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

Assignment 7 - Due date 03/07/24

Aditi Jackson

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

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

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

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

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

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

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

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr   1.1.4     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 errors
```

```r
library(cowplot)
```

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

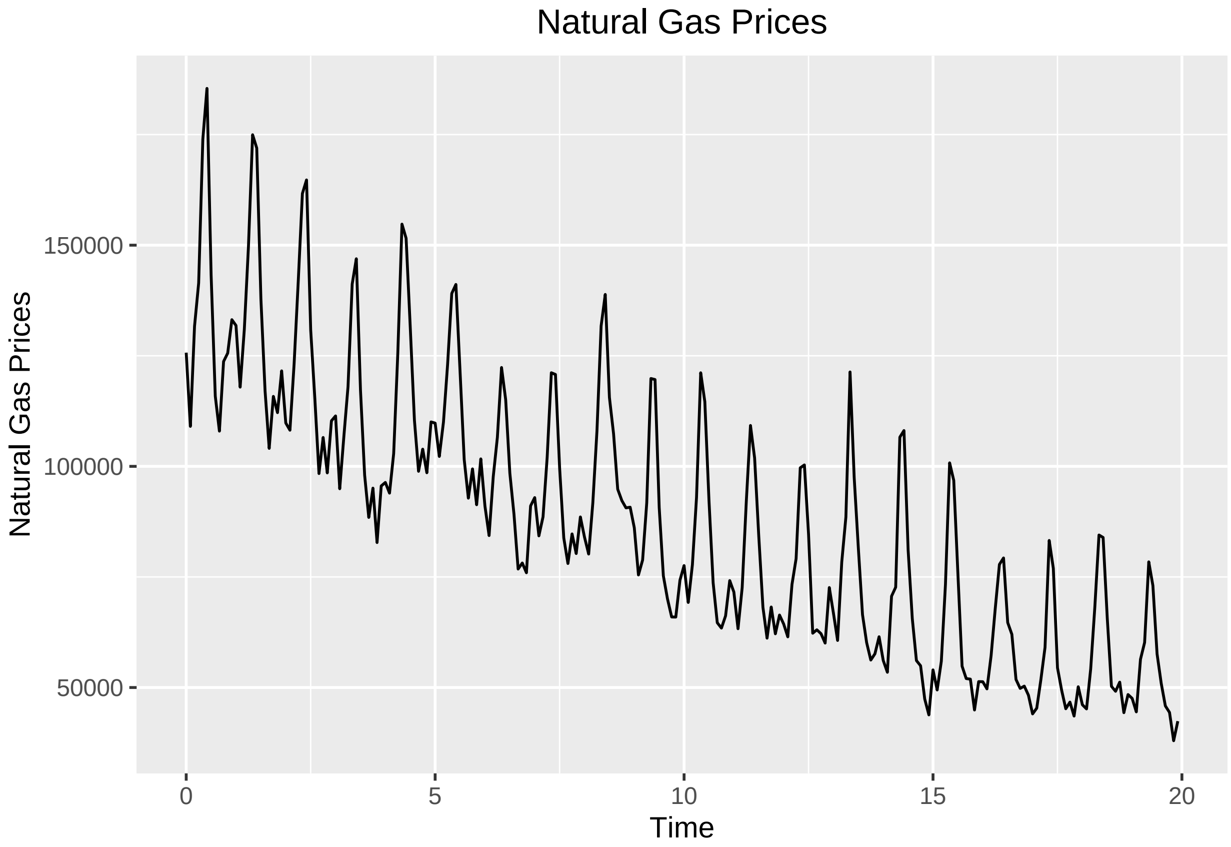
Importing and processing the data set

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**.

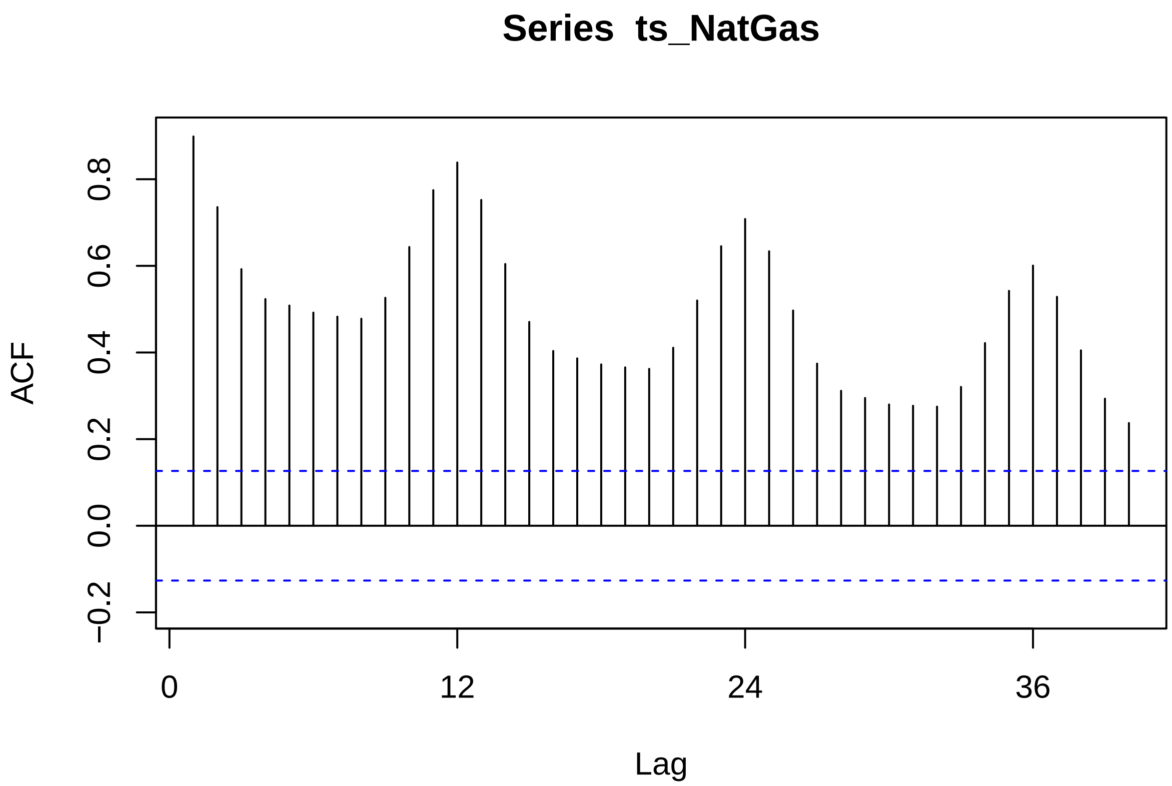
Q1

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.

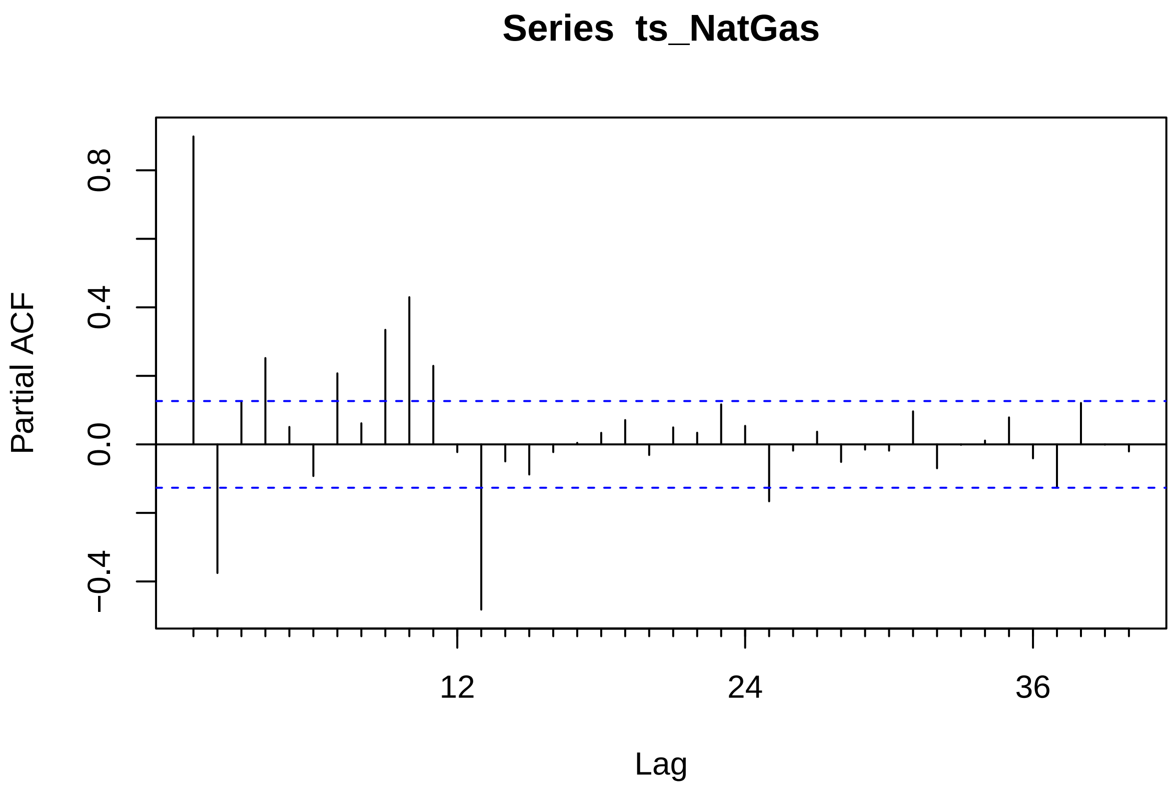
```r
electricity_price_raw <- read.csv("~/ENV797_APJ_S24_NEW/Data/Net_generation_United_States_all_sectors_m

# renaming columns for ease
colnames(electricity_price_raw)[colnames(electricity_price_raw)
                                == "all.fuels..utility.scale..thousand.megawatthours"] <- "All Fuels"
colnames(electricity_price_raw)[colnames(electricity_price_raw)
                                == "coal.thousand.megawatthours"] <- "Coal"
colnames(electricity_price_raw)[colnames(electricity_price_raw)
                                == "natural.gas.thousand.megawatthours" ] <-"NatGas"
colnames(electricity_price_raw)[colnames(electricity_price_raw)
                                == "nuclear.thousand.megawatthours"] <- "Nuclear"
colnames(electricity_price_raw)[colnames(electricity_price_raw)
                                ==  "conventional.hydroelectric.thousand.megawatthours"] <- "Hydro"

# converting Month column to date object
electricity_price_raw$Month <- my(electricity_price_raw$Month)

# creating time series object
ts_NatGas <- ts(electricity_price_raw[,4],start = 1/1/2001,frequency=12)

# initial plot
NatGas_plot <- autoplot(ts_NatGas)+
  ggtitle("Natural Gas Prices")+
  theme(plot.title = element_text(hjust = 0.5))+
  ylab("Natural Gas Prices")
NatGas_plot
```



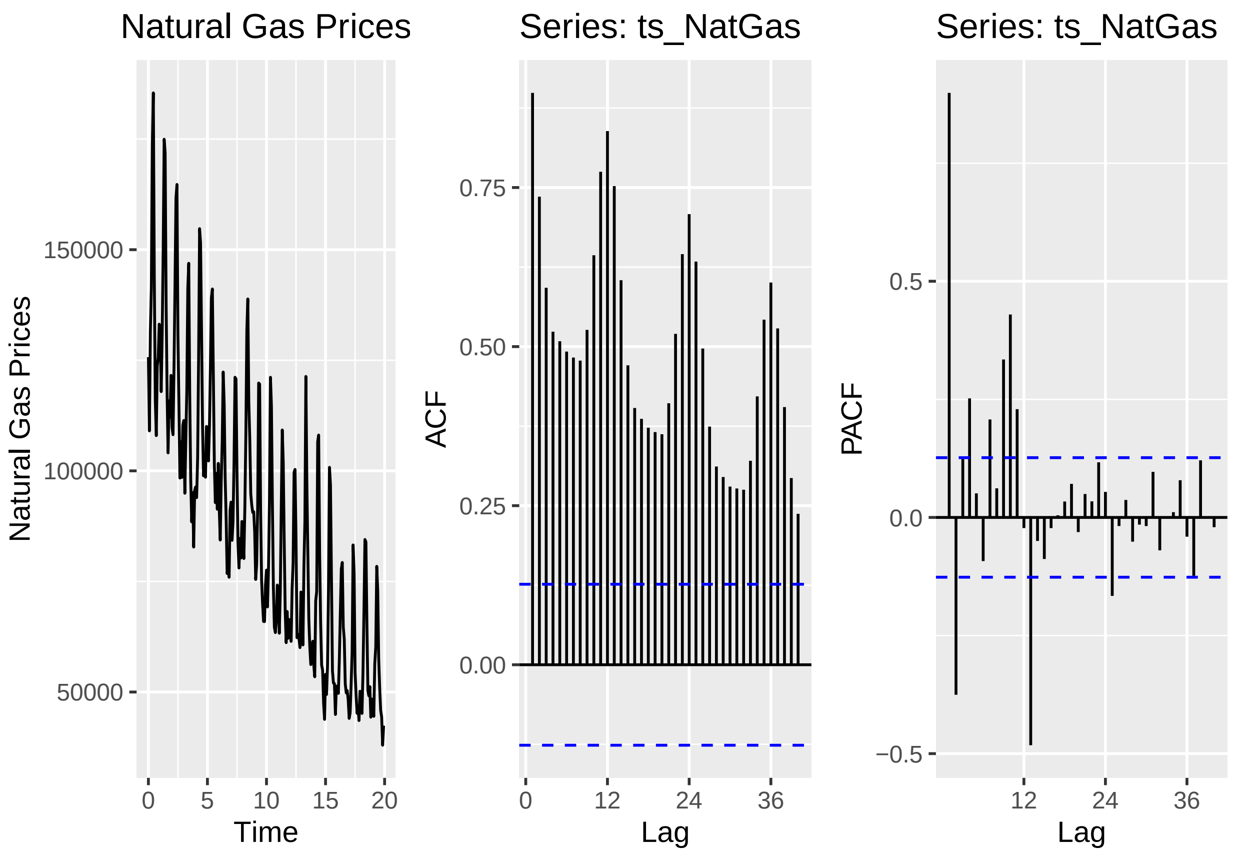
```
# ACF and PACF
NatGas_ACF <- Acf(ts_NatGas,lag.max = 40)
```



```
NatGas_PACF <- Pacf(ts_NatGas,lag.max = 40)
```



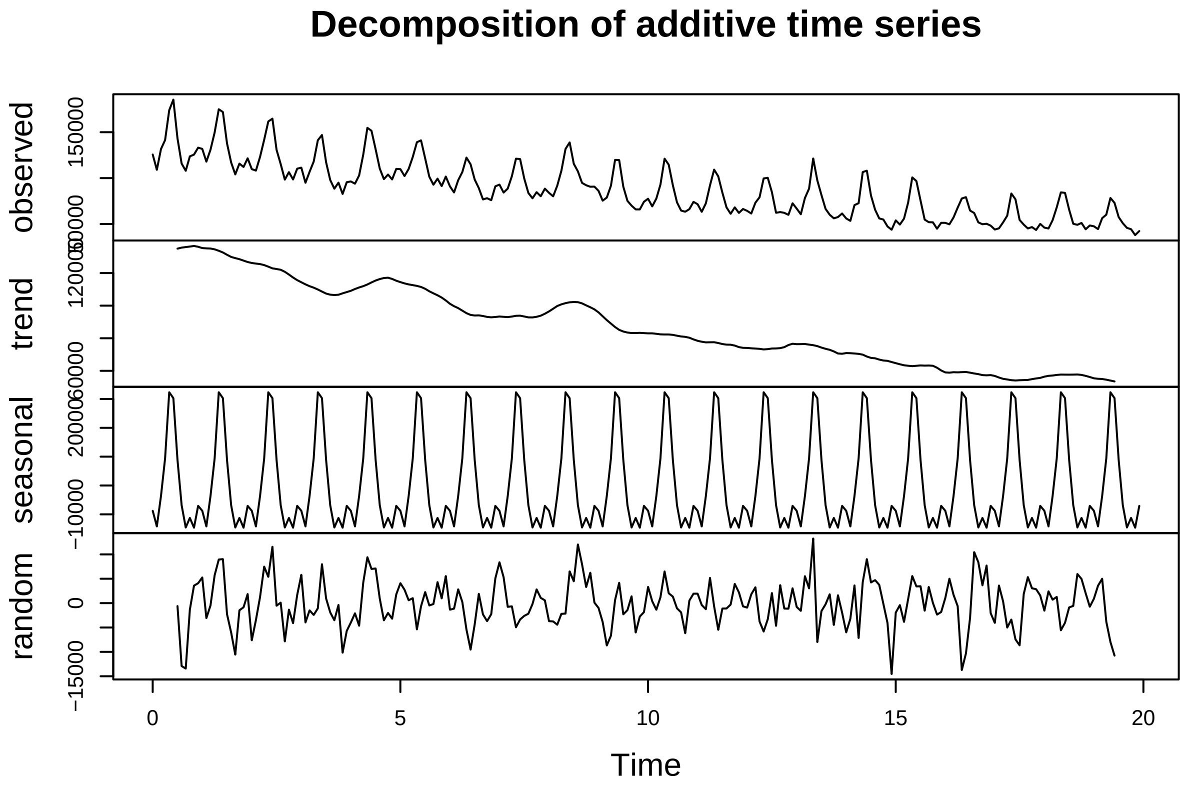
```
# plotting in one grid
plot_grid(NatGas_plot,
          autoplot(NatGas_ACF),
          autoplot(NatGas_PACF),
          ncol=3)
```



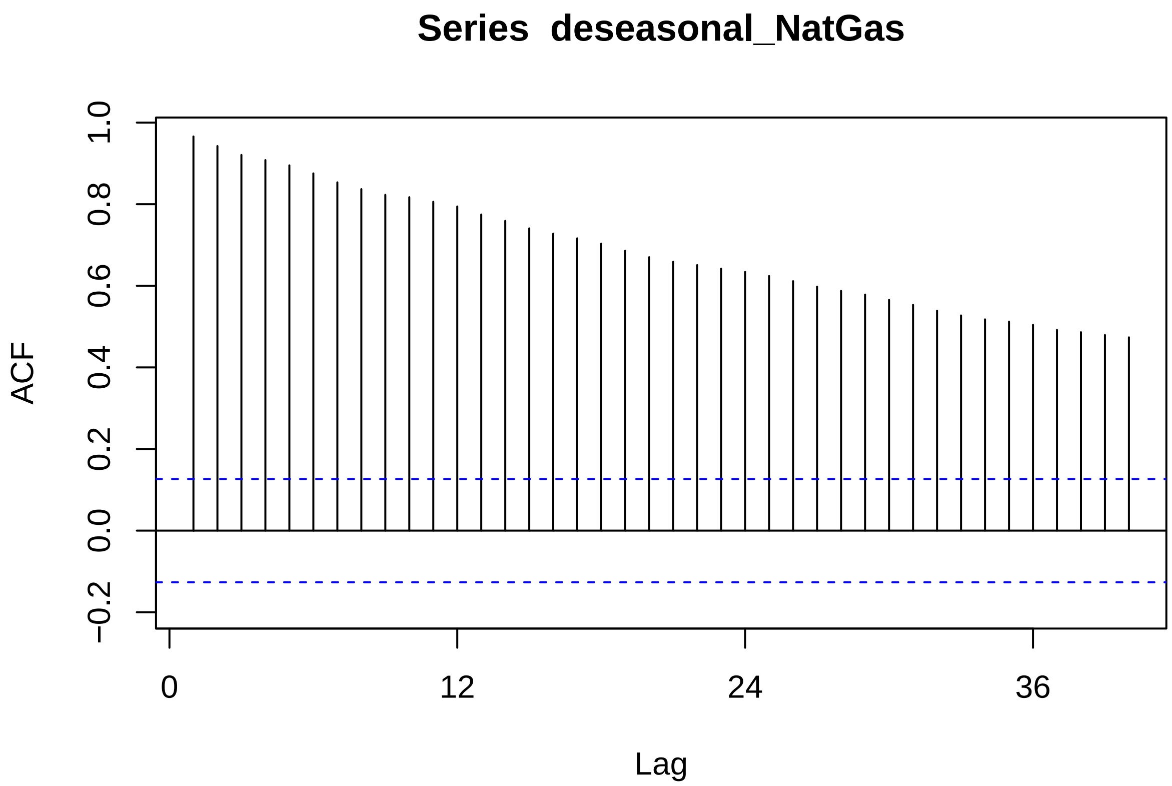
Q2

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.

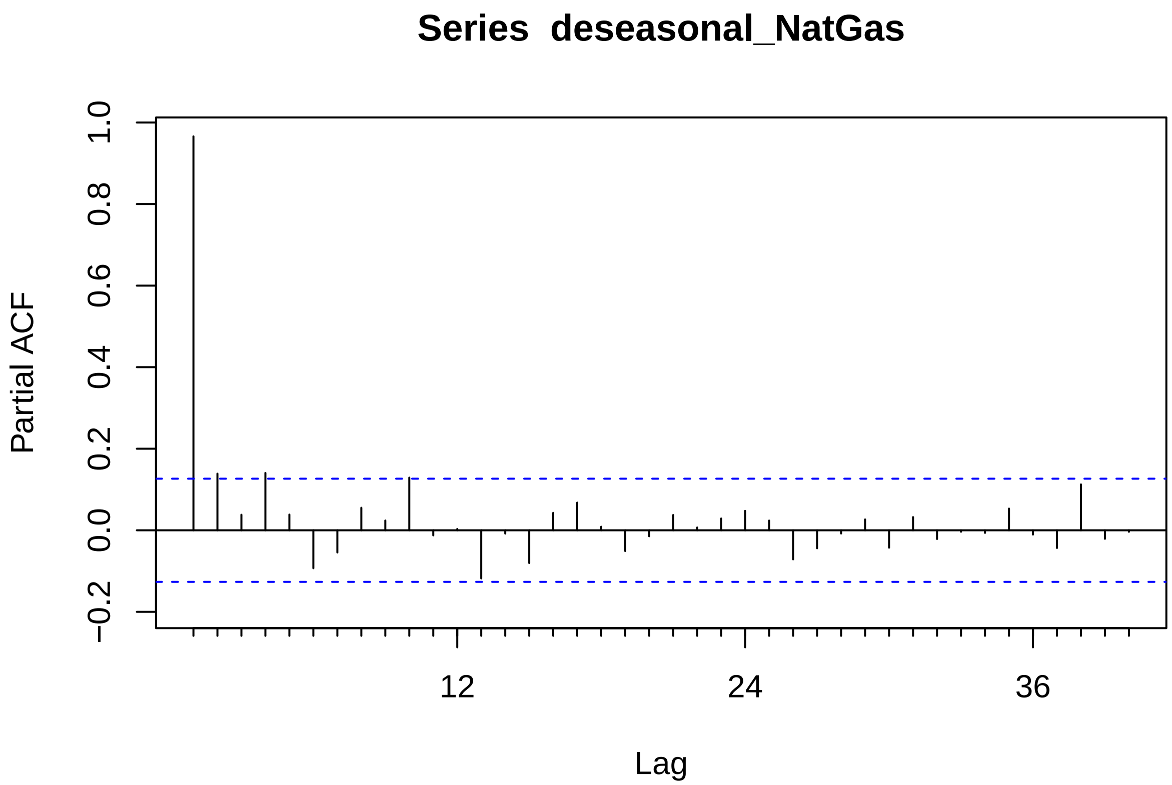
```r
# decomposing series
decomp_NatGas <-decompose(ts_NatGas,"additive")
plot(decomp_NatGas)
```



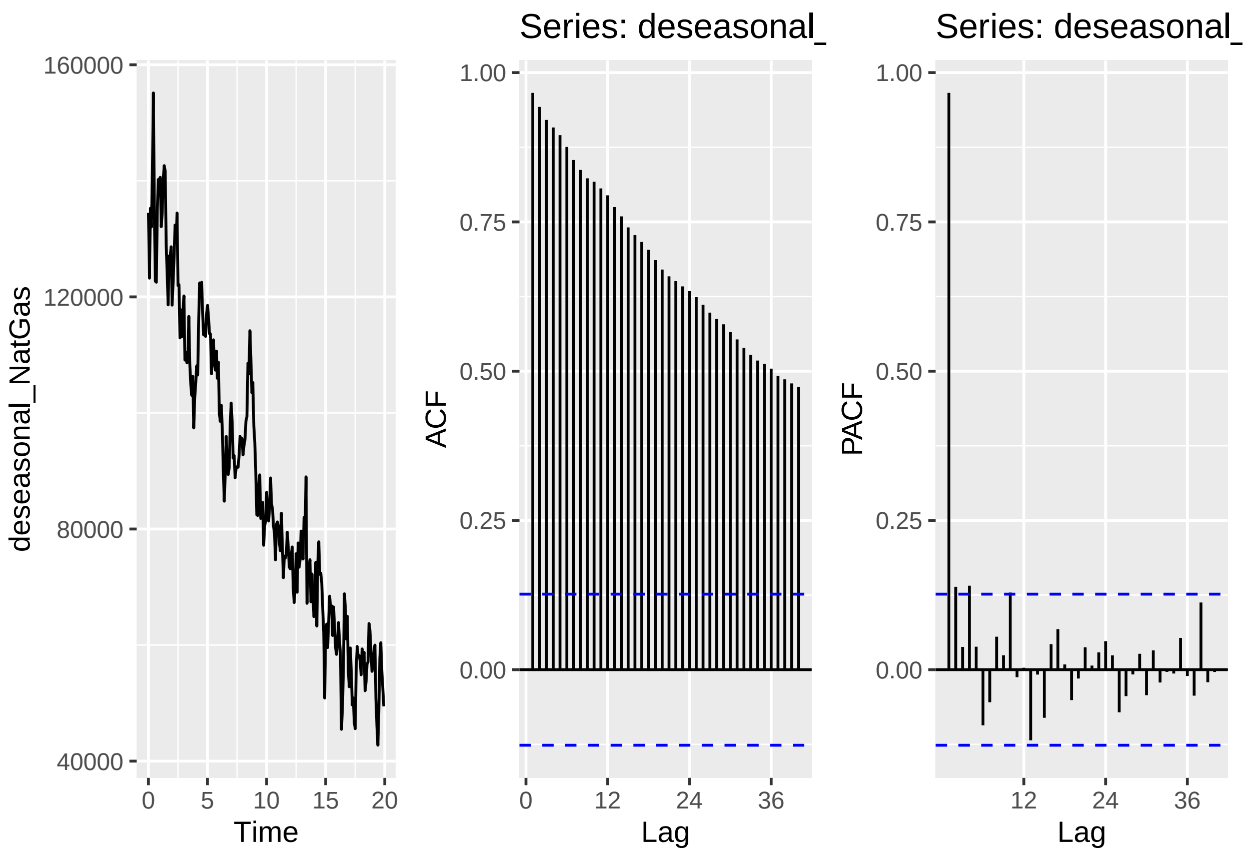
```r
# deseasoning series
deseasonal_NatGas <- seasadj(decomp_NatGas)

# creating ACF and PACF objects
deseasonal_NatGas_ACF <- Acf(deseasonal_NatGas,lag.max = 40)
```



```
deseasonal_NatGas_PACF <- Pacf(deseasonal_NatGas, lag.max = 40)
```



```
# plots
plot_grid(autoplot(deseasonal_NatGas),
          autoplot(deseasonal_NatGas_ACF),
          autoplot(deseasonal_NatGas_PACF),
          ncol=3)
```



> Before and after deseasoning, the series shows a clear downward trend over time. Although deseasoning appears to have altered some of the magnitude in price variation over time. After deseasoning the ACF appears much better (e.g. exponentially declining dependence on time), and the PACF also appears improved with only one highly significant lag (vs. multiple significant lags in Q1).

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.

```
# Mann Kendall test
MannKendall(deseasonal_NatGas)
```

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

```
# ADF test
#Null hypothesis is that data has a unit root
print("Results for ADF test/n")
```

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

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

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

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

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

> The Mann Kendall test has a test-stat (Tau) of -0.843, indicating a decreasing trend is present. The p-value is less than 0.05, which tells us that the identified trend is significant. The ADF test has a test-statistic of -4.0574, which is less than 1 and the p-value is 0.01, which is less than alpha = 0.05. Taken together, this indicates strong evidence against the null hypothersis of non-stationarity. We can surmise that the data is stationary from this test.

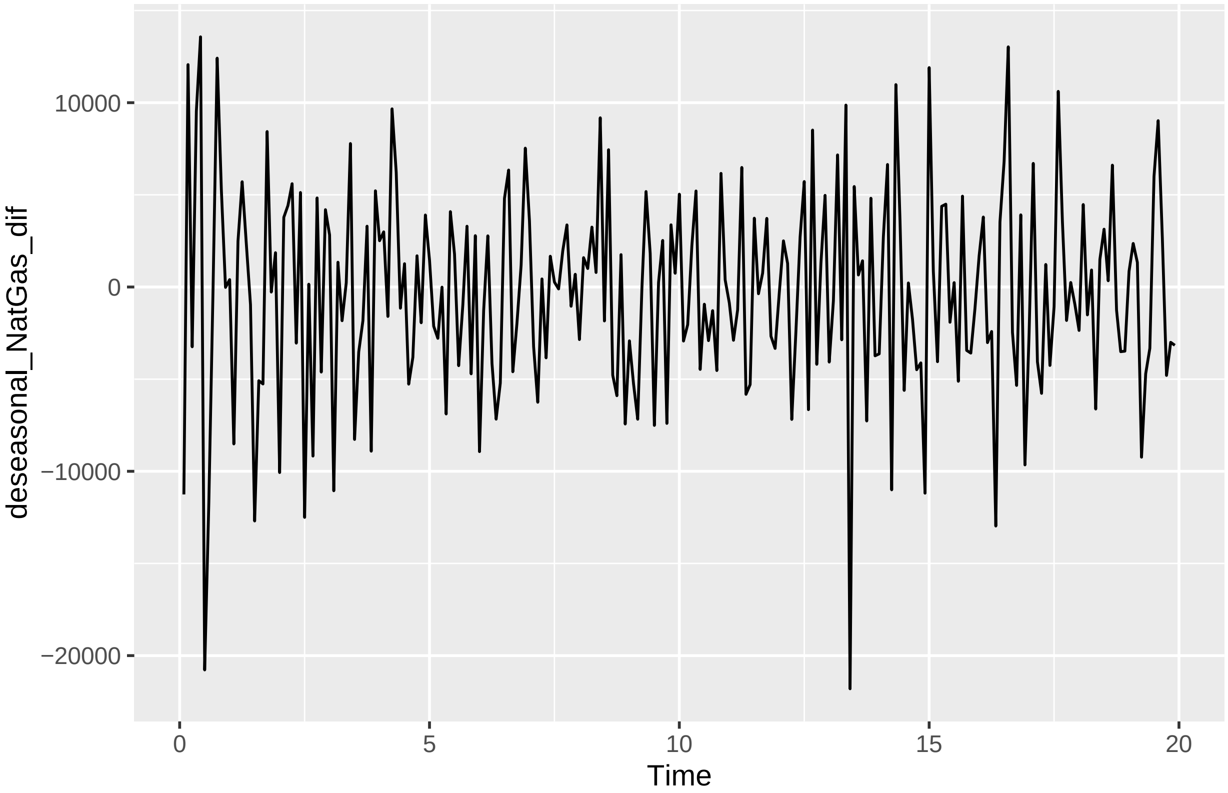
Q4

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.

```r
# d
## since the M-K test indicates a trend is present,
### I believe I would need to difference the series. Therefore d = 1

# Finding # times need to difference
n_diff <- ndiffs(deseasonal_NatGas)
cat("Number of differencing needed: ",n_diff)
```

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

```r
#Differencing series once at lag 1 to remove the trend
deseasonal_NatGas_dif <- diff(deseasonal_NatGas,differences=1,lag=1)

# plotting series
autoplot(deseasonal_NatGas_dif)
```



```
# p
## since the ACF decays exponentially,
### it seems and the PACF cuts off after lag 1,
#### it appears that this an AR(1) process and therefore p = 1


# q
## there does not appear to be a MA component since the PACF
### is not showing an exponential decay. Therefore, I think q = 1
```

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.

```
# fitting arima model using deseasoned series
Model_110 <- Arima(deseasonal_NatGas,
                   order=c(1,1,0),
                   include.mean = TRUE,
                   include.drift = TRUE)
# printing model results
print(Model_110)
```

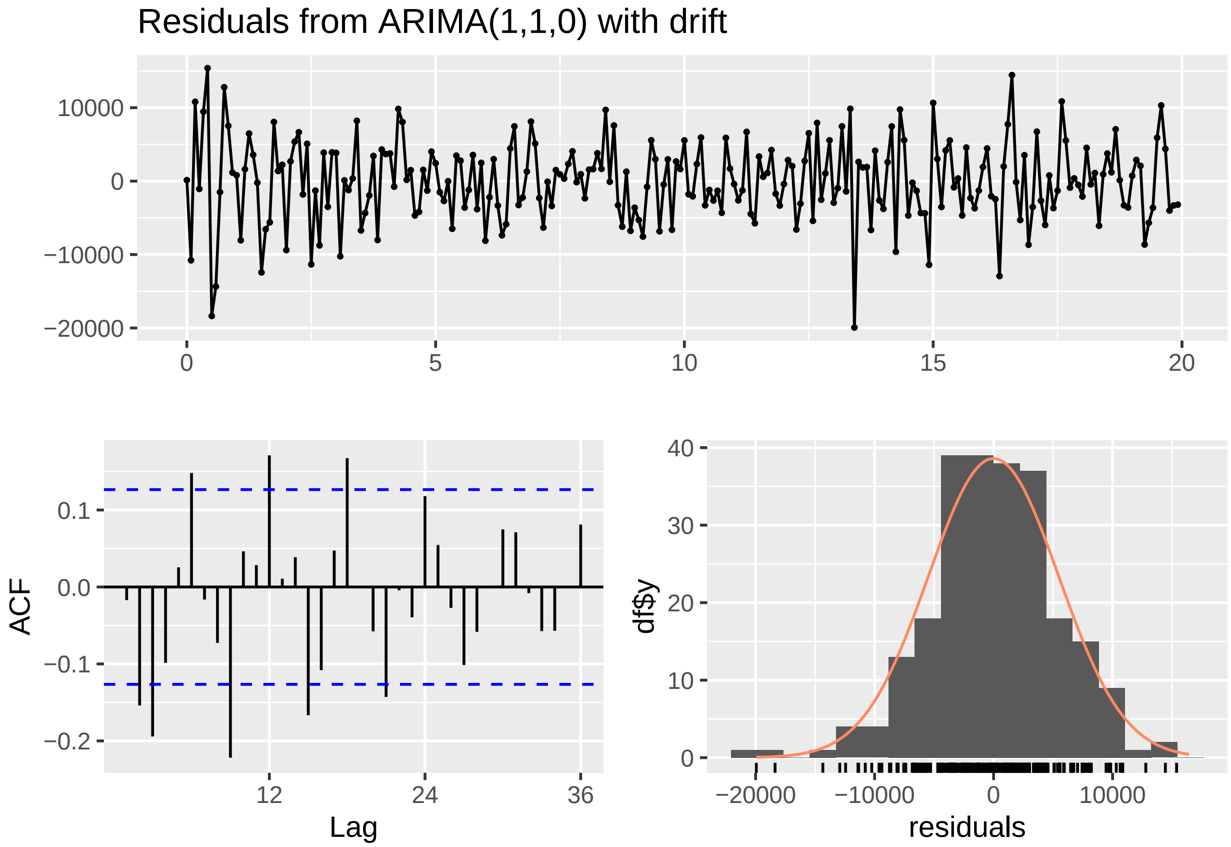
```
## Series: deseasonal_NatGas
## ARIMA(1,1,0) with drift
##
```

```
## Coefficients:
##           ar1      drift
##       -0.1479  -348.3913
## s.e.   0.0644   308.8359
##
## sigma^2 = 30254130:  log likelihood = -2396.54
## AIC=4799.07   AICc=4799.18   BIC=4809.5
```

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?

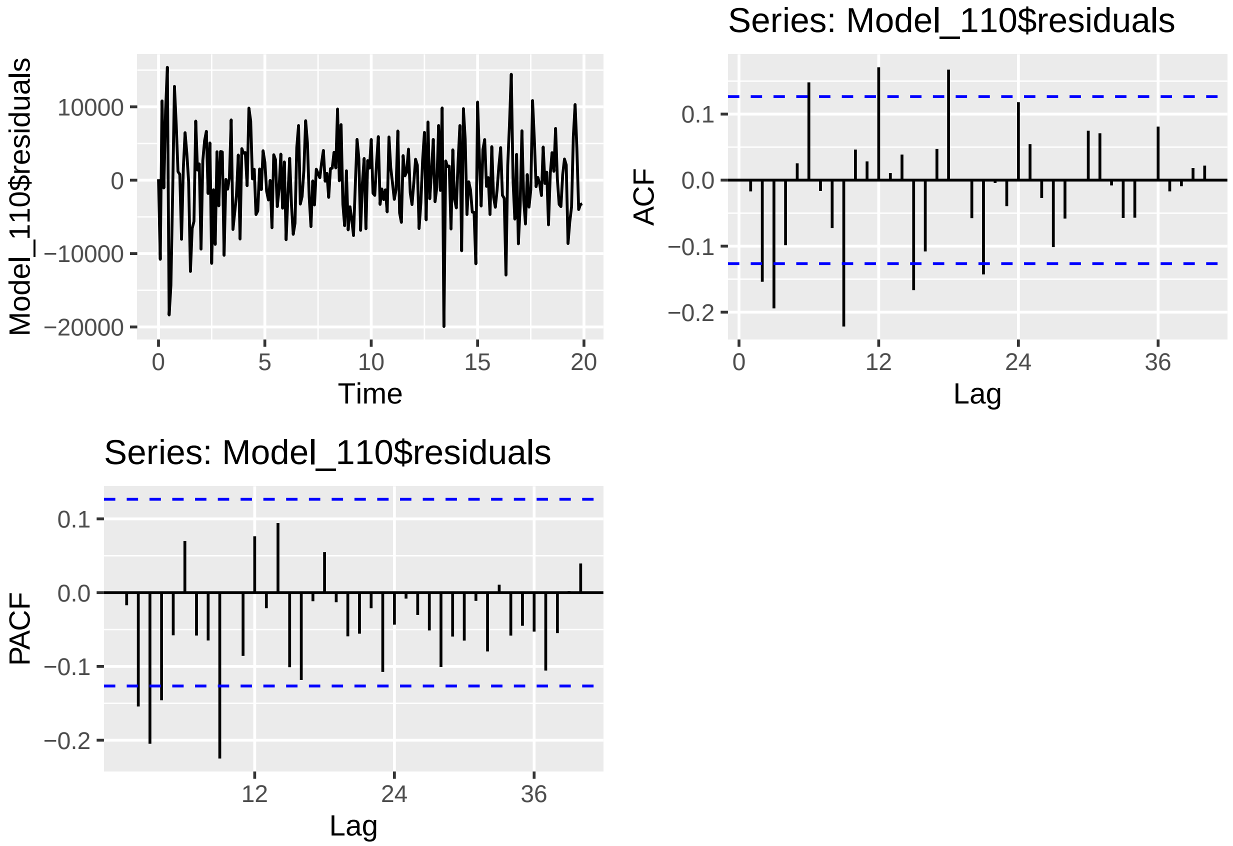
```
# plotting residuals
checkresiduals(Model_110)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,0) with drift
## Q* = 73.991, df = 23, p-value = 2.899e-07
##
## Model df: 1.   Total lags used: 24
```

```
# plottig with autoplot since the function above doe snot appear to generate the PACF
plot_grid(
  autoplot(Model_110$residuals),
```

```r
  autoplot(Acf(Model_110$residuals,lag.max=40, plot = FALSE)),
  autoplot(Pacf(Model_110$residuals,lag.max=40, plot = FALSE))
)
```



> No, the residuals do not look like a white noise series. The series still shows some sort of consisent pattern in how it ocilates and the ACF seems to have consistent spikes every 3-4 observations. Additionally, some of the lags of the PACF are showing as significant. For this to be a white noise series we'd want to see no significance in the PACF and no patterns in the ACF and series itself.

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.

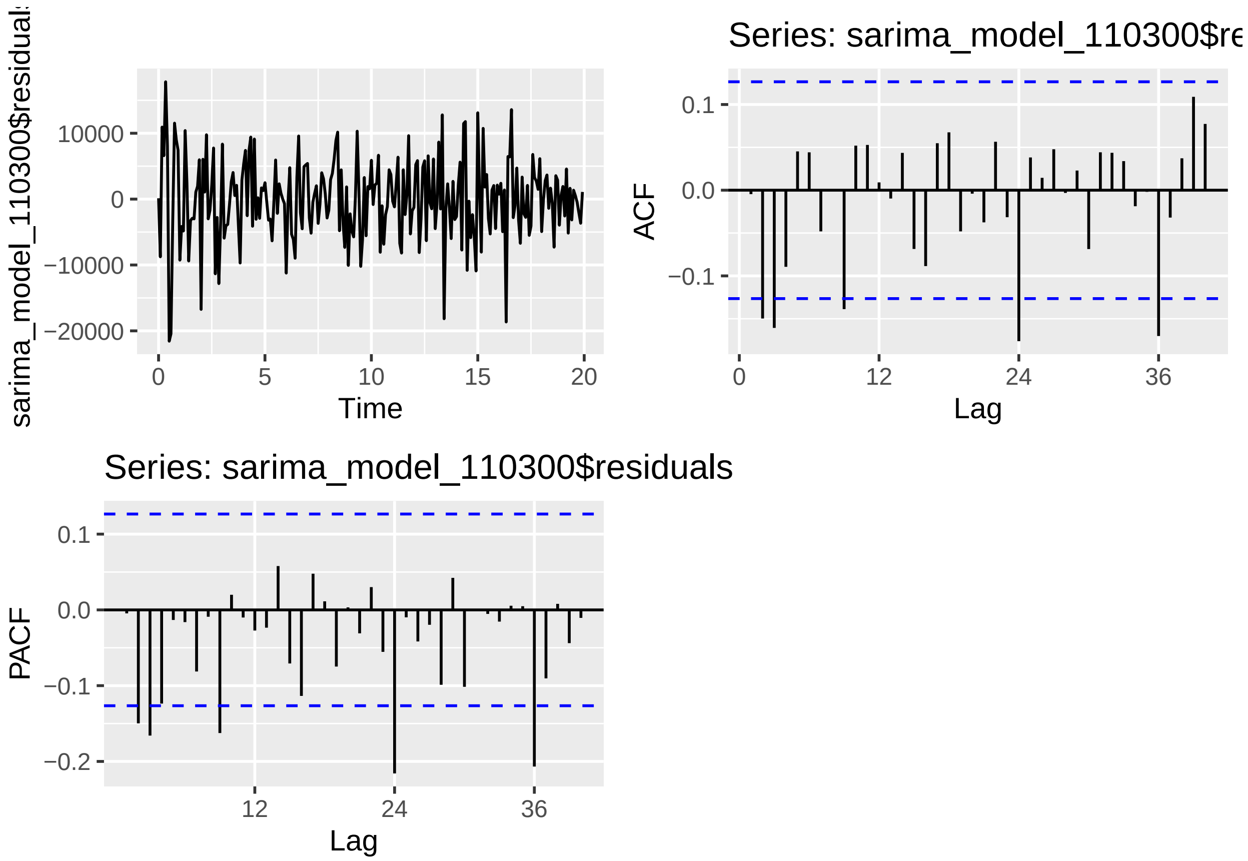
```r
# (p,d,q) from above i (1,1,0)

# determining P, Q, D
## P = 3 because the PACF of the original series shows 3 significant spikes
## D = 0 <- i'm unsure how to tell if the seasonal series needs to be differenced so assuming 0
## Q = 0 because P+Q < 1

# Fitting the SARIMA model and printing results
sarima_model_110300 <- Arima(ts_NatGas, order = c(1, 1, 0), seasonal = c(3, 0, 0))
summary(sarima_model_110300)
```

```
## Series: ts_NatGas
## ARIMA(1,1,0)(3,0,0)[12]
##
## Coefficients:
```

```
## Warning in sqrt(diag(x$var.coef)): NaNs produced
```

```
##           ar1    sar1    sar2    sar3
##        -0.118  0.2934  0.3466  0.2759
## s.e.      NaN     NaN     NaN     NaN
##
## sigma^2 = 38039076:  log likelihood = -2420.64
## AIC=4851.29   AICc=4851.55   BIC=4868.67
##
## Training set error measures:
##                     ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -64.08996 6102.999 4724.979 -0.2952707 5.627399 0.5776666
##                      ACF1
## Training set -0.004601313
```

```
# checking residuals
plot_grid(
  autoplot(sarima_model_110300$residuals),
  autoplot(Acf(sarima_model_110300$residuals,lag.max=40, plot = FALSE)),
  autoplot(Pacf(sarima_model_110300$residuals,lag.max=40, plot = FALSE))
)
```



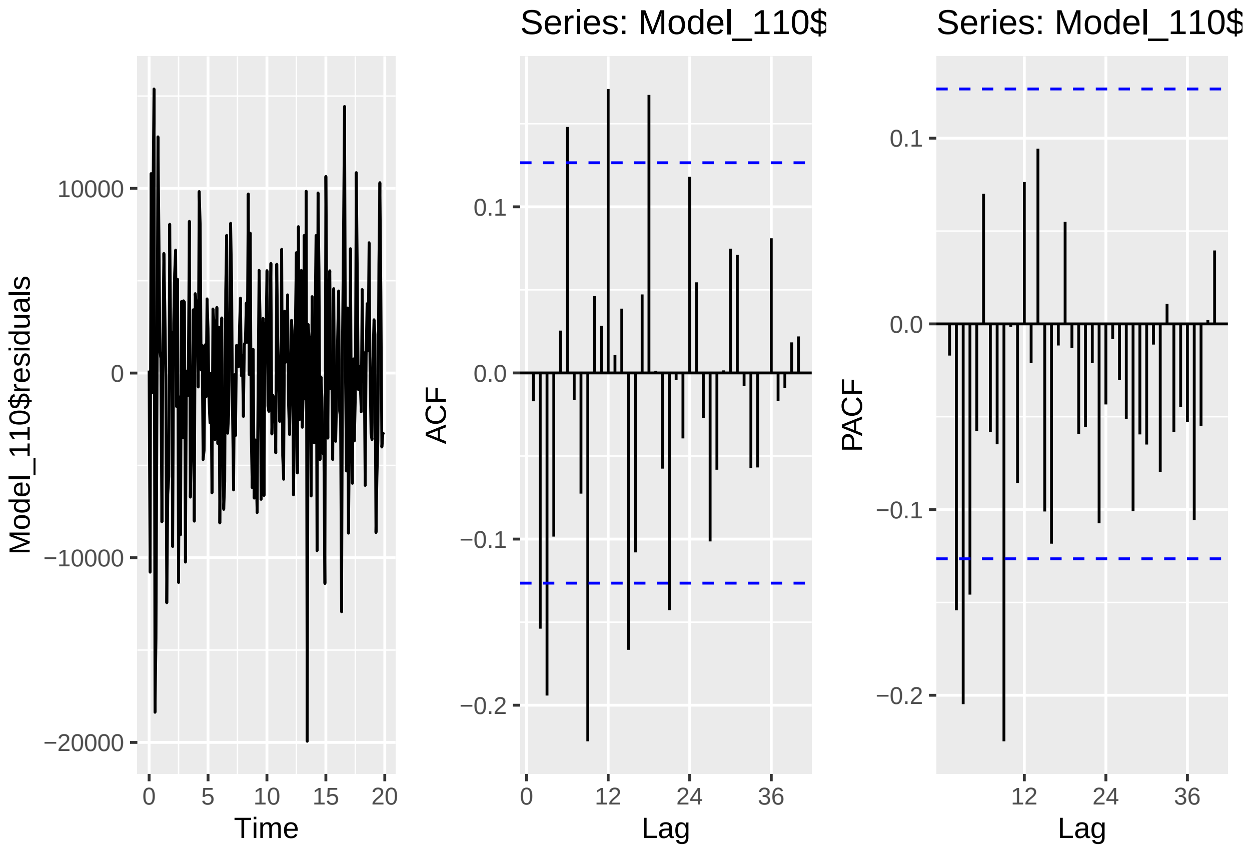
> The residuals still appear to have something going on. The series appears somewhat equally spaced and the ACF and PACF have signicant lags, indicating that this model may not be the best fit / there is soemthing

off with the specified order.

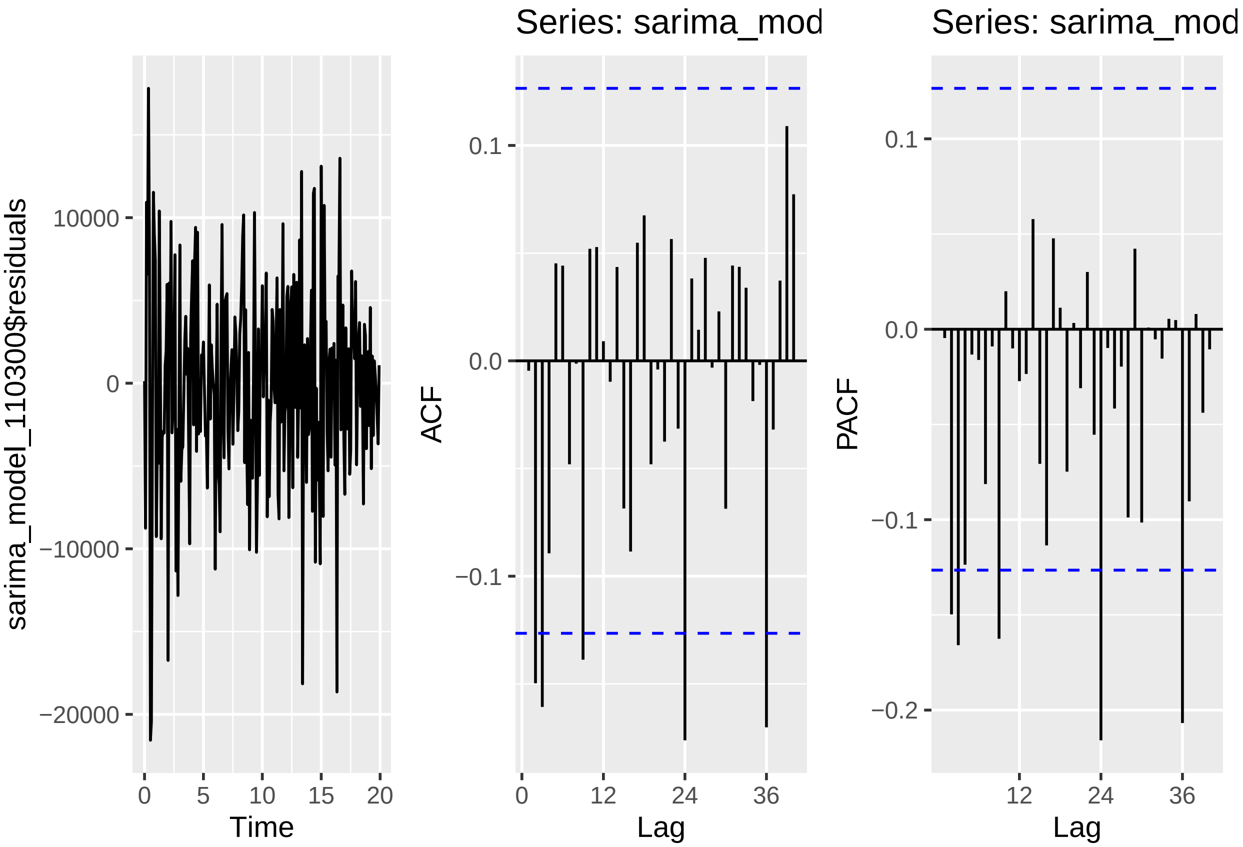
Q8

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.

```r
plot_grid(
  autoplot(Model_110$residuals),
  autoplot(Acf(Model_110$residuals,lag.max=40, plot = FALSE)),
  autoplot(Pacf(Model_110$residuals,lag.max=40, plot = FALSE)),
  ncol = 3
)
```



```r
plot_grid(
  autoplot(sarima_model_110300$residuals),
  autoplot(Acf(sarima_model_110300$residuals,lag.max=40, plot = FALSE)),
  autoplot(Pacf(sarima_model_110300$residuals,lag.max=40, plot = FALSE)),
  ncol=3
)
```



> It appears that the sarima model is better at representing the series. This is because the ACF and PACF appears to have fewer signiciant lags (ideally we'd want no significant PACF lags). Since the model has a seasonal component, it makes sense that fitting a seasonal model would better represent the series. When we subtract out the seasonal component and then try to model the series we are missing a key element of the historical pattern. It's not necessarily fair to compare these two models since one is specifically incorporating seasonality while the other ignores it.

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_Q9 <- auto.arima(deseasonal_NatGas)
print(ARIMA_Q9)
```

```
## Series: deseasonal_NatGas
## 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 non-seasonal component is p = 3, d = 1, q = 0. The order of the seasonal component is P = 1, D = 0, Q = 1. This does not match with my answer in Q4. While I correctly identified d = 1 and q = 0, I said that p =1.

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?

```
ARIMA_Q10 <- auto.arima(ts_NatGas)
print(ARIMA_Q10)
```

```
## Series: ts_NatGas
## 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 non-seasonal component is p = 1, d = 0, q = 1, P = 2, D = 1, Q = 2. This answer does not match my answer to Q7.