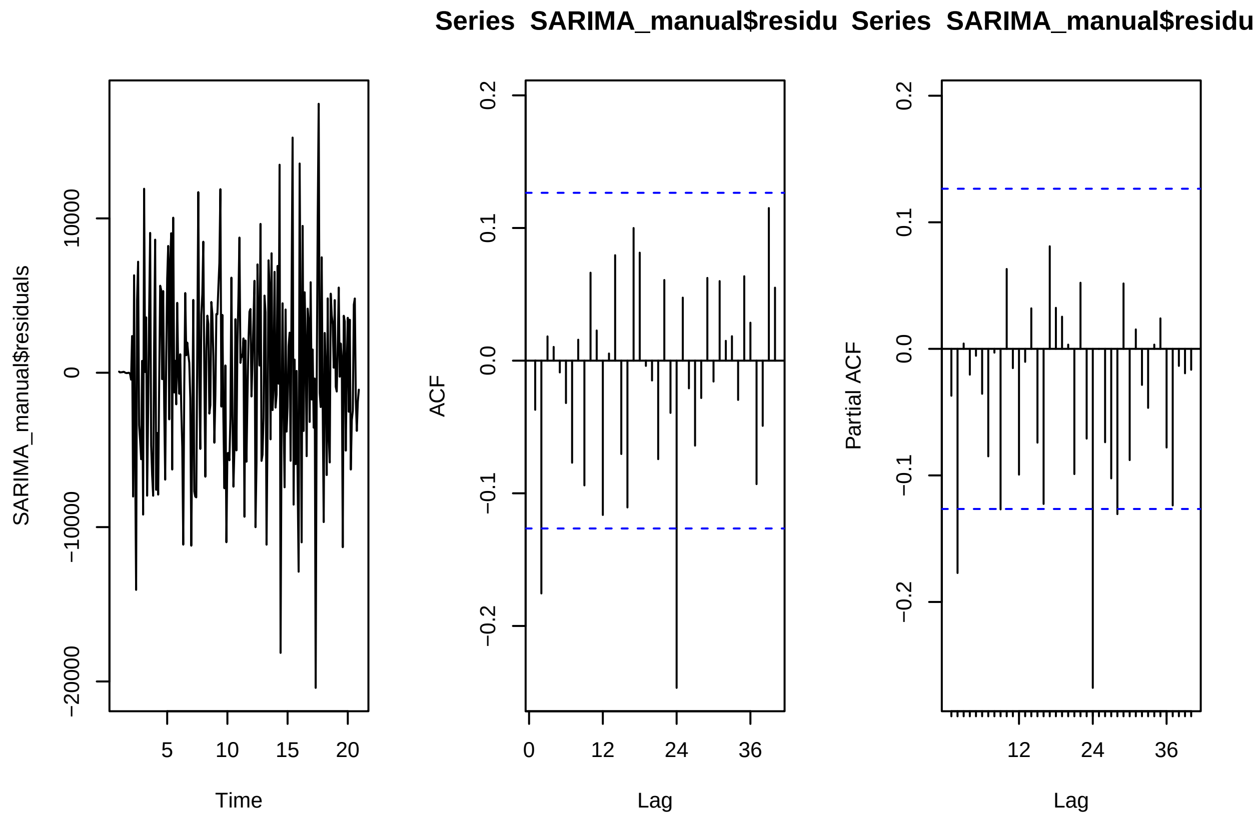

```
SARIMA_manual <- Arima(ts_nat_gas,order=c(1,1,0),seasonal=c(1,1,0),include.mean=FALSE,include.drift=TRUE
```

```
## Warning in Arima(ts_nat_gas, order = c(1, 1, 0), seasonal = c(1, 1, 0), : No
## drift term fitted as the order of difference is 2 or more.
```

```
print(SARIMA_manual)
```

```
## Series: ts_nat_gas
## ARIMA(1,1,0)(1,1,0)[12]
##
## Coefficients:
##           ar1     sar1
##       -0.1897  -0.4659
## s.e.   0.0653   0.0589
##
## sigma^2 = 35371411:  log likelihood = -2295.37
## AIC=4596.74   AICc=4596.85   BIC=4607.01
```

```
par(mfrow=c(1,3))
ts.plot(SARIMA_manual$residuals)
Acf(SARIMA_manual$residuals,lag.max=40)
Pacf(SARIMA_manual$residuals,lag.max=40)
```



Q8

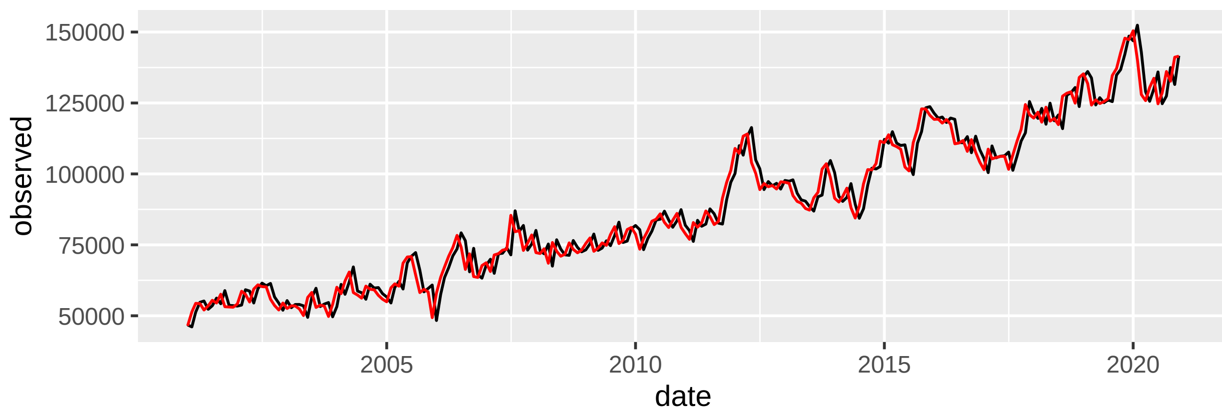
Compare the residual series for Q7 and Q6. Can you tell which ARIMA model is better representing the Natural Gas Series? Is that a fair comparison? Explain your response. The first ARIMA model better fits the series, but that is not a fair comparison since the first ARIMA has already been deseasoned whereas the second ARIMA model still has the seasonality in it.

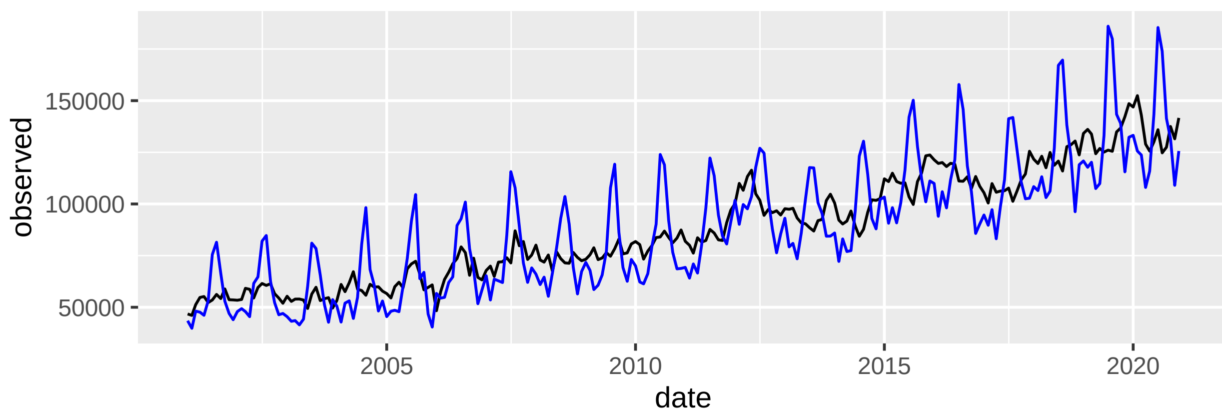
```
df_models <- data.frame(
  date = nat_gas$Date,
  observed = as.numeric(deseasonal_nat_gas),
  ARIMA_110 = as.numeric(Model_110$fitted),
  SARIMA_manual = as.numeric(SARIMA_manual$fitted)
)

Plot1 <-
ggplot(df_models) +
  geom_line(aes(x=date,y=observed),color="black") +
  geom_line(aes(x=date,y=ARIMA_110),color="red")

Plot2 <-
ggplot(df_models) +
  geom_line(aes(x=date,y=observed),color="black") +
  geom_line(aes(x=date,y=SARIMA_manual),color="blue")

cowplot::plot_grid(Plot1,Plot2,nrow=2)
```





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? The best order for the deseasoned data is a ARIMA(1,1,0) with drift which is exactly what I got in Q4.

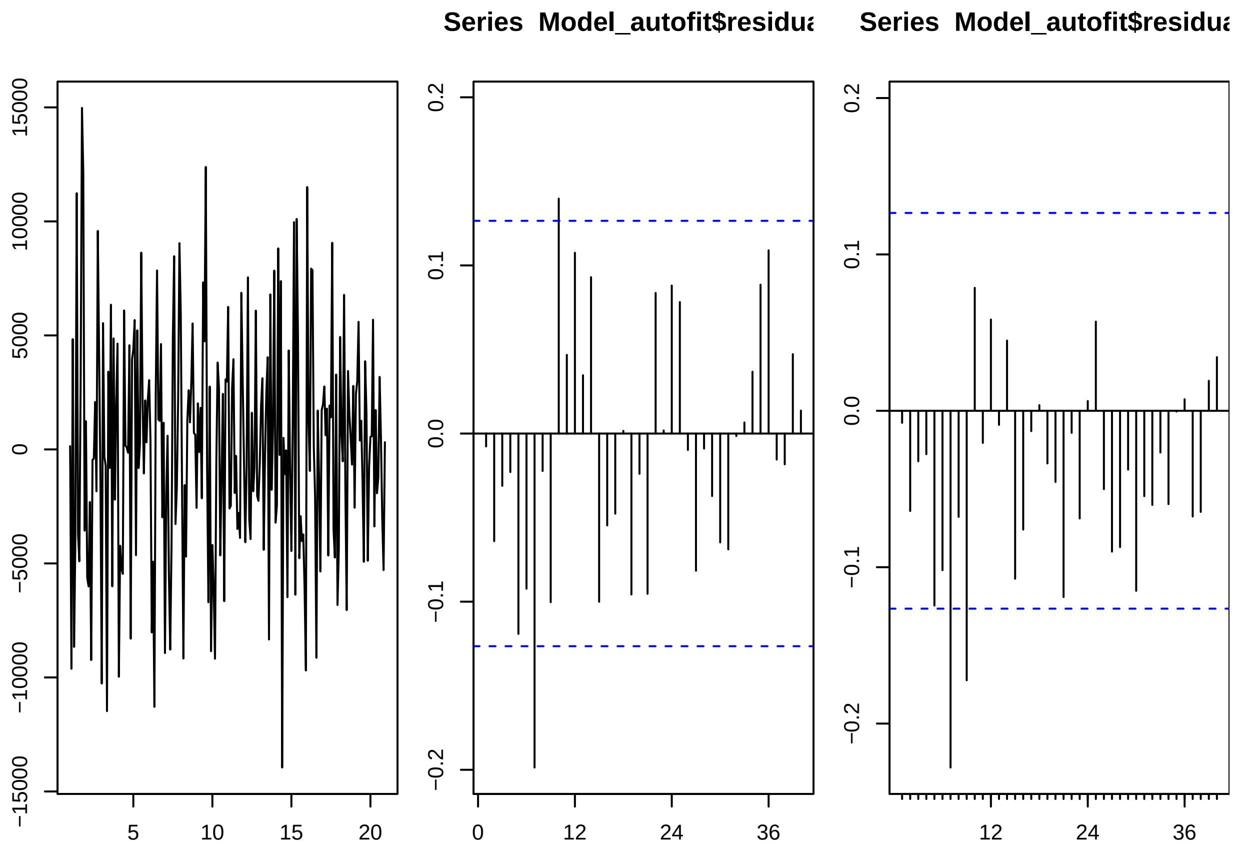
```
Model_autofit <- auto.arima(deseasonal_nat_gas,max.D=0,max.P = 0,max.Q=0)
print(Model_autofit)
```

```
## Series: deseasonal_nat_gas
## ARIMA(1,1,0) with drift
##
## Coefficients:
##           ar1      drift
##       -0.1586  -392.0379
## s.e.   0.0642   278.9583
##
```

```
## sigma^2 = 25146651:  log likelihood = -2374.44
## AIC=4754.89   AICc=4754.99   BIC=4765.31
```

```
compare_aic <- cbind(compare_aic,Model_autofit$aic)

par(mar=c(3,3,3,0));par(mfrow=c(1,3))
ts.plot(Model_autofit$residuals)
Acf(Model_autofit$residuals,lag.max=40)
Pacf(Model_autofit$residuals,lag.max=40)
```




```
print(compare_aic)
```

```
##   Model_110.aic Model_autofit$aic
## 1      4754.885          4754.885
```

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? The model chose the fit, ARIMA(2,0,1)(2,1,2) with drift. This was unfortunately not what I had specified in Q7.

```
SARIMA_autofit <- auto.arima(ts_nat_gas)
print(SARIMA_autofit)
```

```
## Series: ts_nat_gas
## 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  -357.8436
## s.e.  0.4163   0.3180   0.3992   1.3206   0.1014   1.3199  0.9204    44.1285
##
## sigma^2 = 27958165:  log likelihood = -2278.46
## AIC=4574.91   AICc=4575.74   BIC=4605.77
```

```
par(mfrow=c(1,3))
ts.plot(SARIMA_autofit$residuals)
Acf(SARIMA_autofit$residuals,lag.max=40)
Pacf(SARIMA_autofit$residuals,lag.max=40)
```