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

Assignment 7 - Due date 03/20/23

Angela Zeng

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

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

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

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

```
library(tseries)
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✔ dplyr     1.1.0     ✔ readr     2.1.4
## ✔ forcats   1.0.0     ✔ stringr   1.5.0
## ✔ ggplot2   3.4.1     ✔ tibble    3.1.8
## ✔ lubridate 1.9.2     ✔ tidyr     1.3.0
## ✔ purrr     1.0.1
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the ]8;;http://conflicted.r-lib.org/conflicted package]8;; to force all
conflicts to become errors
```

```
library(Kendall)
```

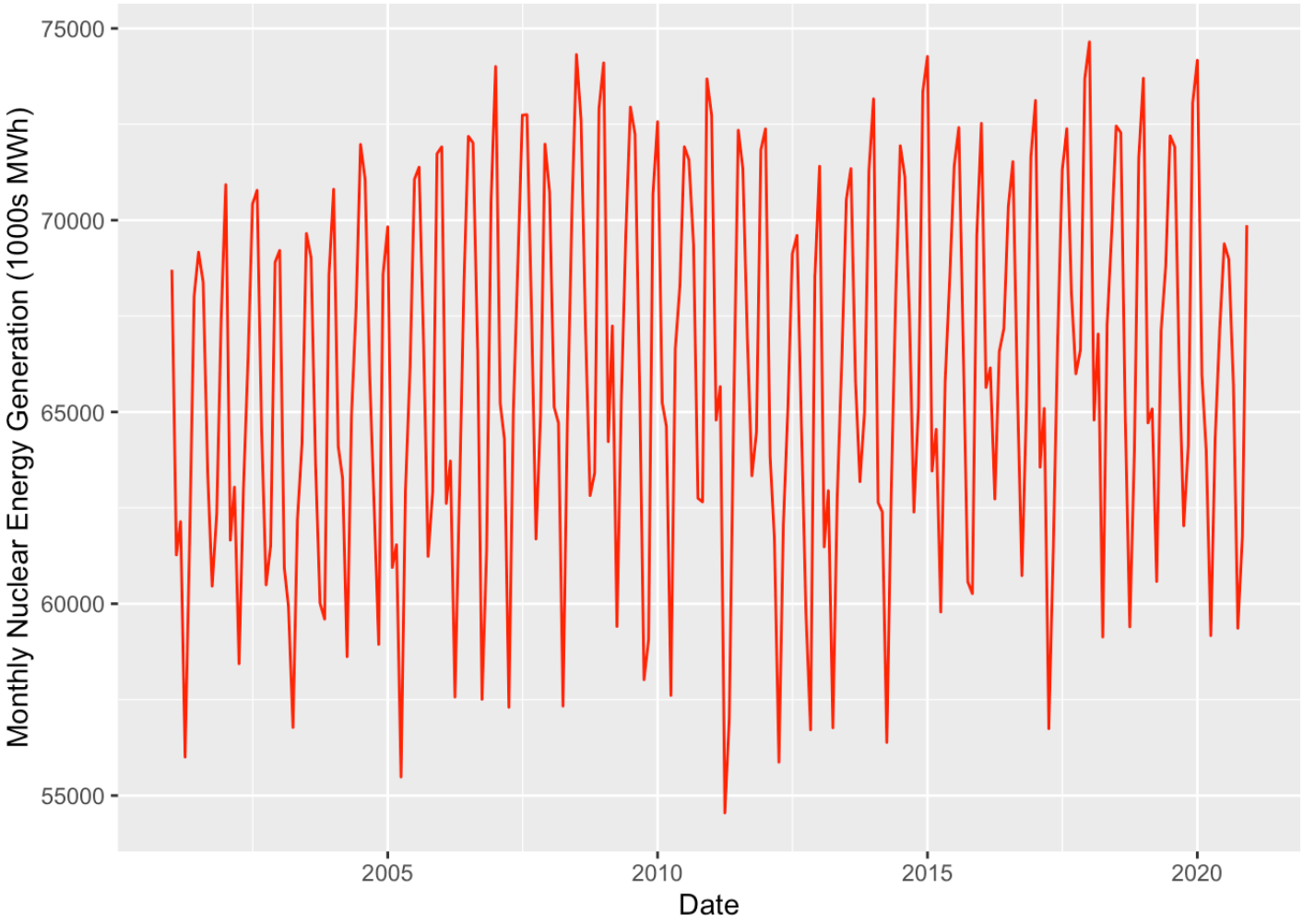
Importing and processing the data set

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**.

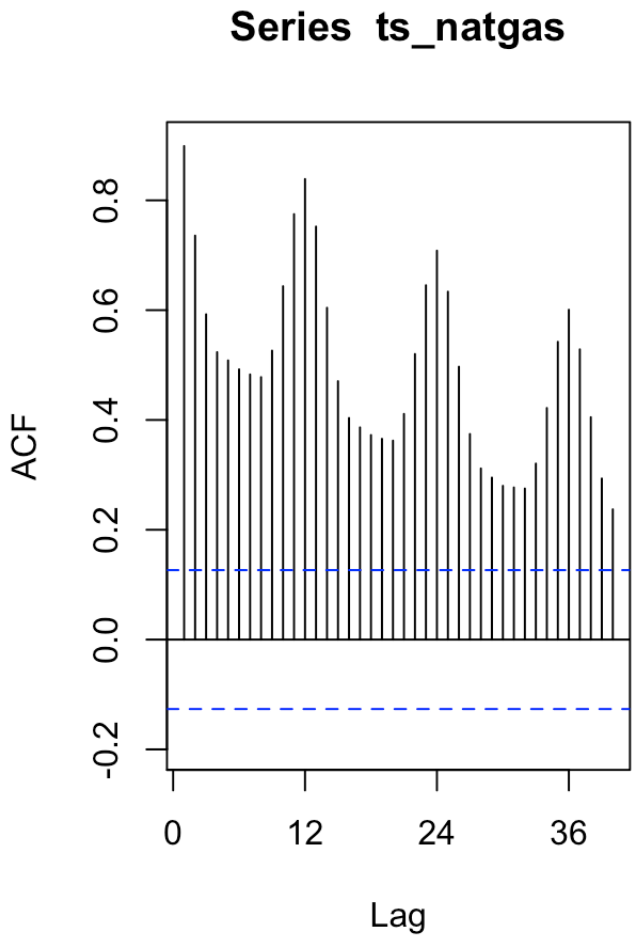
Q1

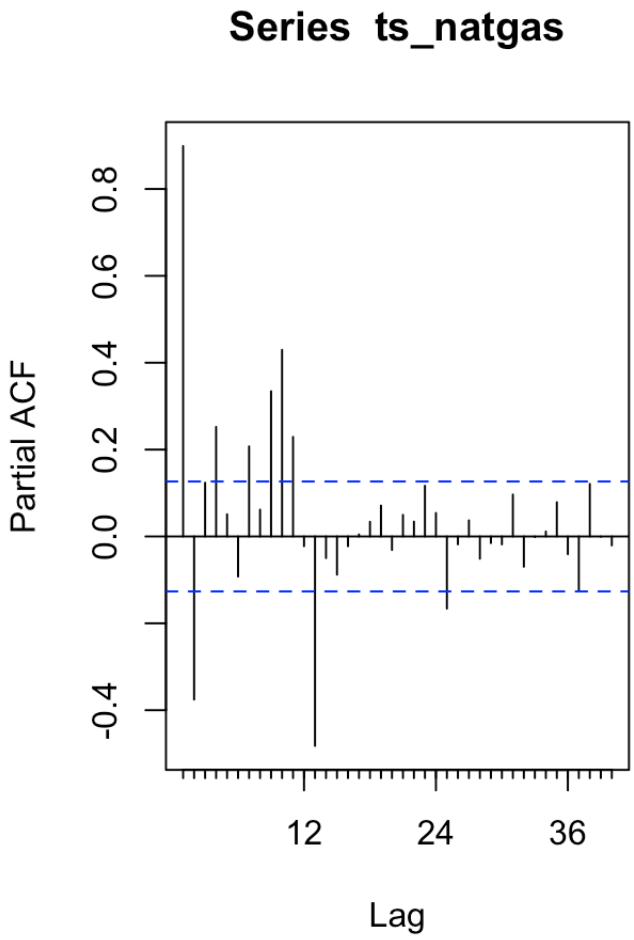
Import the csv file and create a time series object for natural gas. Make you sure you specify the **start=** and **frequency=** arguments. Plot the time series over time, ACF and PACF.

```
#load data
data <- read.csv(file="../Data/Net_generation_United_States_all_sectors_monthly.csv",
skip = 4, header = TRUE, dec = ".", sep=",",stringsAsFactors = TRUE)

#lubridate & reverse dates so that it is chronological
colnames(data)[1]<- "Date"
data$Date<- my(data$Date)
data<- data %>%
  arrange(Date)

#create ts object
ts_natgas<- ts(data[,4], frequency = 12, start = c(2001,1))

#plot ts over time
ggplot(data, aes(x= Date , y= nuclear.thousand.megawatthours)) +
  geom_line(color= "red") +
  ylab('Monthly Nuclear Energy Generation (1000s MWh)')
```




```
#ACF & PACF
par(mfrow=c(1,2))
Acf(ts_natgas,lag.max=40, plot=TRUE)
Pacf(ts_natgas,lag.max=40, plot=TRUE)
```

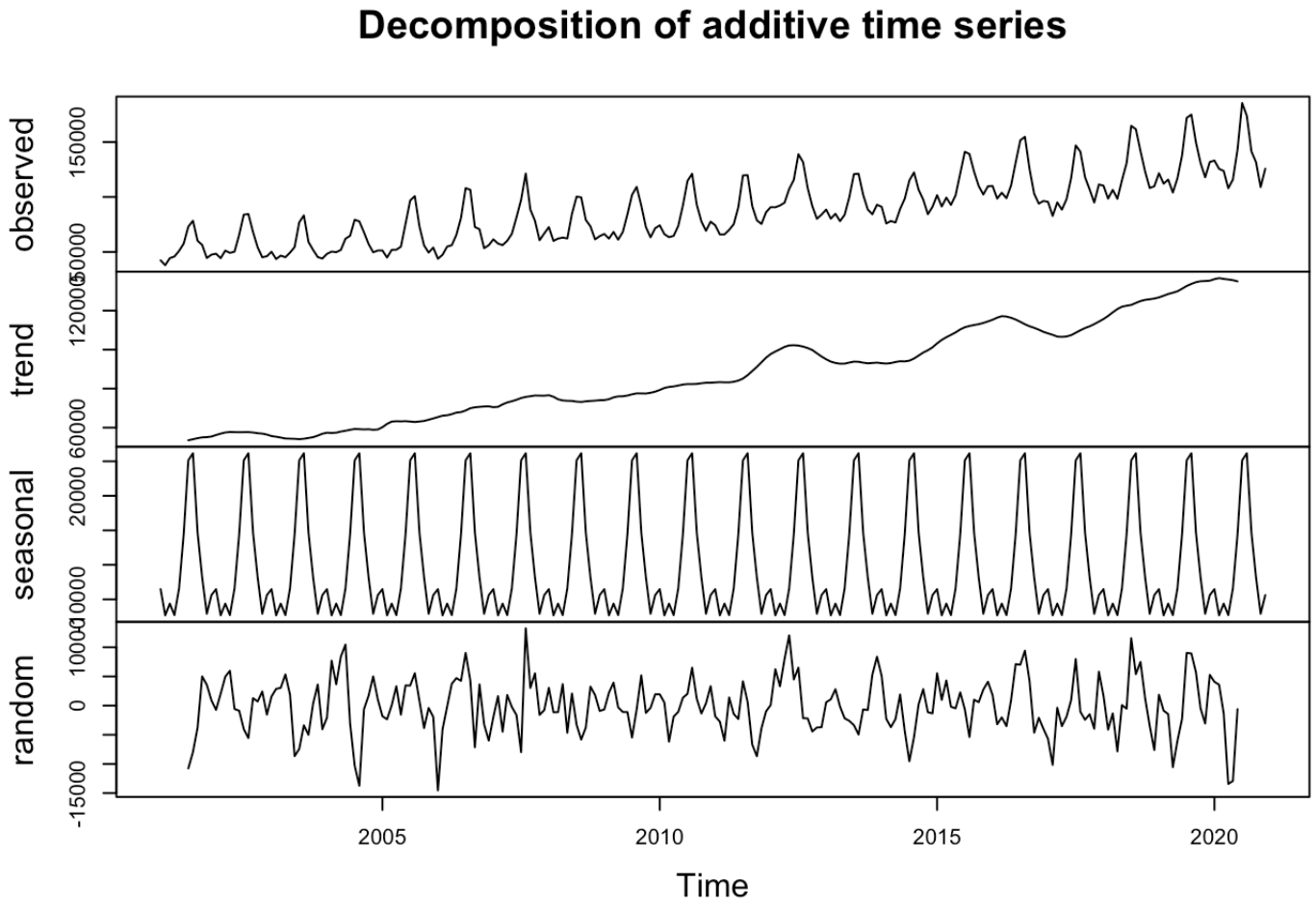




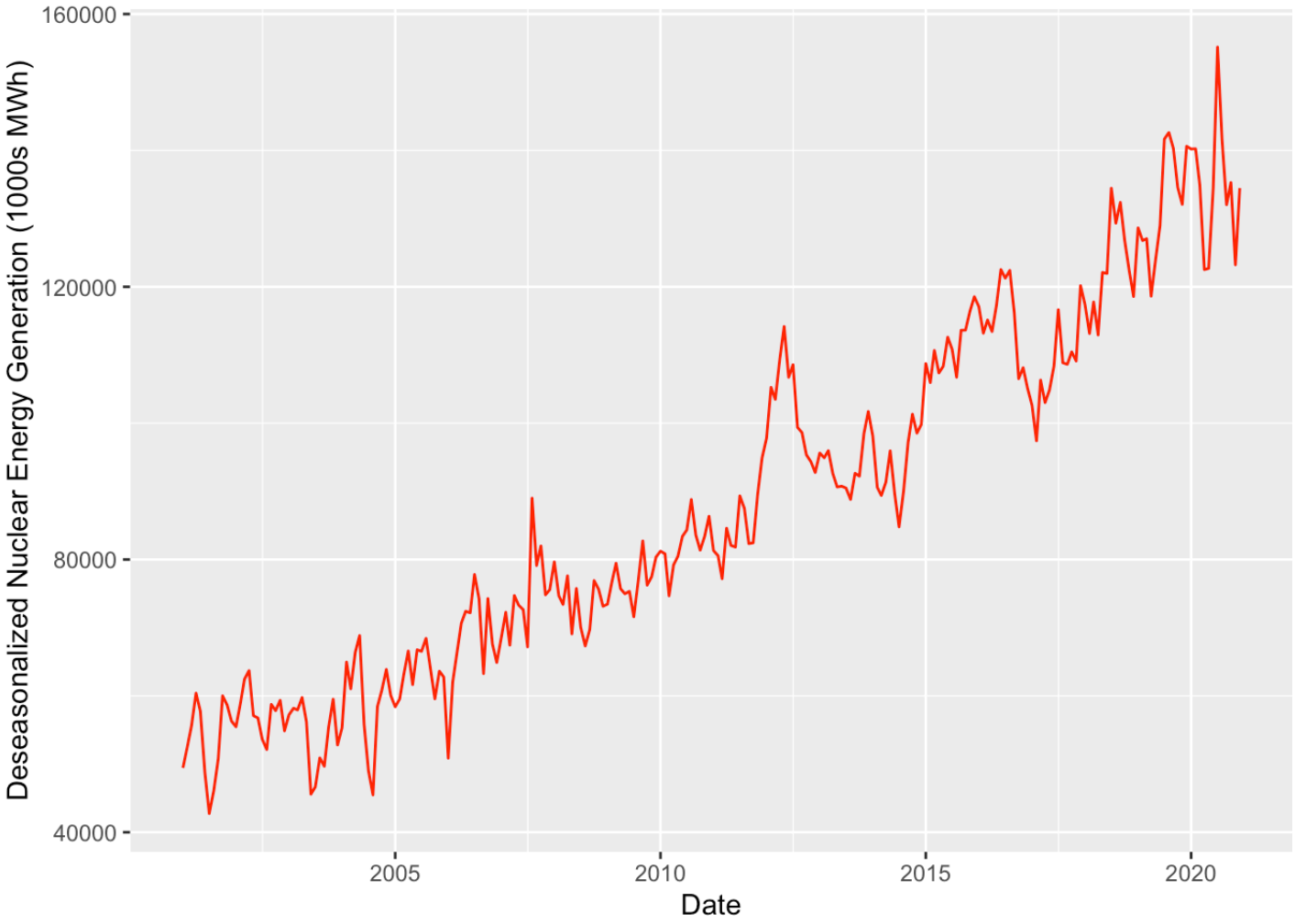
Q2

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.

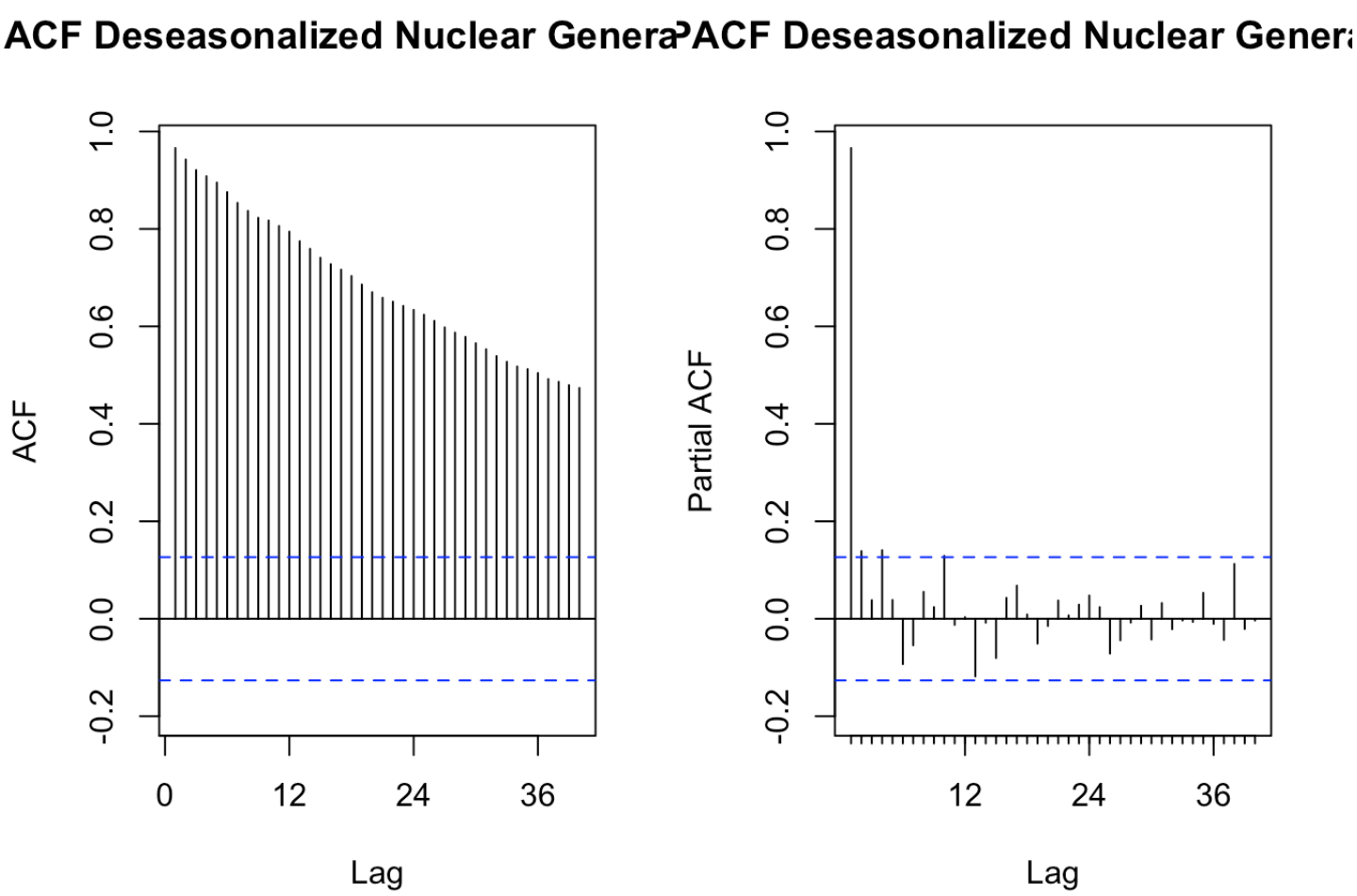
```
#decompose
decompose_natgas<- decompose(ts_natgas)
plot(decompose_natgas)
```




```
#create deseasonalized Nuclear Energy Generation time series
deseasonal_natgas <- seasadj(decompose_natgas)

#df with og series and deseasonalized series
df_natgas <- data.frame(Date = data$Date,
                        natural_gas = data$nuclear.thousand.megawatthours,
                        deseasonal_natural_gas = as.numeric(deseasonal_natgas))

#plot deseasonalized ts over time
ggplot(df_natgas, aes(x= Date , y= deseasonal_natural_gas)) +
  geom_line(color= "red") +
  ylab('Deseasonalized Nuclear Energy Generation (1000s MWh)')
```




```
#ACF & PACF
par(mfrow=c(1,2))
Acf(deseasonal_natgas,lag.max=40, plot=TRUE, main = "ACF Deseasonalized Nuclear Gener
ation")
Pacf(deseasonal_natgas,lag.max=40, plot=TRUE, main = "PACF Deseasonalized Nuclear Gen
eration")
```



There is no longer a seasonal component in the series as shown by the rapid peaks and troughs in the plot over time and in the peaks in the ACF every 12 months. In the deseasonalized plot over time, there is a clear positive trend in the data now.

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.

```
#agumented dickey fuller test (ADF)
#Null hypothesis is that data has a unit root
print("Results for ADF test")
```

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

```
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 = -4.0271, Lag order = 6, p-value = 0.01
## alternative hypothesis: stationary
```

```
#Seasonal Mann-Kendall
SMKtest <- SeasonalMannKendall(deseasonal_natgas)
print("Results for Seasonal Mann Kendall")
```

```
## [1] "Results for Seasonal Mann Kendall"
```

```
print(summary(SMKtest))
```

```
## Score =  2022 , Var(Score) = 11400
## denominator =  2280
## tau = 0.887, 2-sided pvalue =< 2.22e-16
## NULL
```

Answer: The ADF has produces a p-value of 0.01, which means we can reject the null hypothesis in favor of the alternative hypothesis- the data is stationary. The Mann Kendall test produces a p-value ≤ 2.22e-16, we can reject the null in favor of the alternative. There is a trend present in the deseaonalized data.

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: ARIMA(1,0,0) There is a slow decay in the ACF plot and dependency on previous observations in the series plotted against time, so this is an AR process. The sharp cutoff is at lag 1. According to the ADF test, the data is stationary. Thus, the d component is 0. This is not an MA process because there is no slow decay in the PACF plot, and the ACF plot does not have a sharp cutoff.

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(deseasonal_natgas,order=c(1,0,0),include.drift=TRUE)
print(model_100)
```

```
## Series: deseasonal_natgas
## 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
```

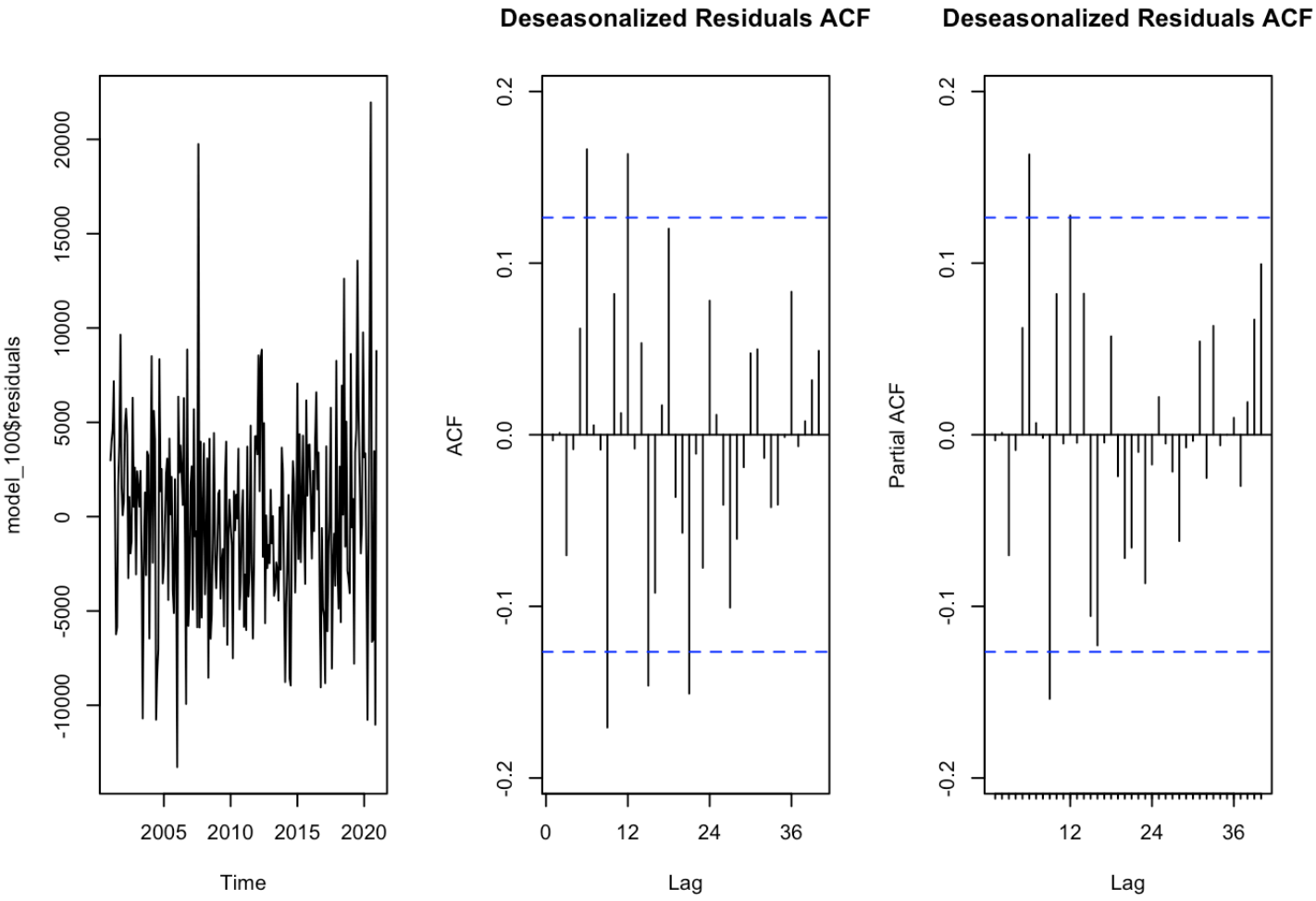
```
#print coefficients
cat("phi =", round(model_100$coef[1],4), "theta =", round(model_100$coef[2],4), "drif
t =", round(model_100$coef[3],4))
```

```
## phi = 0.7182 theta = 44800.49 drift = 359.3965
```

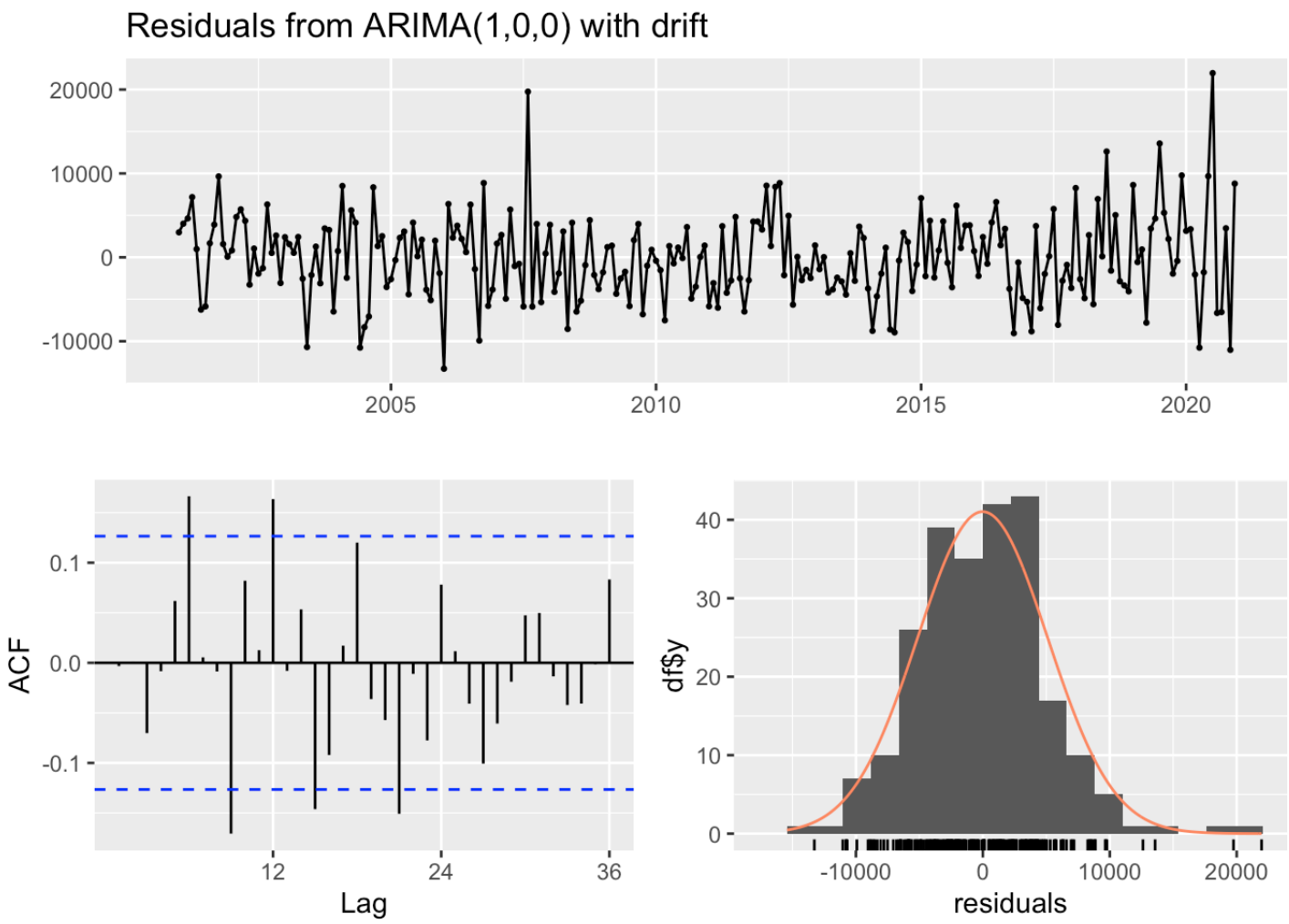
Q6

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?

```
par(mfrow=c(1,3))
ts.plot(model_100$residuals)
Acf(model_100$residuals,lag.max=40, main="Deseasonalized Residuals ACF")
Pacf(model_100$residuals,lag.max=40, main="Deseasonalized Residuals ACF")
```




```
checkresiduals(model_100)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0) with drift
## Q* = 47.775, df = 23, p-value = 0.001787
##
## Model df: 1.   Total lags used: 24
```

Answer: The residual series does not look completely like a white noise series because there are taller peaks in the residuals graph. In addition, the ACF plot has some points outside of the blue lines.

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.

> Answer: ARIMA(1,0,0)(1,1,0)[12] According to the ADF test on the original series, there is stationarity in the series, thus d = 0. This is an AR model- there is slow decay in the ACF plot and cut off at lag one in the PACF plot. Thus, p = 1 and q = 0. Looking at the seasonal lags, there are multiple spikes in the ADF plot and a single spike in the PACF plot. Thus, this is a SAR process and P = 1 and Q = 0. The seasonal pattern is strong and stable over time. Thus, D = 1.

```
#run ADF test
print(adf.test(ts_natgas, alternative = "stationary"))
```

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

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

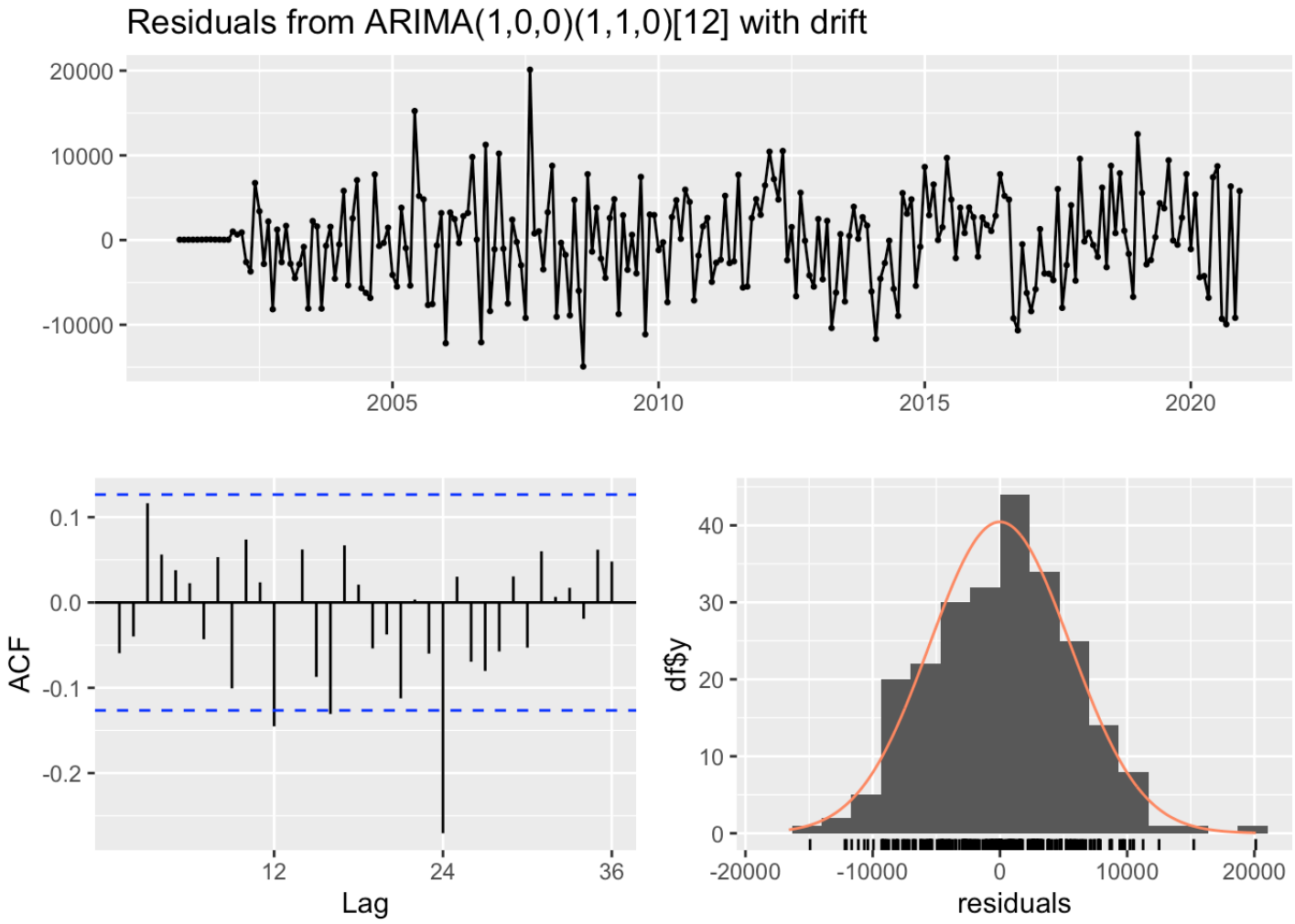
```
#ARIMA
model_100_110<- Arima(ts_natgas,order=c(1,0,0),seasonal = list(order = c(1,1,0), peri
od = 12), include.drift=TRUE)
print(model_100_110)
```

```
## Series: ts_natgas
## ARIMA(1,0,0)(1,1,0)[12] with drift
##
## Coefficients:
##          ar1     sar1     drift
##       0.7646  -0.4542  358.3892
## s.e.  0.0424   0.0593   91.4518
##
## sigma^2 = 32457520:  log likelihood = -2295.5
## AIC=4598.99   AICc=4599.17   BIC=4612.71
```

```
#print coefficients
cat("phi =", round(model_100_110$coef[1],4), "theta =", round(model_100_110$coef[2],4
), "drift =", round(model_100_110$coef[3],4))
```

```
## phi = 0.7646 theta = -0.4542 drift = 358.3892
```

```
#check residuals
checkresiduals(model_100_110)
```



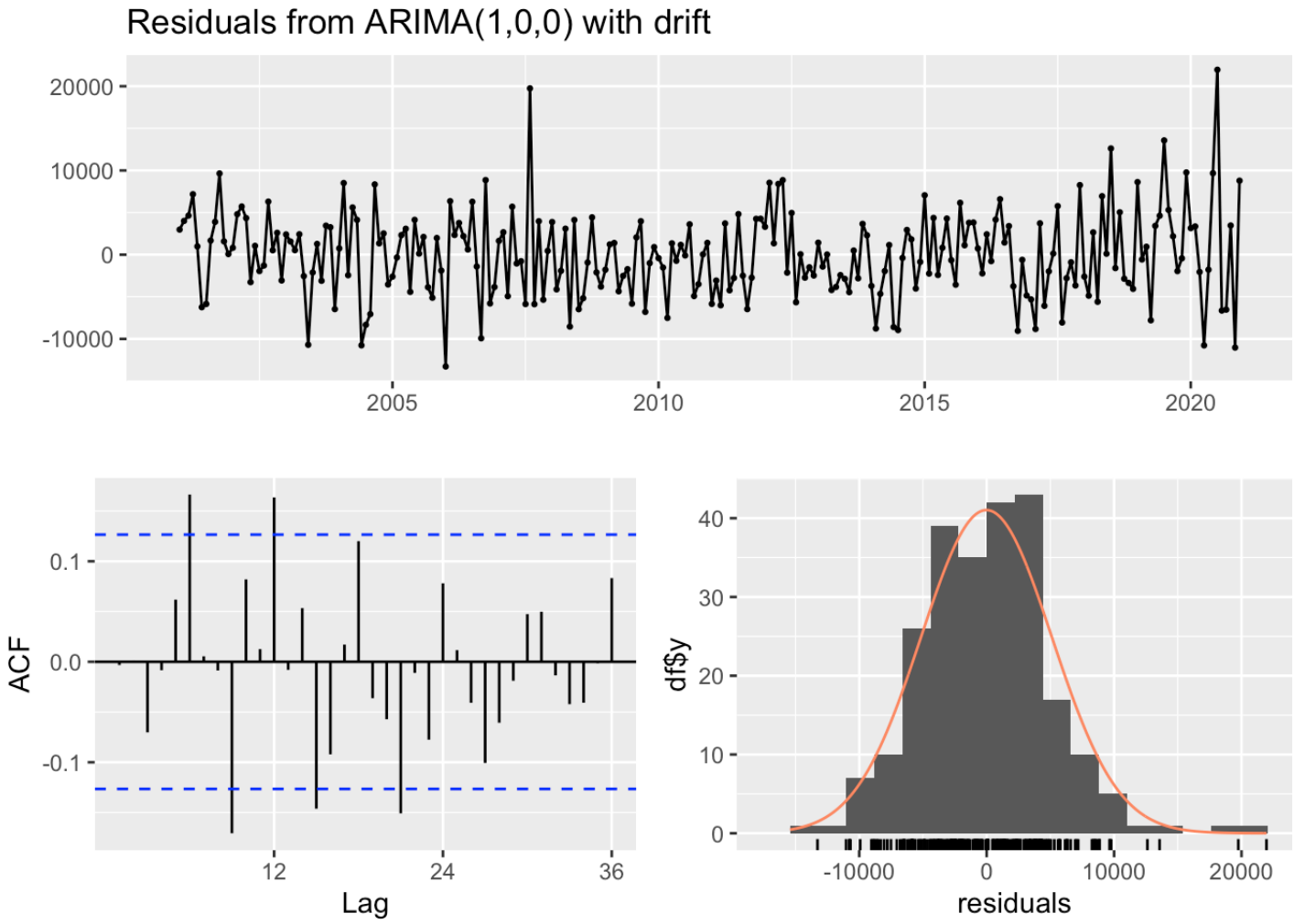

```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0)(1,1,0)[12] with drift
## Q* = 50.251, df = 22, p-value = 0.0005424
##
## Model df: 2.   Total lags used: 24
```

Answer: The residual series does not look completely like a white noise series because there are taller peaks in the residuals graph. In addition, the ACF plot has a few points outside of the blue lines.

Q8

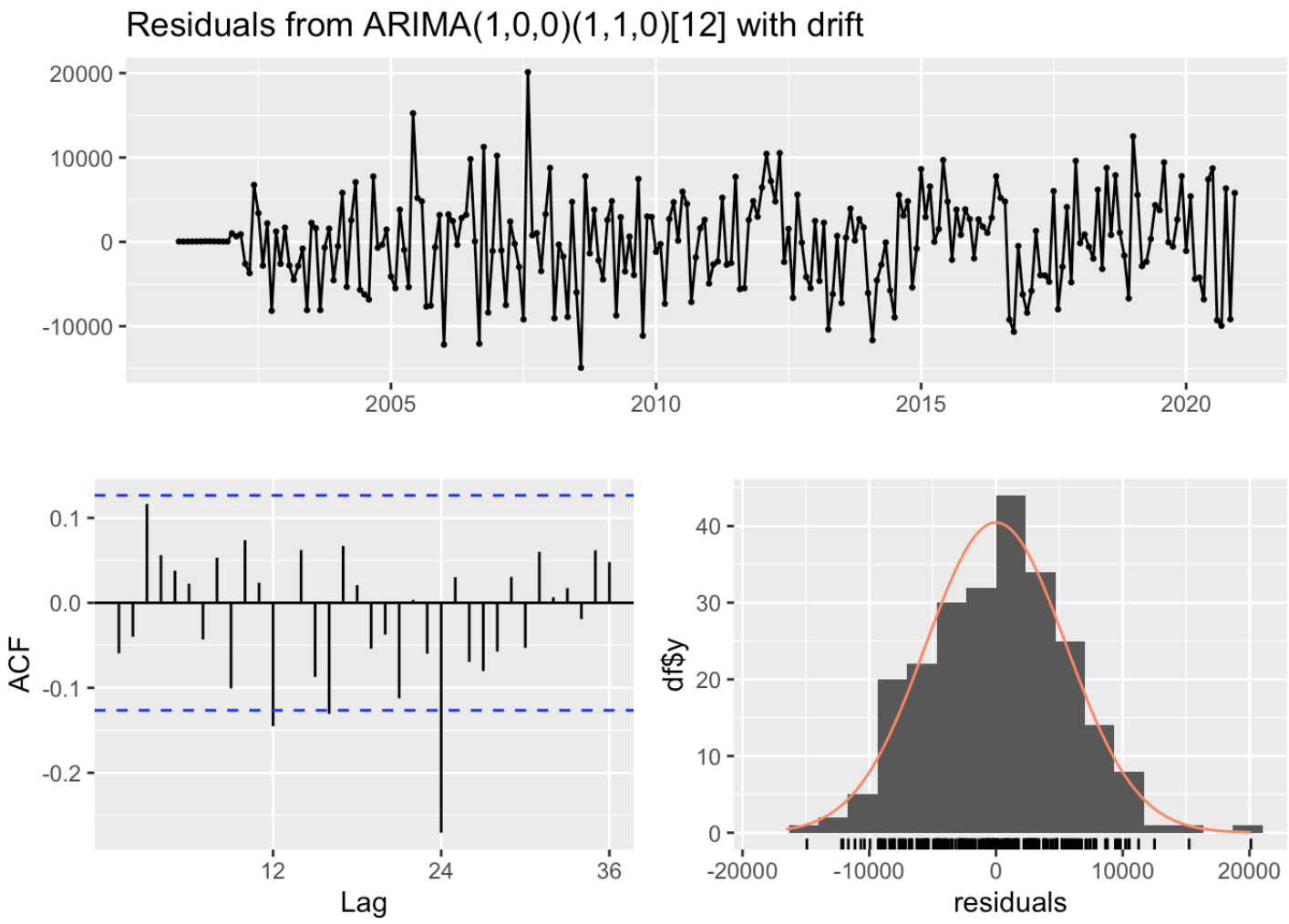
Compare the residual series for Q7 and Q6. Can you tell which ARIMA model is better representing the Natural Gas Series? Is that a fair comparison? Explain your response.

```
checkresiduals(model_100)
```



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0) with drift
## Q* = 47.775, df = 23, p-value = 0.001787
##
## Model df: 1.   Total lags used: 24
```

```
checkresiduals(model_100_110)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0)(1,1,0)[12] with drift
## Q* = 50.251, df = 22, p-value = 0.0005424
##
## Model df: 2.   Total lags used: 24
```

> Answer: It is difficult to tell which ARIMA model better represents the Natural Gas Series. Neither ARIMA model has a perfect white noise series.

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?

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

```
## Series: deseasonal_natgas
## 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 for the deseasonalized series is ARIMA(1,1,1). This model identifies a MA component, which I didn't do.

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?

```
og_natgas_autofit <- auto.arima(ts_natgas)
print(og_natgas_autofit)
```

```
## Series: ts_natgas
## 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: The best ARIMA model for the original series is ARIMA(1,0,0)(0,1,1)[12]. In my model I identified a SAR term instead of an SMA term.