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

Assignment 7 - Due date 03/07/24

Julia Kagiliery

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(lubridate)
```

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

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

```r
library(ggplot2)
library(forecast)
```

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

```r
library(Kendall)
library(tseries)
library(outliers)
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr   1.1.3     v stringr 1.5.0
## v forcats 1.0.0     v tibble  3.2.1
## v purrr   1.0.2     v tidyr   1.3.0
## v readr   2.1.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

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**.

```
#Importing time series data from text file
NaturalGas <- read.csv(file="~/Julia_Kagiliery_TSA_Sp24/Data/Net_generation_United_States_all_sectors_m

#Inspect data
head(NaturalGas)
```

```
##      Month all.fuels..utility.scale..thousand.megawatthours
## 1 Dec 2020                                          344970.4
## 2 Nov 2020                                          302701.8
## 3 Oct 2020                                          313910.0
## 4 Sep 2020                                          334270.1
## 5 Aug 2020                                          399504.2
## 6 Jul 2020                                          414242.5
##   coal.thousand.megawatthours natural.gas.thousand.megawatthours
## 1                    78700.33                           125703.7
## 2                    61332.26                           109037.2
## 3                    59894.57                           131658.2
## 4                    68448.00                           141452.7
## 5                    91252.48                           173926.6
## 6                    89831.36                           185444.8
##   nuclear.thousand.megawatthours
## 1                       69870.98
## 2                       61759.98
## 3                       59362.46
## 4                       65727.32
## 5                       68982.19
## 6                       69385.44
##   conventional.hydroelectric.thousand.megawatthours
## 1                                          23086.37
## 2                                          21831.88
## 3                                          18320.72
## 4                                          19161.97
## 5                                          24081.57
## 6                                          27675.94
```

```
nvar <- ncol(NaturalGas) - 1
nobs <- nrow(NaturalGas)

#Preparing the data - create date object and rename columns
NaturalGas_processed <-
  NaturalGas |>
```

```
 mutate( Month = my(Month) ) |>
  rename(Gas = natural.gas.thousand.megawatthours) |>
    arrange( Month )

NaturalGas_processed <- NaturalGas_processed[,c(1,4)]

head(NaturalGas_processed)
```

```
##        Month      Gas
## 1 2001-01-01 42388.66
## 2 2001-02-01 37966.93
## 3 2001-03-01 44364.41
## 4 2001-04-01 45842.75
## 5 2001-05-01 50934.21
## 6 2001-06-01 57603.15
```

```
summary(NaturalGas_processed)
```

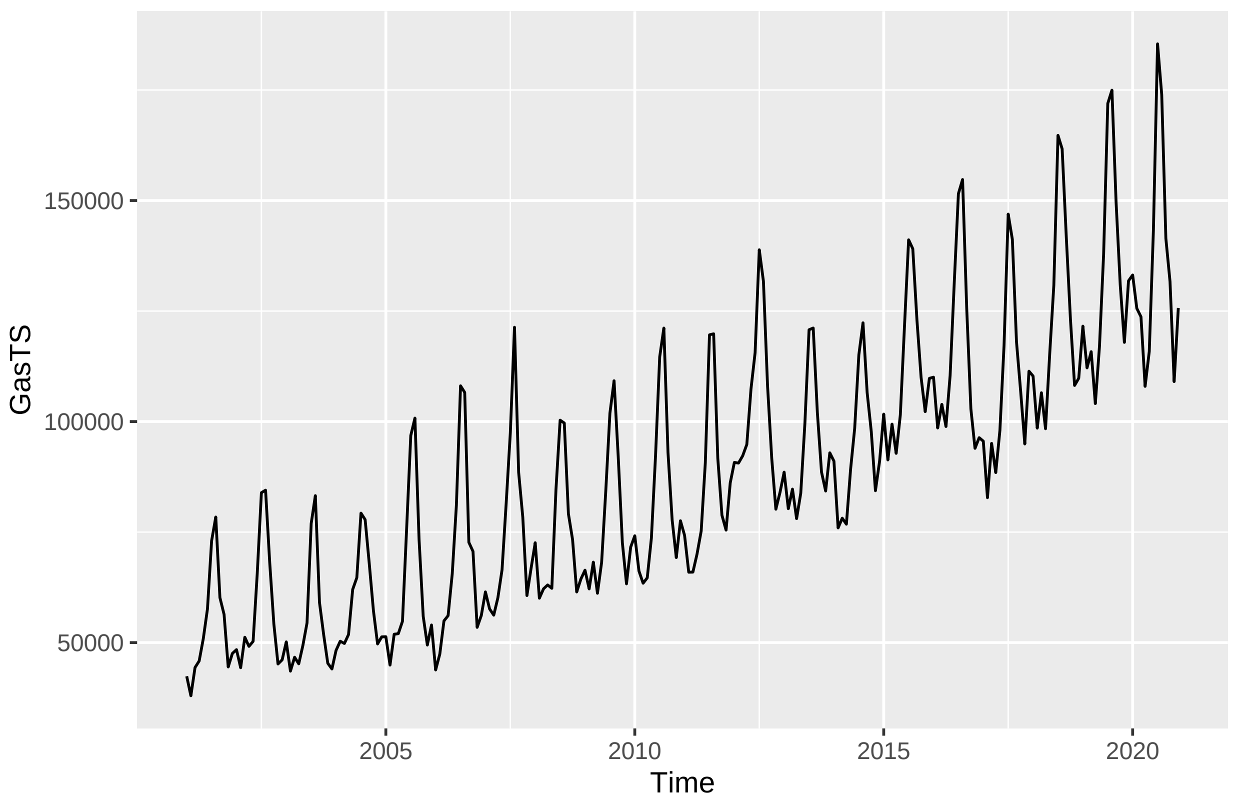
```
##      Month                 Gas
##  Min.   :2001-01-01   Min.   : 37967
##  1st Qu.:2005-12-24   1st Qu.: 62245
##  Median :2010-12-16   Median : 84415
##  Mean   :2010-12-16   Mean   : 88028
##  3rd Qu.:2015-12-08   3rd Qu.:108385
##  Max.   :2020-12-01   Max.   :185445
```

Q1

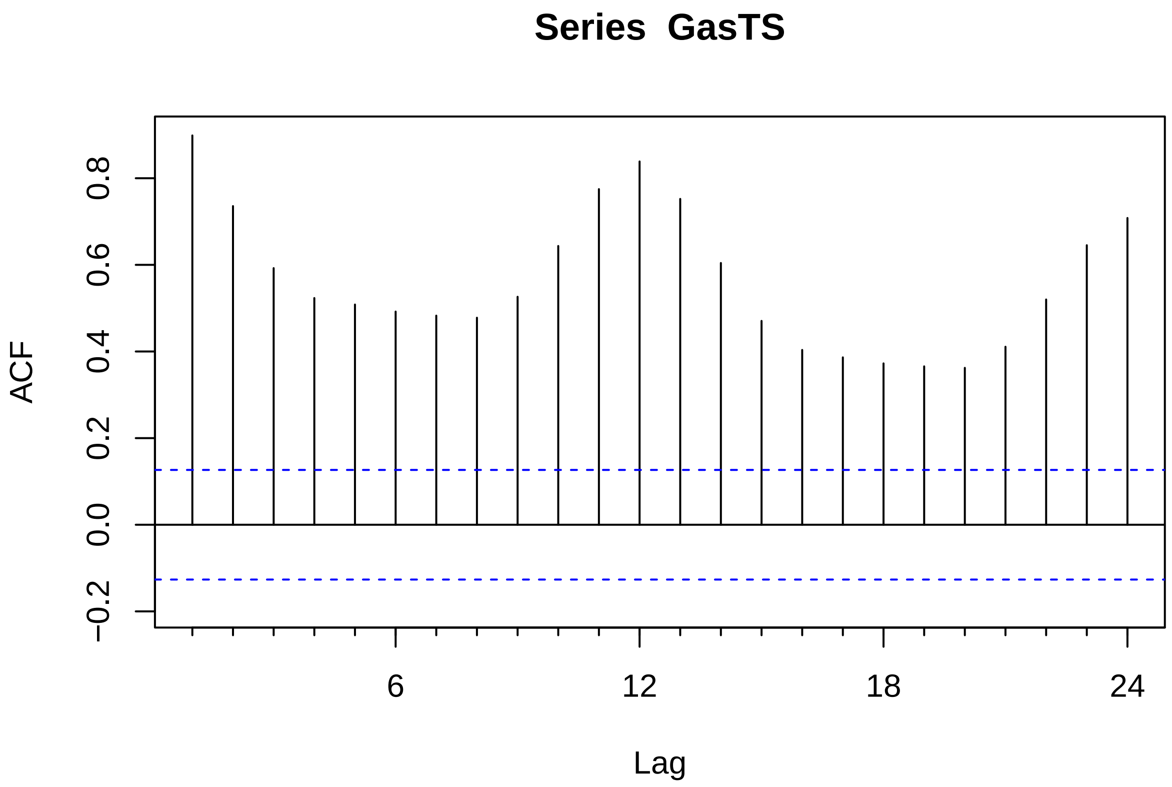
Import the csv file and create a time series object for natural gas. Make you sure you specify the **start=** and **frequency=** arguments. Plot the time series over time, ACF and PACF.

```
GasTS <- ts(NaturalGas_processed[,2], start=c(year(NaturalGas_processed$Month[1]),month(NaturalGas_proce
                           frequency=12)
```

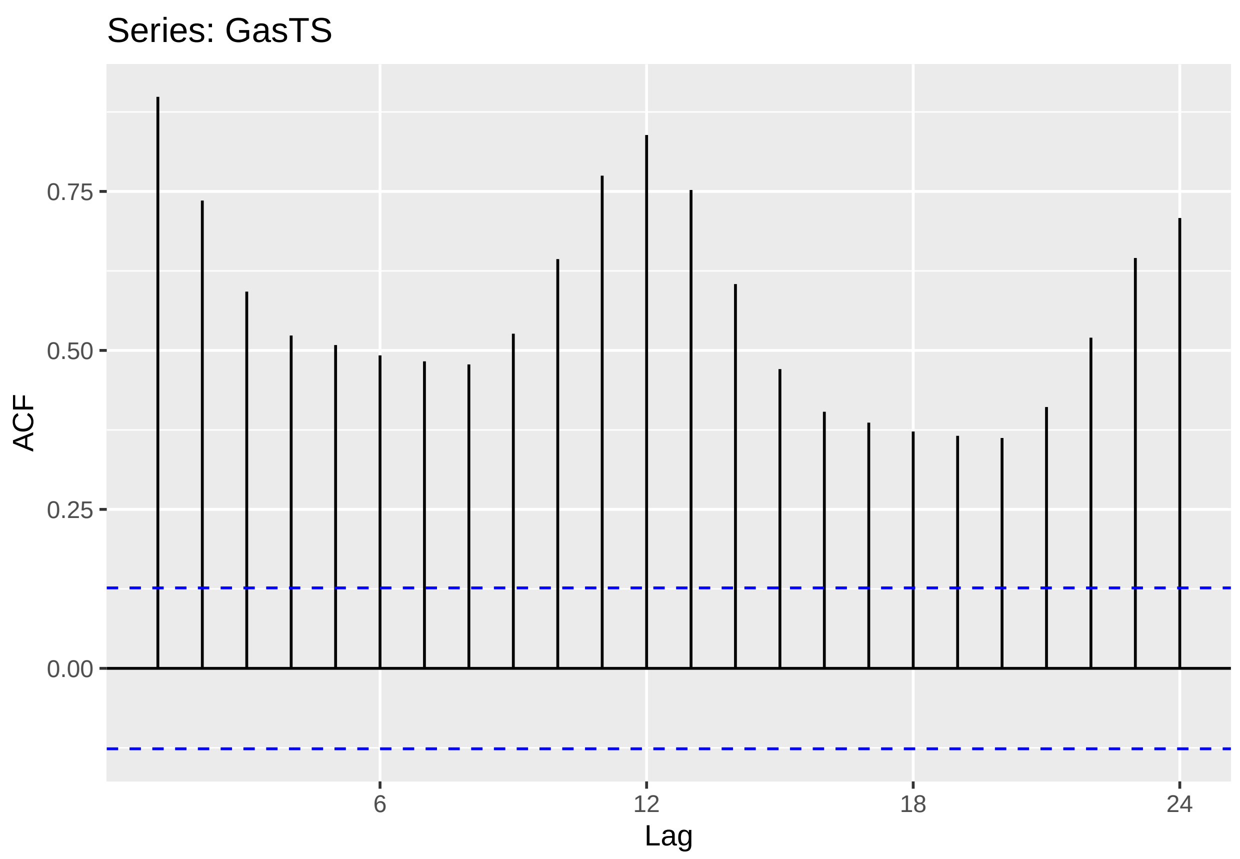
```
autoplot(GasTS)
```



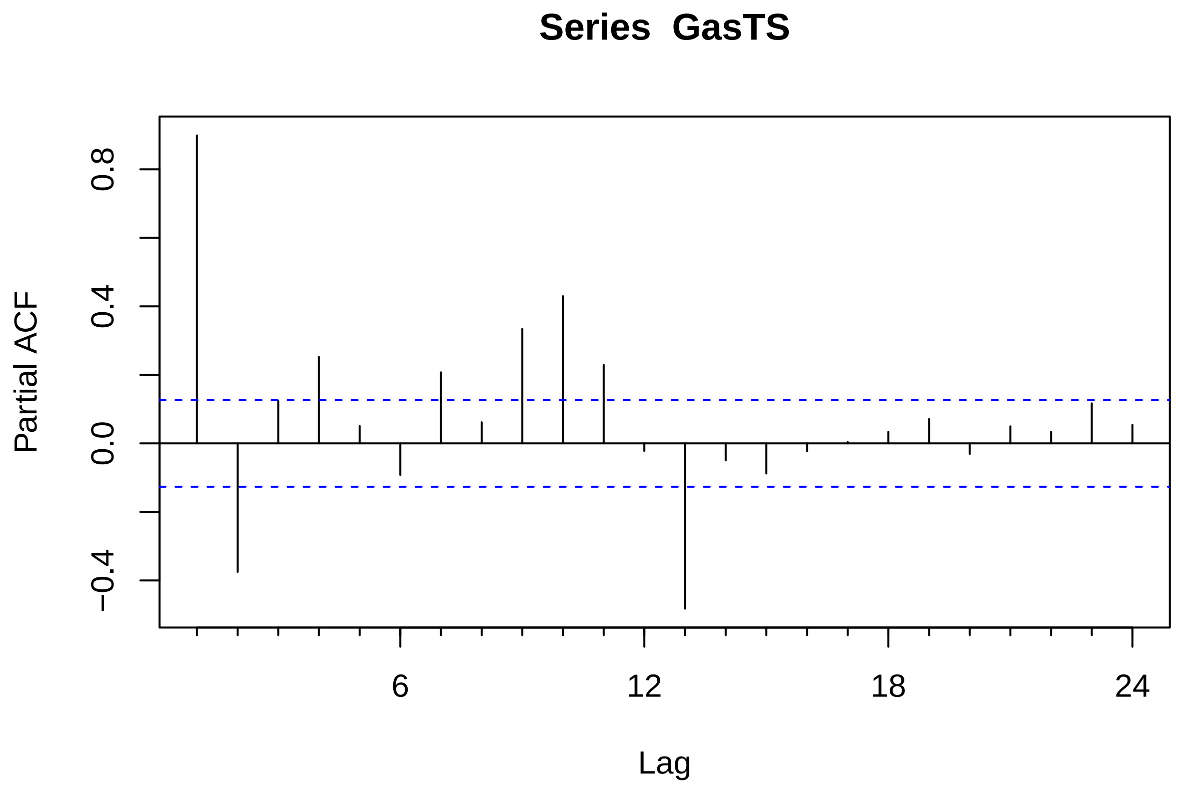
```r
autoplot(Acf(GasTS))
```

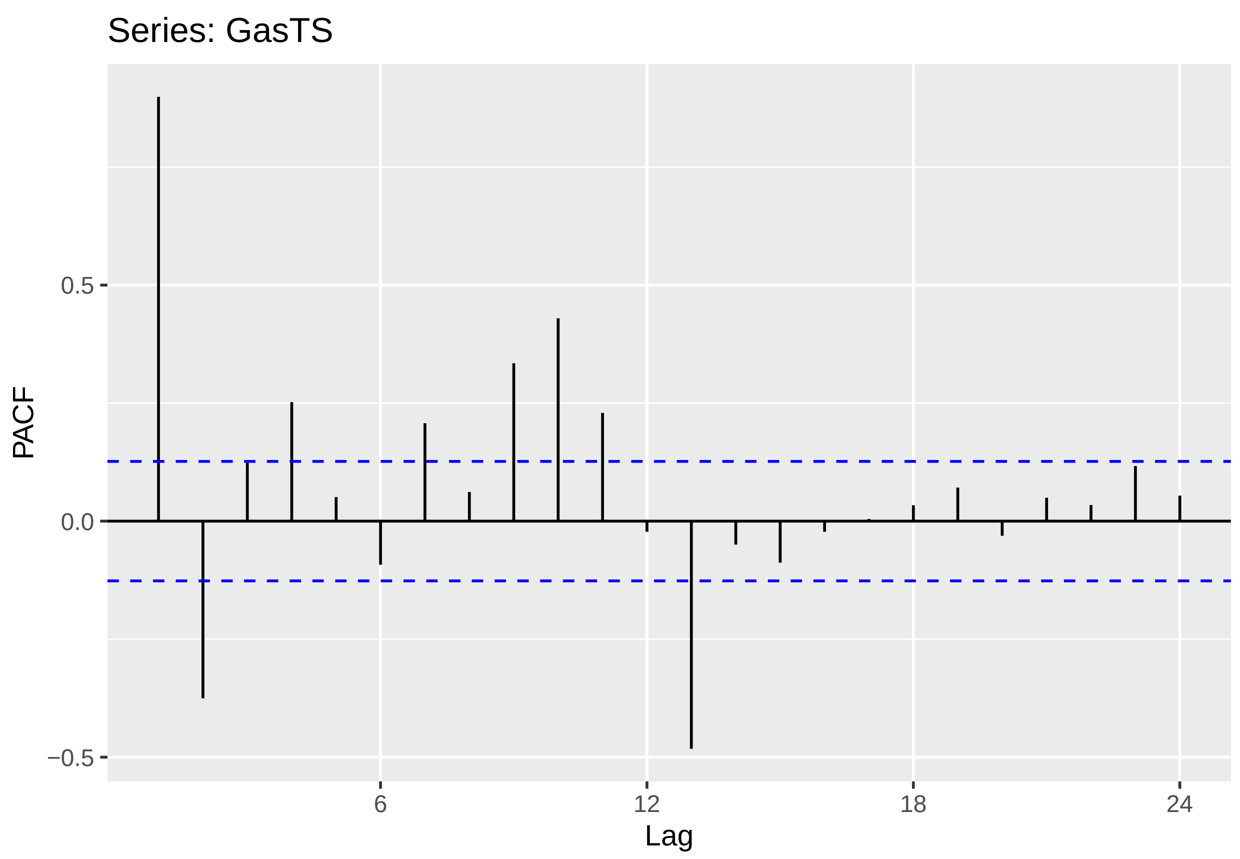


Series GasTS
ACF
0.8
0.6
0.4
0.2
0.0
−0.2
6
12
18
24
Lag



```r
autoplot(Pacf(GasTS))
```

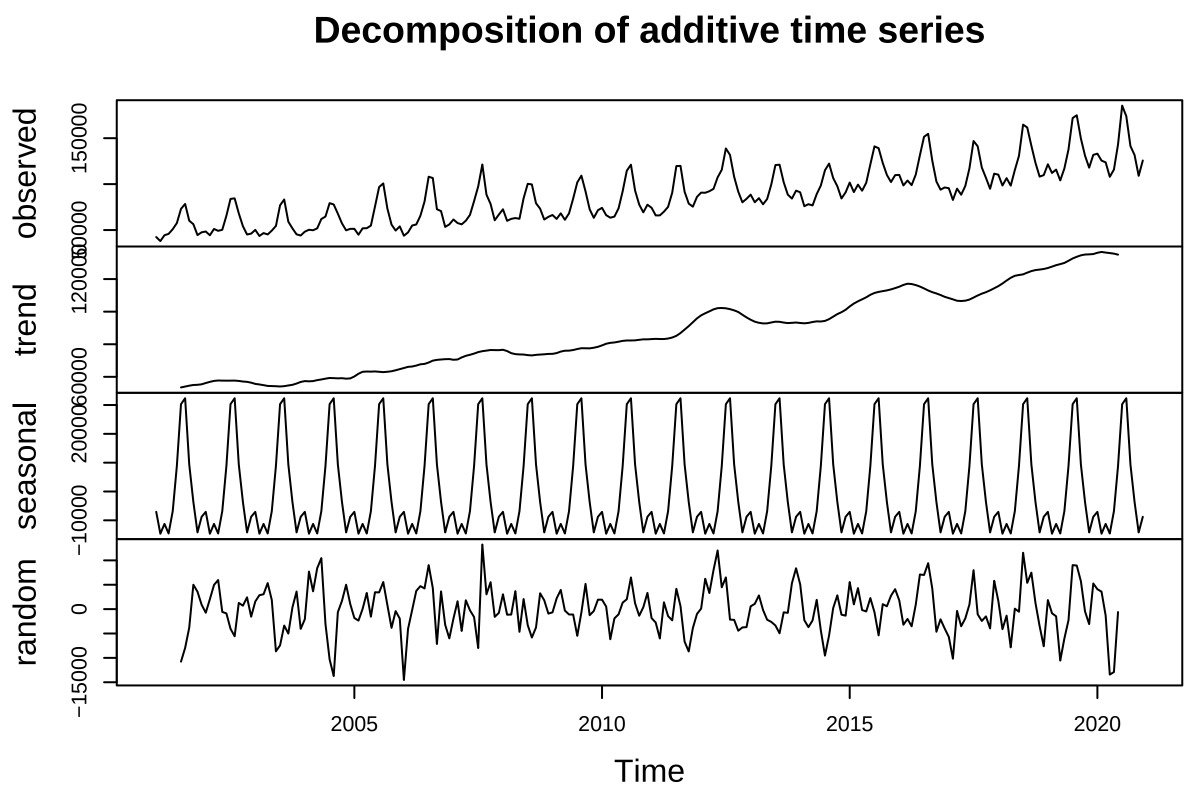

Series GasTS
Partial ACF
0.8
0.4
0.0
−0.4
6
12
18
24
Lag



Q2

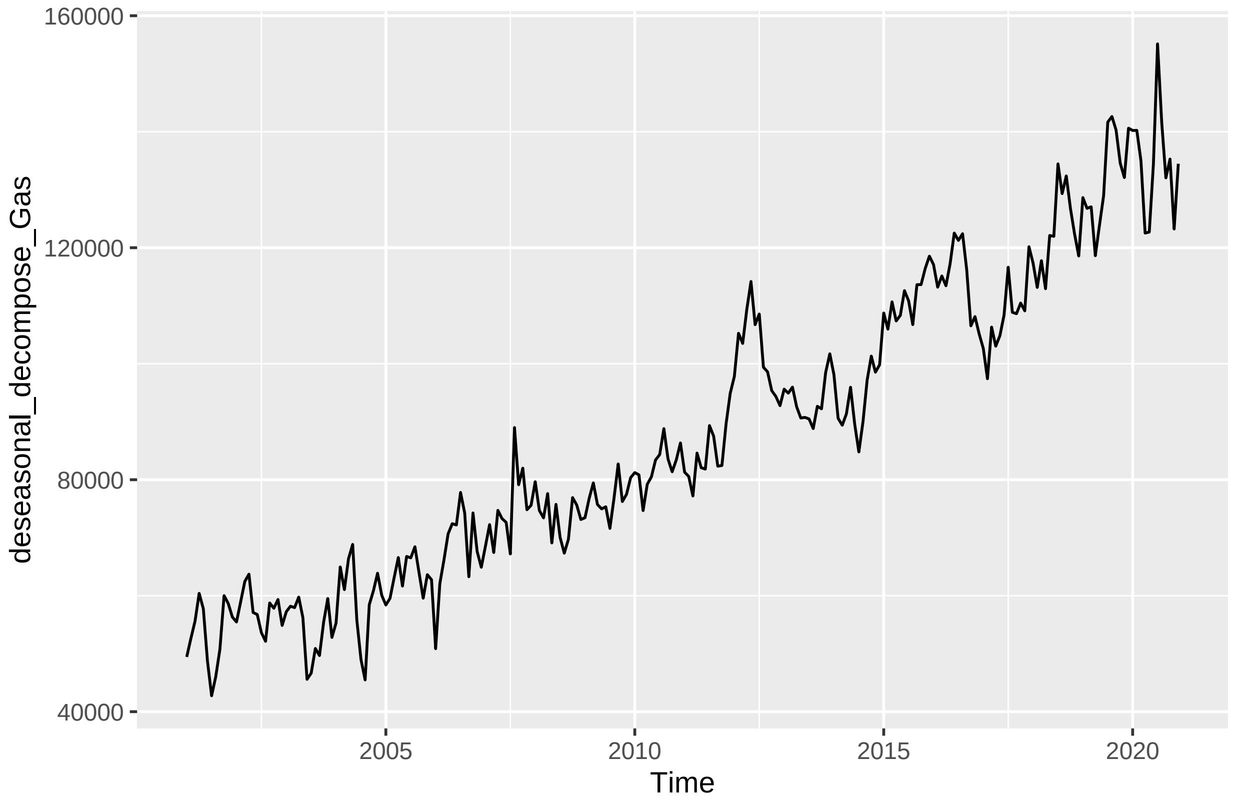
Using the *decompose*() or *stl*() and the *seasadj*() functions create a series without the seasonal component, i.e., a deseasonalized natural gas series. Plot the deseasonalized series over time and corresponding ACF and PACF. Compare with the plots obtained in Q1.

```
#Using R decompose function
decompose_Gas <- decompose(GasTS,"additive")
plot(decompose_Gas)
```

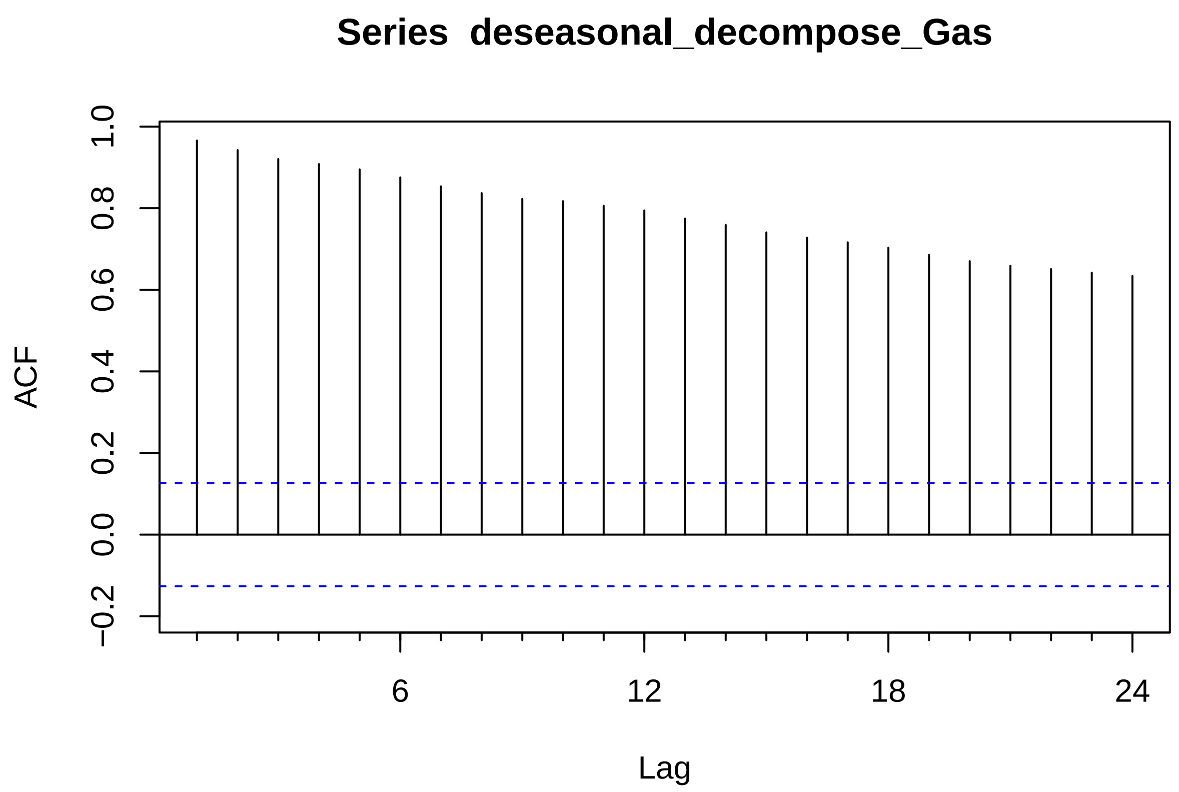


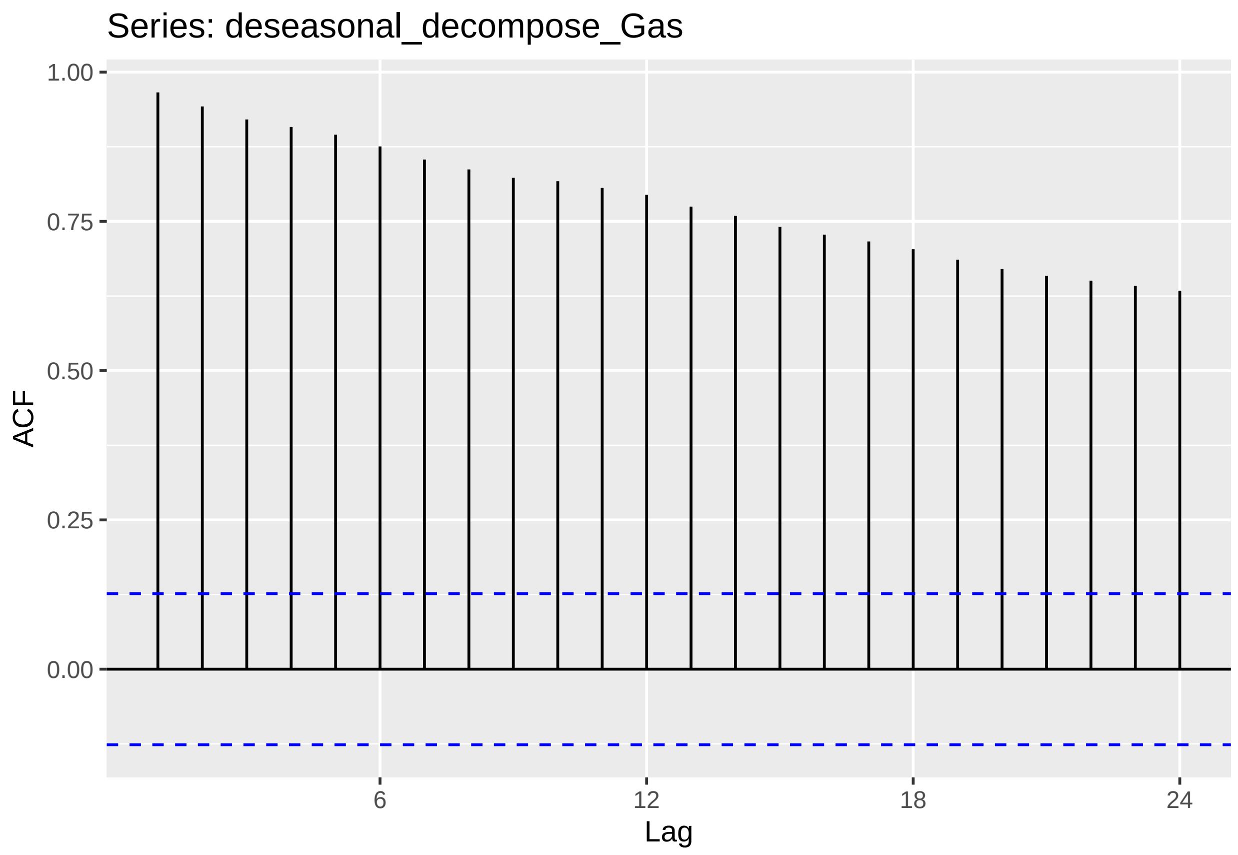
```
#The ACF plot show a slow decay which is a sign of non-stationarity.

#Creating non-seasonal residential price time series because some models can't handle seasonality
deseasonal_decompose_Gas <- seasadj(decompose_Gas)
```

```
autoplot(deseasonal_decompose_Gas)
```

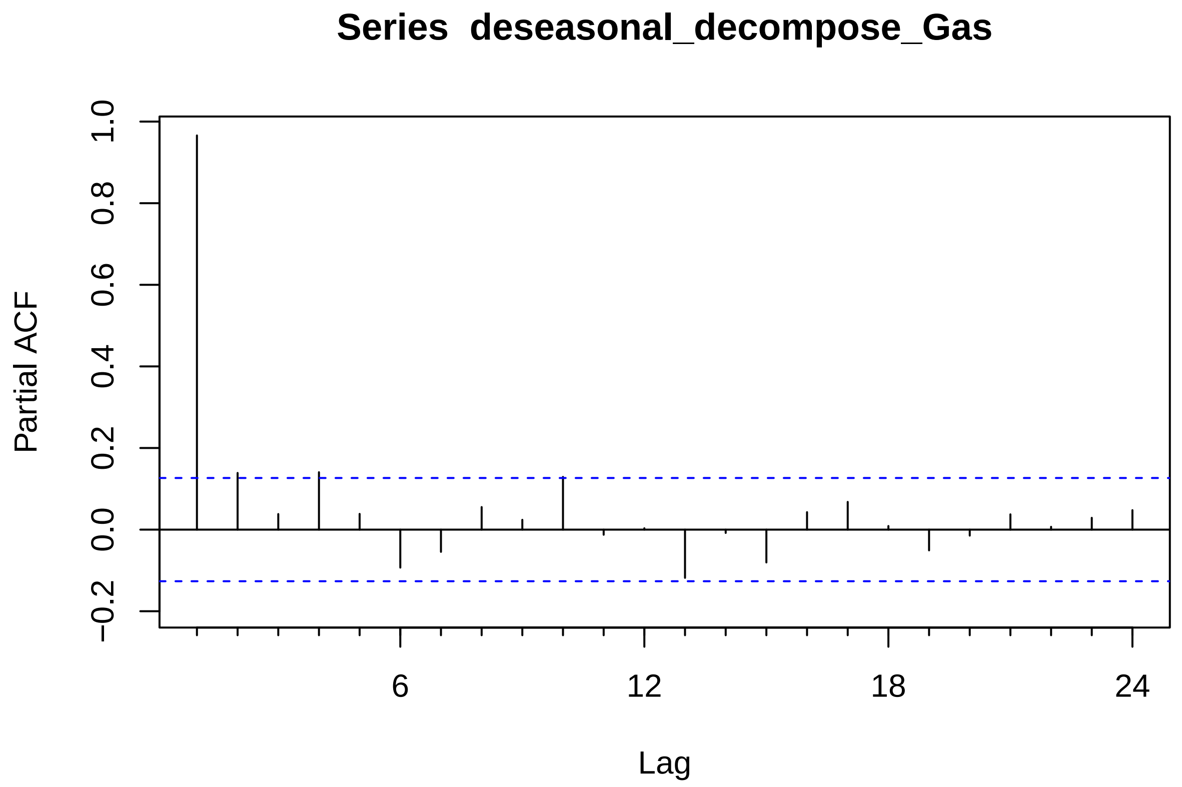


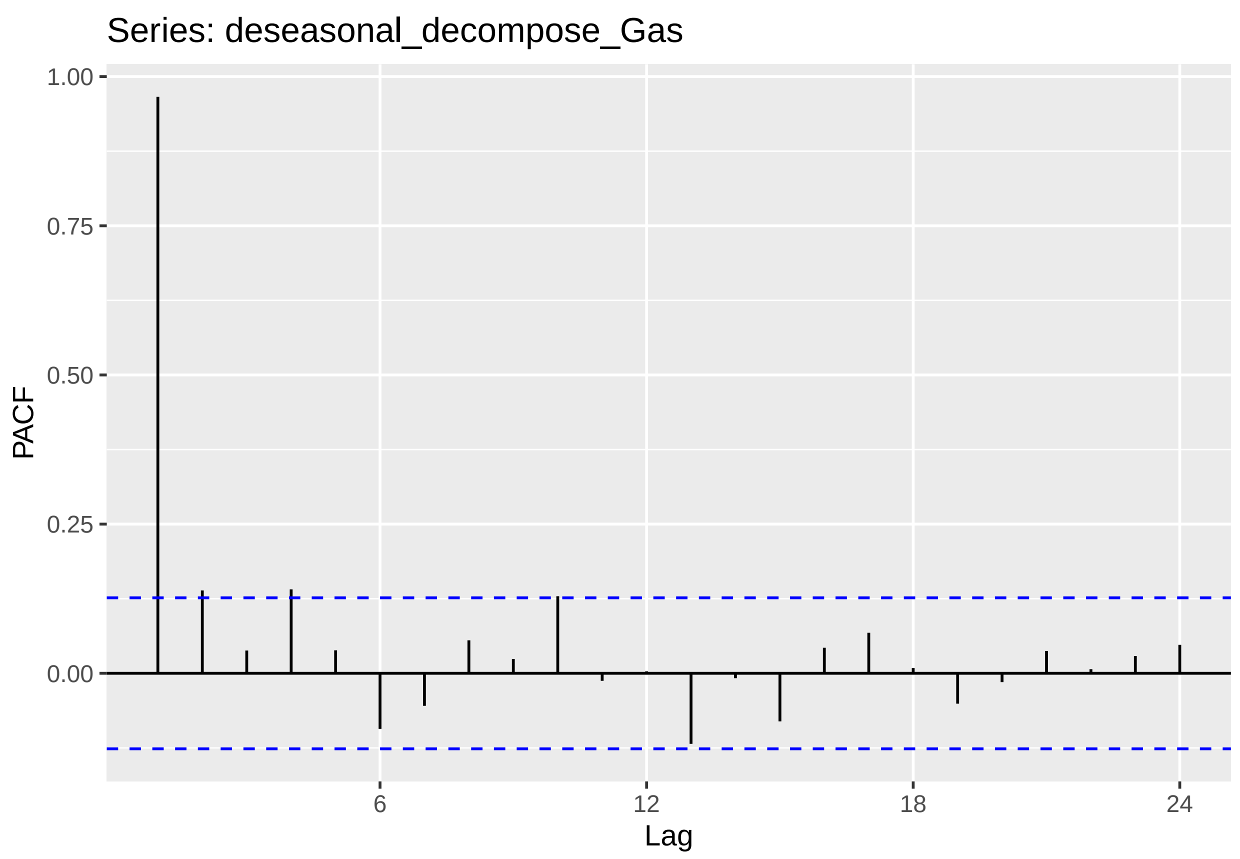
```r
autoplot(Acf(deseasonal_decompose_Gas))
```


Series deseasonal_decompose_Gas
ACF
Lag
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
6
12
18
24



```r
autoplot(Pacf(deseasonal_decompose_Gas))
```


Series deseasonal_decompose_Gas
Partial ACF
−0.2
0.0
0.2
0.4
0.6
0.8
1.0
6
12
18
24
Lag



The deseasoned decomposed object shows less ovbious seasonality but still seems to have some over all trend which makes me think I may need to diffrence the data. The ACF confimrs this as it is decreasing across the lags but significant. The Pacf seems to suggest the time series has an autoregressive coefficent of 1.

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(deseasonal_decompose_Gas))
```

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

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

Here we reject the null hypothesis and say that the data is stationary. This does not see, to make sense in the conetxt of visual inspection in which the timeseries seems to trend upward. This test assume parametric data and linear trends so it could just be a poor test to use in this case.

```r
print(MannKendall(deseasonal_decompose_Gas))
```

```
## tau = 0.843, 2-sided pvalue =< 2.22e-16
```

Here we see a positive 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.

d = 1 because there is a trend according to the Mann Kendall test and visual inspecition p = 1 beacuse the Pacf spikes at 1 lag q = 1 beacuse my Acf is significant for many lags, so my moving average will be imprtant as the data obvbiously has some sort of retrospecitive component at many lags

```
Arima(deseasonal_decompose_Gas, c(1,1,1))
```

```
## Series: deseasonal_decompose_Gas
## ARIMA(1,1,1)
##
## Coefficients:
##          ar1      ma1
##       0.5720  -0.8167
## s.e.  0.1024   0.0696
##
## sigma^2 = 28492791:  log likelihood = -2389.48
## AIC=4784.95   AICc=4785.06   BIC=4795.38
```

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.

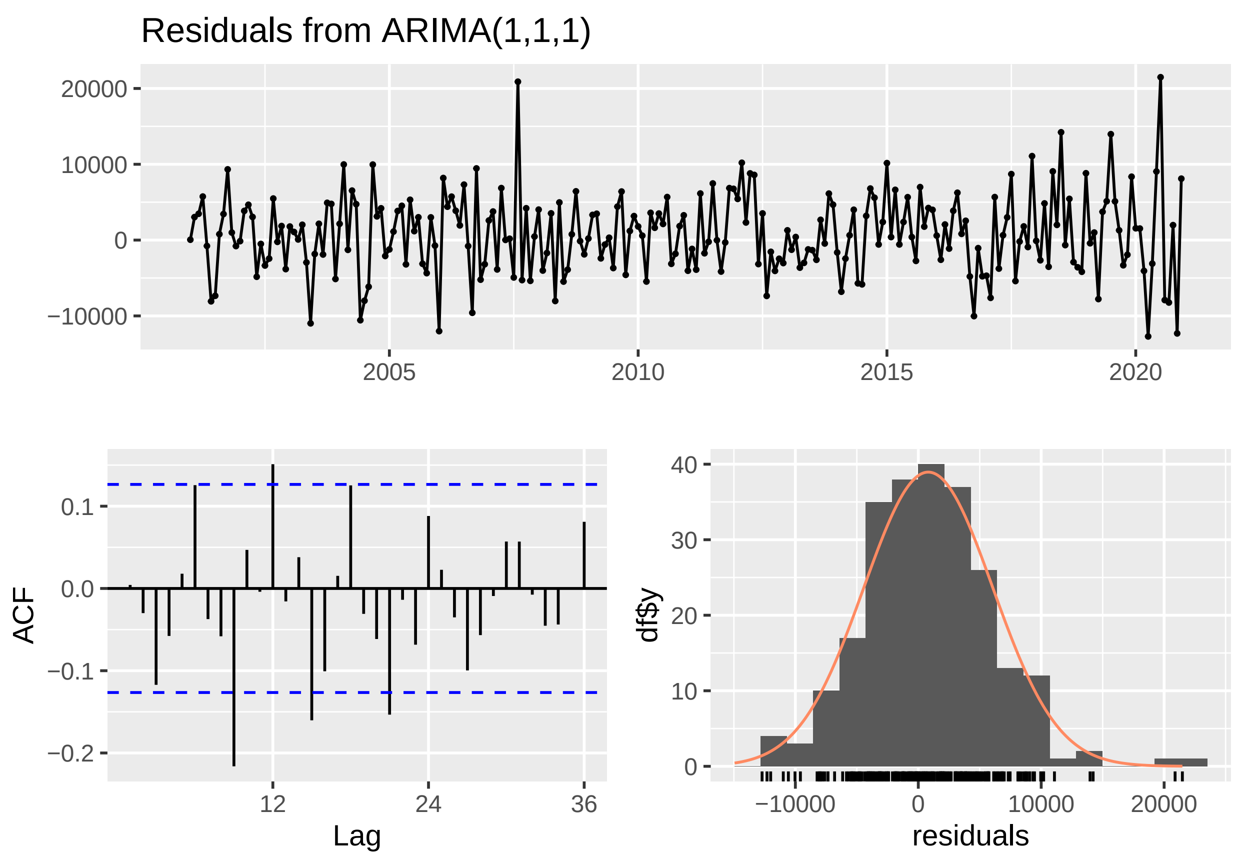
```
arima_modelDCDT <- Arima(deseasonal_decompose_Gas, order=c(1,1,1), include.mean=TRUE)
print(coef(arima_modelDCDT))
```

```
##       ar1        ma1
##  0.5720263 -0.8167161
```

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_modelDCDT)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,1)
## Q* = 52.457, df = 22, p-value = 0.0002702
##
## Model df: 2.   Total lags used: 24
```

Yes this looks like white noise; the resiuduals look random so I think this was a decent model.

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.

```
ns_diff <- nsdiffs(GasTS)
cat("Number of seasonal differencing needed: ",ns_diff)
```

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

above shows I need to difference once so D = 1

```
print(adf.test(GasTS))
```

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

##
```

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

The data is coming back as stationary but that does not really seem true by looking at the plot.

```
print(MannKendall(GasTS))
```

```
## tau = 0.651, 2-sided pvalue =< 2.22e-16
```

p = 1 beacuse the Pacf spikes at 1 lag d = 1 because there is a trend according to the Mann Kendall test and visual inspection q = 0 beacuse my Acf is spiking at lags that look seasonal so my Q = 1

P = 0 beacse the Pacf spikes at each lag = 12 which is already accounted for D = 1 shown above Q = 1 beacuse I have a pacf spike after the one cycle?

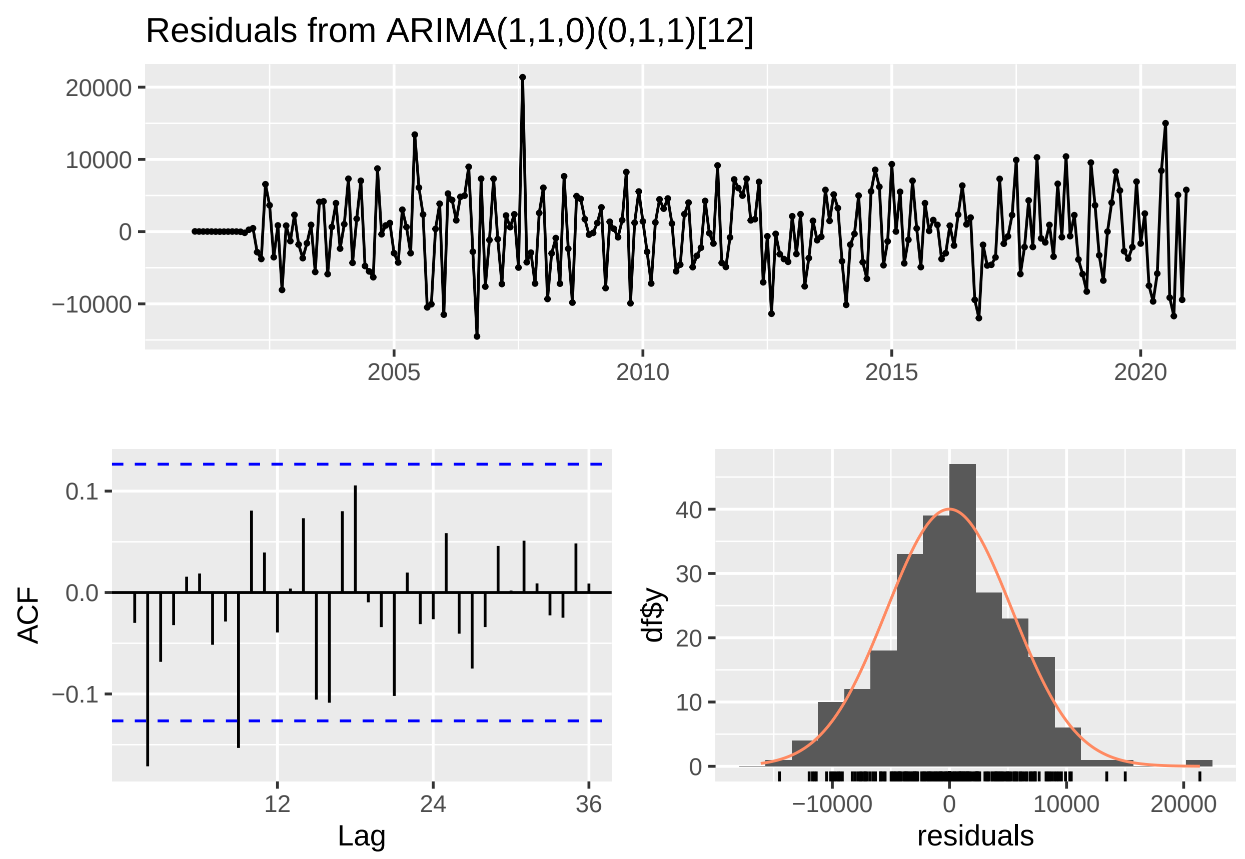
```
Arima(deseasonal_decompose_Gas, order = c(1,1,0), seasonal = c(0,1,1))
```

```
## Series: deseasonal_decompose_Gas
## ARIMA(1,1,0)(0,1,1)[12]
##
## Coefficients:
##           ar1     sma1
##       -0.1808  -0.6898
## s.e.   0.0655   0.0557
##
## sigma^2 = 30626300:  log likelihood = -2281.43
## AIC=4568.86   AICc=4568.96   BIC=4579.13
```

```
arima_modelregular <- Arima(deseasonal_decompose_Gas, order = c(1,1,0), seasonal = c(0,1,1), include.mea
print(coef(arima_modelregular))
```

```
##        ar1       sma1
## -0.1808296 -0.6897965
```

```
checkresiduals(arima_modelregular)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,0)(0,1,1)[12]
## Q* = 33.674, df = 22, p-value = 0.053
##
## Model df: 2.   Total lags used: 24
```

again, the residuals look pretty random

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.

It is kind of difficult to tell here as the residuals are relatively similar

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?

```
auto.arima(deseasonal_decompose_Gas)
```

```
## Series: deseasonal_decompose_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
```

Yes I was right about this one!

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(GasTS)
```

```
## Series: GasTS
## 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
```

I was close but my model had d = 1 due to the trend I visully observed?