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

Assignment 7 - Due date 03/07/24

Shubhangi Gupta

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
```

```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 errors
```

```r
#install.packages("smooth")
library(smooth)
```

```
## Loading required package: greybox
## Package "greybox", v2.0.0 loaded.
##
##
## Attaching package: 'greybox'
##
## The following object is masked from 'package:tidyr':
##
##     spread
##
## The following object is masked from 'package:lubridate':
##
##     hm
##
## This is package "smooth", v4.0.0
```

```r
library(cowplot)
```

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

```r
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**.

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.

```
#Importing data
USElecGen_Raw <- read.csv(file="Data/Net_generation_United_States_all_sectors_monthly.csv",
                          skip=4, header=TRUE, stringsAsFactors = TRUE)
#Checking imported data
head(USElecGen_Raw)
```

```
##      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
```

```
#Subsetting natural gas
USElecGen_NG <- USElecGen_Raw[,c(1,4)]
#Renaming columns
colnames(USElecGen_NG) <- c("Date", "NG_Gen_MWh")
#Converting the date column into a date object
USElecGen_NG$Date <- my(USElecGen_NG$Date)
```

```r
#Checking subsetted & renamed data
glimpse(USElecGen_NG)
```

```
## Rows: 240
## Columns: 2
## $ Date       <date> 2020-12-01, 2020-11-01, 2020-10-01, 2020-09-01, 2020-08-01~
## $ NG_Gen_MWh <dbl> 125703.7, 109037.2, 131658.2, 141452.7, 173926.6, 185444.8,~
```

```r
#The data is in reverse temporal order so rearranging it.
USElecGen_NG <- USElecGen_NG %>% arrange(Date)
#Checking arranged data
head(USElecGen_NG)
```

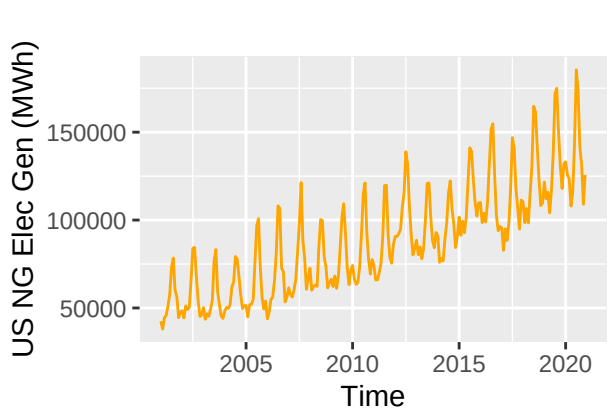
```
##         Date NG_Gen_MWh
## 1 2001-01-01   42388.66
## 2 2001-02-01   37966.93
## 3 2001-03-01   44364.41
## 4 2001-04-01   45842.75
## 5 2001-05-01   50934.21
## 6 2001-06-01   57603.15
```

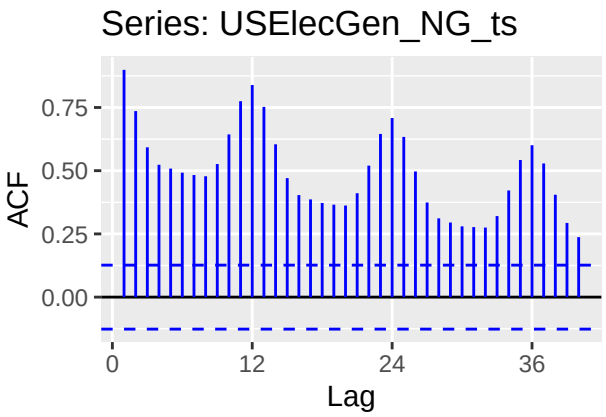
```r
#Converting to time series
USElecGen_NG_ts <- ts(USElecGen_NG$`NG_Gen_MWh`, start=c(2001,01), frequency=12)
#Checking the ts
USElecGen_NG_ts
```

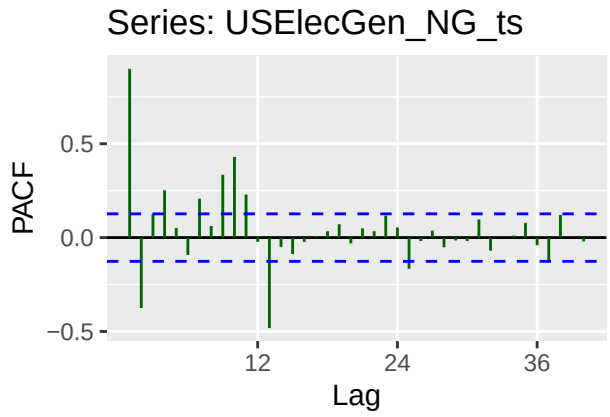
```
##            Jan       Feb       Mar       Apr       May       Jun       Jul
## 2001  42388.66  37966.93  44364.41  45842.75  50934.21  57603.15  73030.14
## 2002  48412.83  44308.43  51214.46  49146.41  50275.24  65631.02  83917.27
## 2003  50175.75  43546.57  46699.03  45195.38  49372.96  54452.95  76938.28
## 2004  48253.37  50319.87  49801.19  51821.99  62021.81  64685.67  79290.02
## 2005  51337.63  44912.62  51896.97  52016.29  54826.00  75635.49  96819.41
## 2006  43806.92  47408.99  54921.74  56090.86  65585.61  81060.21 108093.51
## 2007  61474.53  57622.07  56203.82  60152.85  66469.70  81511.45  97482.68
## 2008  72599.90  60042.01  62170.58  63046.10  62270.45  84619.98 100320.88
## 2009  66390.11  62138.85  68202.74  61158.67  68145.65  84205.18 101893.83
## 2010  74172.74  66198.08  63430.66  64644.03  73665.44  92268.43 114624.20
## 2011  74254.33  65923.98  65947.12  70029.00  75242.80  90691.11 119623.59
## 2012  90760.88  90609.78  92250.65  94828.60 107351.81 115597.50 138862.85
## 2013  88559.05  80283.07  84725.26  78036.43  83815.59  99615.06 120770.98
## 2014  91060.93  75942.31  78150.92  76781.88  89119.68  98467.61 115081.27
## 2015 101687.39  91315.16  99422.72  92806.00 101516.35 121477.75 141119.08
## 2016 110043.75  98552.18 103889.94  98875.95 110430.19 131395.23 151553.87
## 2017  95572.23  82767.65  95073.66  88455.10  98019.31 117235.70 146929.44
## 2018 110292.91  98511.85 106523.78  98371.08 115283.98 130826.40 164749.06
## 2019 121588.68 112141.91 115813.10 104058.59 117058.87 137836.14 171954.56
## 2020 133157.61 125593.92 123696.97 107960.03 115870.92 143245.39 185444.82
##            Aug       Sep       Oct       Nov       Dec
## 2001  78409.80  60181.14  56376.44  44490.62  47540.86
## 2002  84476.87  68161.13  54200.82  45160.88  46100.39
## 2003  83249.69  59089.94  51824.17  45327.92  44034.89
## 2004  77820.69  67853.99  57228.81  49693.03  51309.57
```

```
## 2005 100786.65  73355.48  55940.88  49440.23  53992.60
## 2006 106591.67  72673.35  70640.04  53439.69  56128.19
## 2007 121338.43  88531.61  78358.02  60636.80  66807.83
## 2008  99673.49  79136.03  73283.34  61454.20  64363.65
## 2009 109239.55  92126.62  72602.73  63285.13  71589.61
## 2010 121151.27  93004.18  77738.34  69226.56  77573.32
## 2011 119855.79  91739.07  78819.21  75441.30  86121.63
## 2012 131735.86 108012.30  91725.37  80169.46  83989.10
## 2013 121156.40 102063.23  88587.40  84286.78  92936.29
## 2014 122348.42 106581.59  97683.02  84353.51  91037.82
## 2015 139083.75 123035.92 110005.04 102236.25 109776.68
## 2016 154759.67 125602.74 102897.99  93941.83  96363.60
## 2017 141201.14 118035.85 106826.28  94928.34 111397.79
## 2018 161676.24 141785.53 123142.24 108167.64 109801.98
## 2019 174968.29 149697.03 130947.62 117910.47 131838.92
## 2020 173926.62 141452.67 131658.19 109037.19 125703.65
```

```r
#Plotting the time series, ACF and PACF
USElecGen_NG_ts_plot <- autoplot(USElecGen_NG_ts, col='orange')+ylab('US NG Elec Gen (MWh)')
USElecGen_NG_ts_ACF <- autoplot(Acf(USElecGen_NG_ts, lag.max = 40, type = "correlation",
                                    plot = FALSE), col="blue")
USElecGen_NG_ts_PACF <- autoplot(Pacf(USElecGen_NG_ts, lag.max = 40, type = "correlation",
                                      plot = FALSE), col="darkgreen")
plot_grid(USElecGen_NG_ts_plot, USElecGen_NG_ts_ACF, USElecGen_NG_ts_PACF)
```



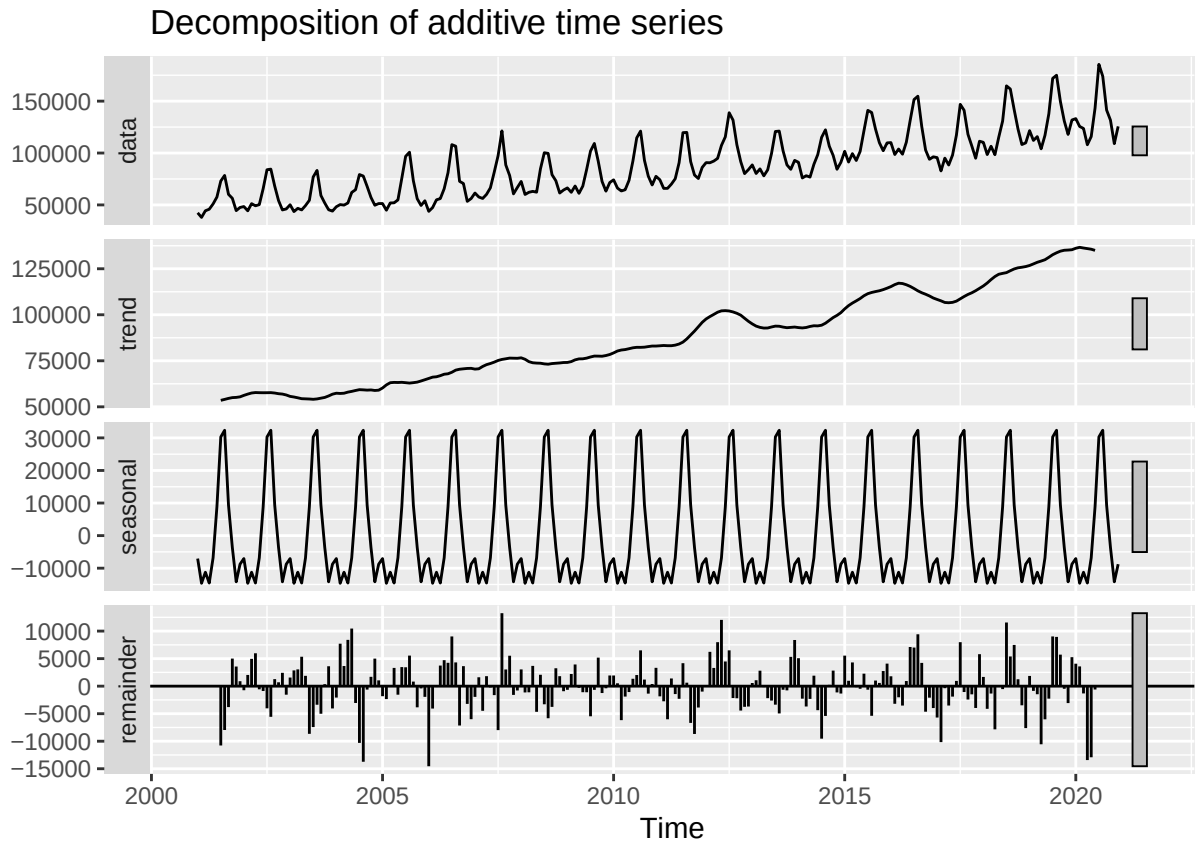




Q2

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.

```
#decomposing the data using decompose()
USElecGen_NG_ts_decompose <- decompose(USElecGen_NG_ts, "additive")
autoplot(USElecGen_NG_ts_decompose)
```

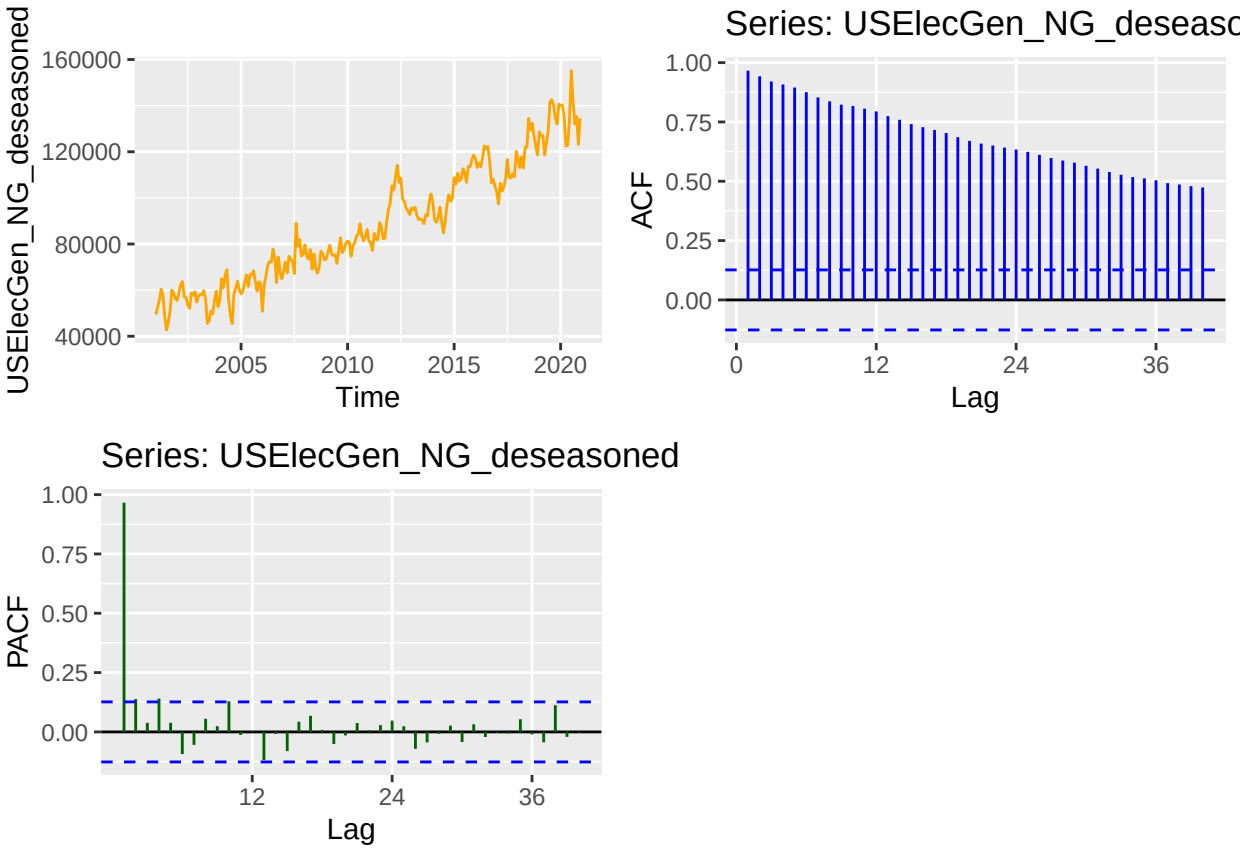



```
#deseseasoning the data
USElecGen_NG_deseasoned <- seasadj(USElecGen_NG_ts_decompose)
USElecGen_NG_deseasoned
```

```
##           Jan       Feb       Mar       Apr       May       Jun       Jul
## 2001 49442.78 52606.62 55607.81 60406.30 57758.76 48734.57 42739.13
## 2002 55466.95 58948.13 62457.86 63709.97 57099.80 56762.44 53626.26
## 2003 57229.86 58186.27 57942.42 59758.94 56197.52 45584.37 46647.27
## 2004 55307.49 64959.57 61044.58 66385.55 68846.37 55817.09 48999.02
## 2005 58391.74 59552.31 63140.36 66579.84 61650.56 66766.91 66528.40
## 2006 50861.04 62048.69 66165.13 70654.41 72410.16 72191.63 77802.51
## 2007 68528.65 72261.77 67447.21 74716.40 73294.26 72642.87 67191.67
## 2008 79654.02 74681.71 73413.97 77609.66 69095.01 75751.39 70029.87
## 2009 73444.23 76778.55 79446.13 75722.23 74970.21 75336.60 71602.83
## 2010 81226.85 80837.78 74674.05 79207.58 80490.00 83399.85 84333.19
```

```
## 2011  81308.45  80563.68  77190.51  84592.56  82067.36  81822.53  89332.58
## 2012  97815.00 105249.48 103494.04 109392.15 114176.37 106728.92 108571.84
## 2013  95613.17  94922.77  95968.65  92599.99  90640.15  90746.48  90479.97
## 2014  98115.05  90582.01  89394.31  91345.44  95944.24  89599.03  84790.26
## 2015 108741.51 105954.86 110666.11 107369.56 108340.91 112609.17 110828.07
## 2016 117097.86 113191.88 115133.33 113439.50 117254.75 122526.65 121262.86
## 2017 102626.35  97407.35 106317.05 103018.66 104843.87 108367.12 116638.43
## 2018 117347.03 113151.55 117767.17 112934.64 122108.54 121957.82 134458.05
## 2019 128642.79 126781.61 127056.49 118622.15 123883.43 128967.56 141663.56
## 2020 140211.72 140233.62 134940.37 122523.59 122695.48 134376.81 155153.81
##            Aug       Sep       Oct       Nov       Dec
## 2001  46064.29  50773.19  60007.05  58675.50  56313.11
## 2002  52131.36  58753.18  57831.43  59345.75  54872.64
## 2003  50904.18  49681.99  55454.78  59512.79  52807.14
## 2004  45475.18  58446.04  60859.43  63877.91  60081.81
## 2005  68441.14  63947.53  59571.49  63625.10  62764.85
## 2006  74246.15  63265.39  74270.65  67624.57  64900.43
## 2007  88992.91  79123.66  81988.64  74821.67  75580.08
## 2008  67327.98  69728.08  76913.95  75639.07  73135.89
## 2009  76894.04  82718.66  76233.34  77470.01  80361.85
## 2010  88805.76  83596.22  81368.95  83411.43  86345.57
## 2011  87510.28  82331.12  82449.82  89626.17  94893.87
## 2012  99390.35  98604.35  95355.98  94354.34  92761.34
## 2013  88810.89  92655.28  92218.01  98471.66 101708.54
## 2014  90002.91  97173.63 101313.63  98538.39  99810.07
## 2015 106738.24 113627.97 113635.65 116421.13 118548.93
## 2016 122414.16 116194.79 106528.60 108126.71 105135.84
## 2017 108855.63 108627.90 110456.89 109113.22 120170.04
## 2018 129330.73 132377.57 126772.85 122352.51 118574.22
## 2019 142622.77 140289.08 134578.23 132095.34 140611.16
## 2020 141581.11 132044.71 135288.80 123222.06 134475.90
```

```
USElecGen_NG_deseasoned_plot <- autoplot(USElecGen_NG_deseasoned, col="orange")
USElecGen_NG_deseasoned_Acf <- autoplot(Acf(USElecGen_NG_deseasoned, lag.max=40,
                                            type = "correlation", plot=FALSE), col="blue")
USElecGen_NG_deseasoned_Pacf <- autoplot(Pacf(USElecGen_NG_deseasoned, lag.max=40,
                                              type = "correlation", plot=FALSE), col="darkgreen")
plot_grid(USElecGen_NG_deseasoned_plot, USElecGen_NG_deseasoned_Acf, USElecGen_NG_deseasoned_Pacf)
```



Analysis: The series plot (orange) clearly shows that the seasonal component has been removed. This is corroborated by the ACF as the original plot shows a seasonal component whereas the deseasoned ACF is declining over time, with no seasonality. The PACF is also more clean, as fluctuations at later lags are gone and all but the first lag are now insignificant. The plot now only showcases a rising trend over time.

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.

```
#ADF test to check for stochasticity
print(adf.test(USElecGen_NG_deseasoned), alternative="stationary")
```

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

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

```r
#Mann Kedndall test to check for a deterministic trend.
print(summary(MannKendall(USElecGen_NG_deseasoned)))
```

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

Analysis: The ADF test gave a p-value < 0.05 meaning we reject the null hypothesis and don't reject the alternative hypothesis ie the the data is possibly stationary and thus not stochastic. The Mann Kendall test also gives a significant result since the p-value < 0.05 so we reject the null hypothesis and don't reject the alternative, indicating that the data has a deterministic 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.

```r
ndiffs(USElecGen_NG_deseasoned)
```

```
## [1] 1
```

Analysis: The answer from qn 3 indicates that the series is not stationary, and so it will have to be differenced to achieved stationarity. Thus d=1. This is corroborated by the ndiffs() result. Next, the ACF from Qn 2 shows a declining trend over time and PACF is only significant at lag 1, indicating the need for an AR component. Since there is only 1 significant PACF spike, the order of the AR model would be 1. Thus p=1. Since PACF is not decaying over time, I assume that there is no MA component. So q=0. Thus, I would estimate the model of best fit to be ARIMA(1,1,0)

Q5

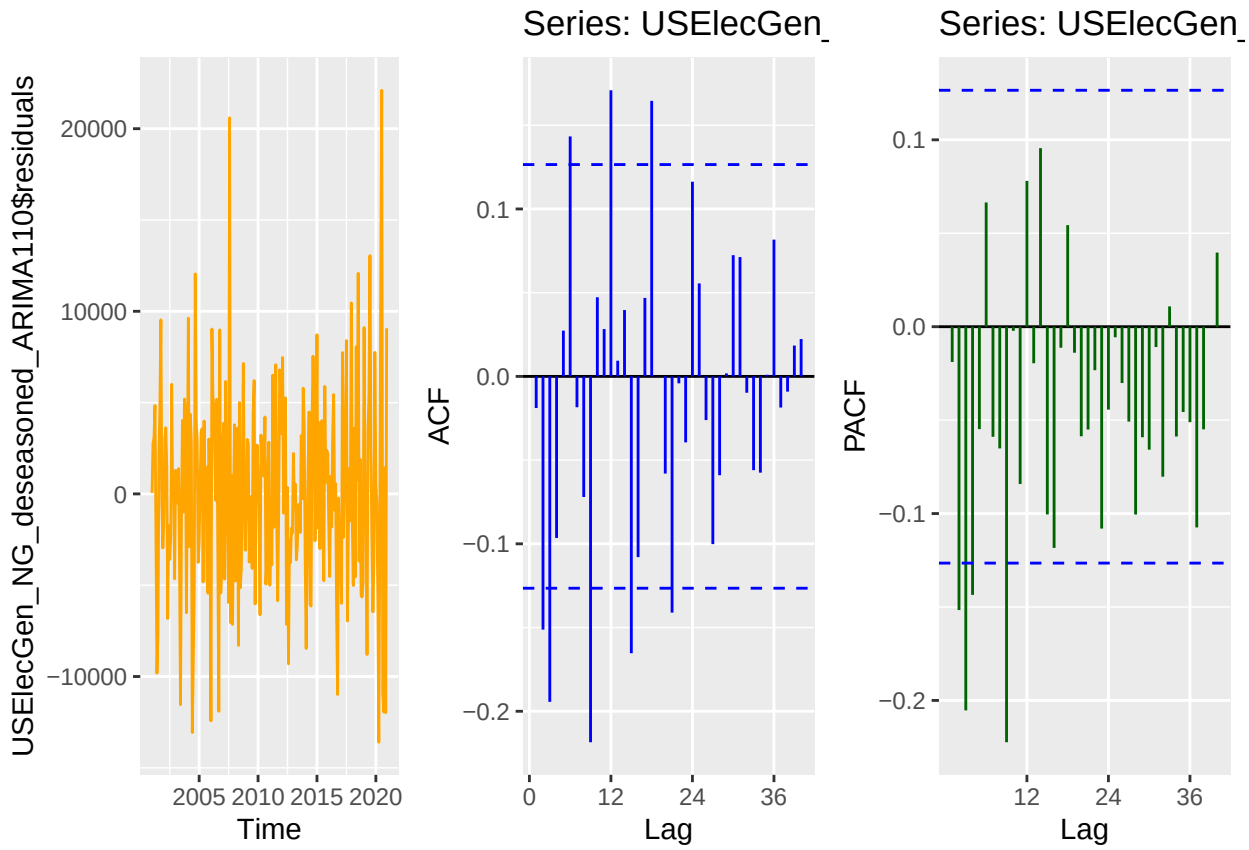
Use `Arima()` from package "forecast" to fit an ARIMA model to your series considering the order estimated in Q4. You should allow constants in the model, i.e., `include.mean = TRUE` or `include.drift=TRUE`. **Print the coefficients** in your report. Hint: use the `cat()` r `print()` function to print.

```r
#Starting with my assumption of the model ARIMA (1,1,0)
USElecGen_NG_deseasoned_ARIMA110 <- Arima(USElecGen_NG_deseasoned, order=c(1,1,0),
                                          include.mean=TRUE, include.drift = TRUE)
print(summary(USElecGen_NG_deseasoned_ARIMA110))
```

```
## Series: USElecGen_NG_deseasoned
## ARIMA(1,1,0) with drift
##
## Coefficients:
##           ar1     drift
##       -0.1479  348.3927
## s.e.   0.0644  308.8385
```

```
##
## sigma^2 = 30254066:  log likelihood = -2396.54
## AIC=4799.07   AICc=4799.18   BIC=4809.5
##
## Training set error measures:
##                     ME     RMSE      MAE        MPE     MAPE      MASE
## Training set 1.809377 5465.884 4234.794 -0.2842222 5.174439 0.5177375
##                      ACF1
## Training set -0.01888781
```

```
plot_grid(autoplot(USElecGen_NG_deseasoned_ARIMA110$residuals, col="orange"),
  autoplot(Acf(USElecGen_NG_deseasoned_ARIMA110$residuals,lag.max=40, plot = FALSE), col="blue"),
  autoplot(Pacf(USElecGen_NG_deseasoned_ARIMA110$residuals,lag.max=40, plot = FALSE), col="darkgreen"),
  nrow=1)
```

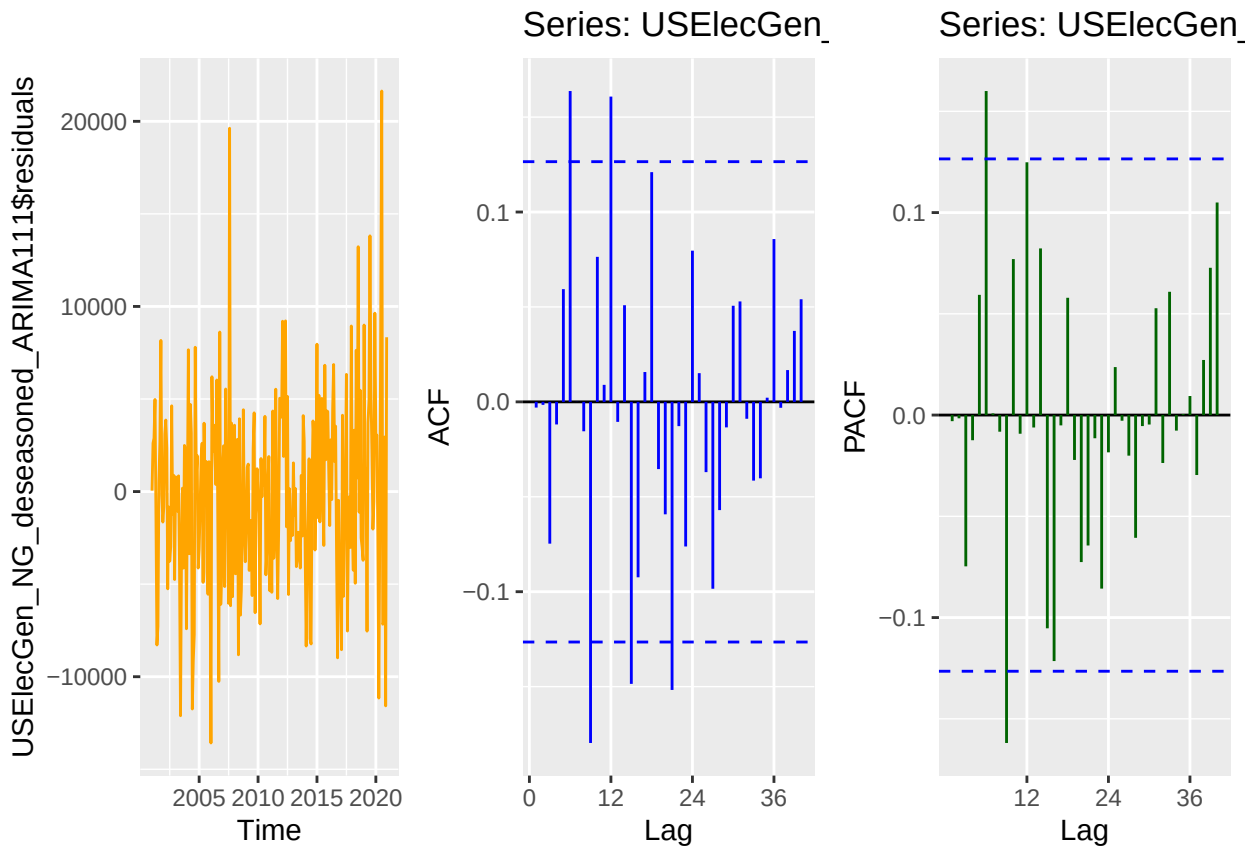



```
#Adding an MA component
USElecGen_NG_deseasoned_ARIMA111 <- Arima(USElecGen_NG_deseasoned, order=c(1,1,1),
                                          include.mean=TRUE, include.drift = TRUE)
print(summary(USElecGen_NG_deseasoned_ARIMA111))
```

```
## Series: USElecGen_NG_deseasoned
## 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
##
## Training set error measures:
##                      ME     RMSE     MAE        MPE     MAPE      MASE
## Training set -141.3123 5150.819 3984.38 -0.7171368 4.850437 0.4871225
##                       ACF1
## Training set -0.003014461
```

```
plot_grid(autoplot(USElecGen_NG_deseasoned_ARIMA111$residuals, col="orange"),
  autoplot(Acf(USElecGen_NG_deseasoned_ARIMA111$residuals,lag.max=40, plot = FALSE), col="blue"),
  autoplot(Pacf(USElecGen_NG_deseasoned_ARIMA111$residuals,lag.max=40, plot = FALSE), col="darkgreen"),
  nrow=1)
```

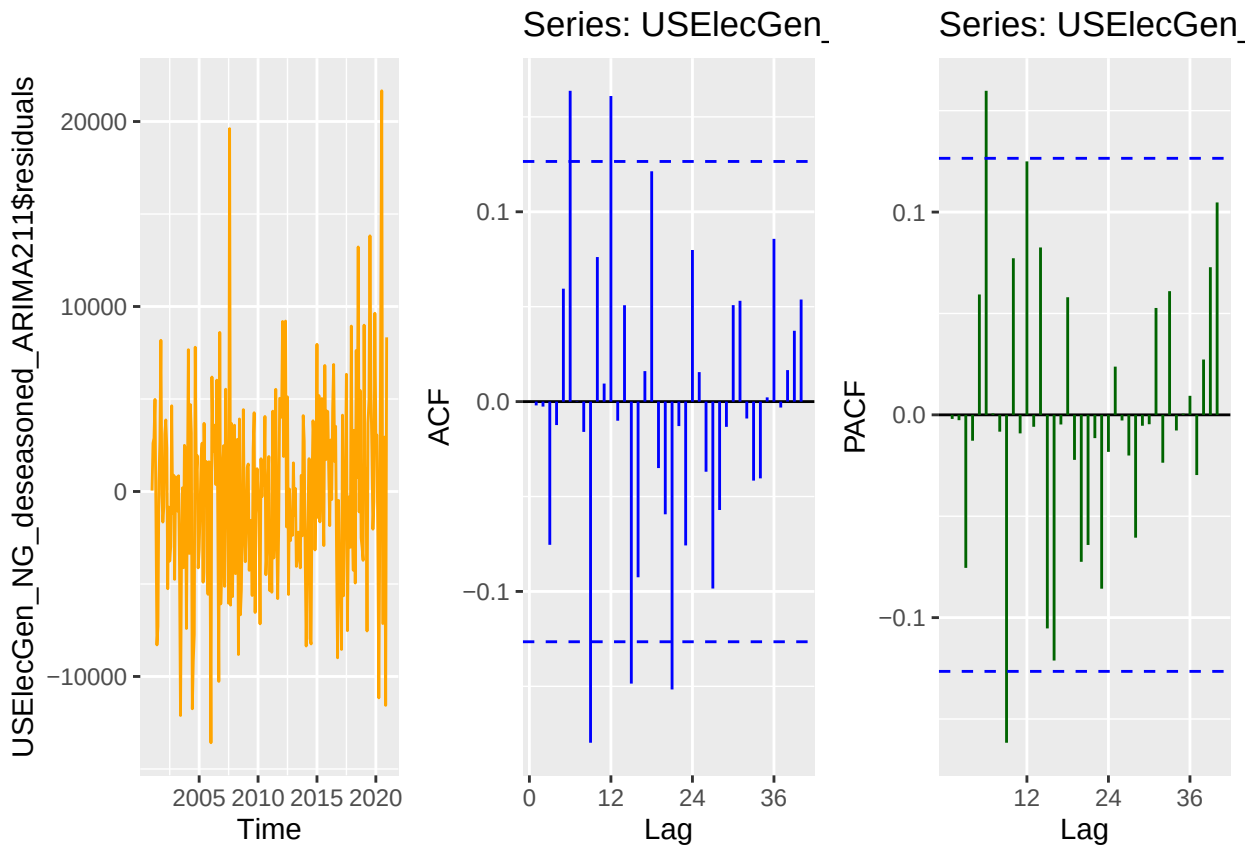



```
#Adding a second AR component since there is still some autocorrelation
USElecGen_NG_deseasoned_ARIMA211 <- Arima(USElecGen_NG_deseasoned, order=c(2,1,1),
                                          include.mean=TRUE, include.drift = TRUE)
print(summary(USElecGen_NG_deseasoned_ARIMA211))
```

```
## Series: USElecGen_NG_deseasoned
## ARIMA(2,1,1) with drift
##
```

```
## Coefficients:
##          ar1     ar2      ma1     drift
##       0.7057  0.0017  -0.9798  359.4921
## s.e.  0.0710  0.0707   0.0360   29.3046
##
## sigma^2 = 27094287:  log likelihood = -2383.11
## AIC=4776.21   AICc=4776.47   BIC=4793.59
##
## Training set error measures:
##                       ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -143.3404 5150.711 3984.167 -0.7194801 4.850628 0.4870963
##                       ACF1
## Training set -0.001995024
```

```
plot_grid(autoplot(USElecGen_NG_deseasoned_ARIMA211$residuals, col="orange"),
  autoplot(Acf(USElecGen_NG_deseasoned_ARIMA211$residuals,lag.max=40, plot = FALSE), col="blue"),
  autoplot(Pacf(USElecGen_NG_deseasoned_ARIMA211$residuals,lag.max=40, plot = FALSE), col="darkgreen"),
  nrow=1)
```

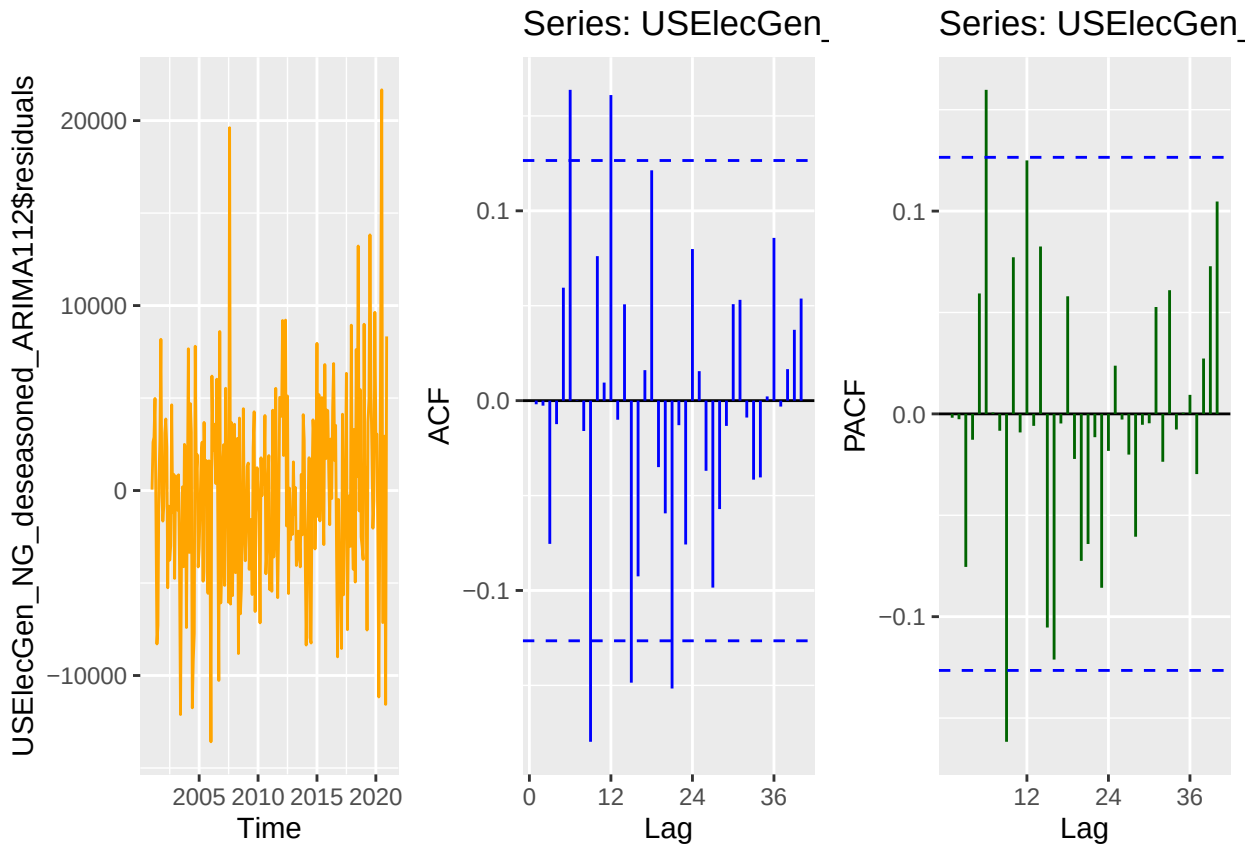



```
#The second AR component increased the AIC so removing that and adding a second MA component
USElecGen_NG_deseasoned_ARIMA112 <- Arima(USElecGen_NG_deseasoned, order=c(1,1,2),
                                          include.mean=TRUE, include.drift = TRUE)
print(summary(USElecGen_NG_deseasoned_ARIMA112))
```

```
## Series: USElecGen_NG_deseasoned
```

```
## ARIMA(1,1,2) with drift
##
## Coefficients:
##          ar1      ma1     ma2     drift
##       0.7081  -0.9823  0.0025  359.4980
## s.e.  0.0939   0.1189  0.0982   29.3122
##
## sigma^2 = 27094333:  log likelihood = -2383.11
## AIC=4776.21   AICc=4776.47   BIC=4793.59
##
## Training set error measures:
##                      ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -143.3387 5150.715 3984.161 -0.7195044 4.850627 0.4870957
##                       ACF1
## Training set -0.001899017
```

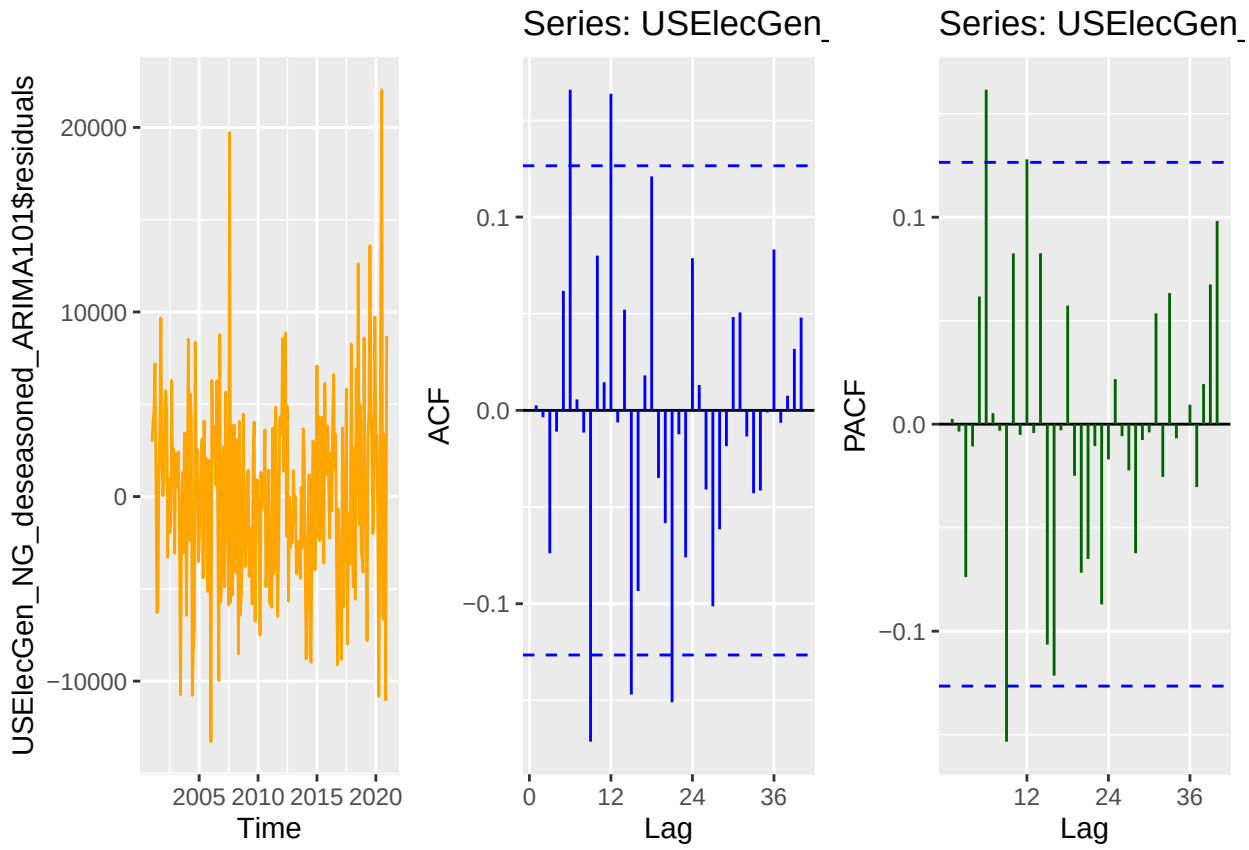
```
plot_grid(autoplot(USElecGen_NG_deseasoned_ARIMA112$residuals, col="orange"),
  autoplot(Acf(USElecGen_NG_deseasoned_ARIMA112$residuals,lag.max=40, plot = FALSE), col="blue"),
  autoplot(Pacf(USElecGen_NG_deseasoned_ARIMA112$residuals,lag.max=40, plot = FALSE), col="darkgreen"),
  nrow=1)
```




```
#Checking how this would fair without differencing
USElecGen_NG_deseasoned_ARIMA101 <- Arima(USElecGen_NG_deseasoned, order=c(1,0,1),
                                          include.mean=TRUE, include.drift = TRUE)
print(summary(USElecGen_NG_deseasoned_ARIMA101))
```

```
## Series: USElecGen_NG_deseasoned
## ARIMA(1,0,1) with drift
##
## Coefficients:
##          ar1      ma1  intercept     drift
##       0.7237  -0.0115  44807.456  359.3516
## s.e.  0.0620   0.0916   2314.027   16.5570
##
## sigma^2 = 26742051:  log likelihood = -2391.1
## AIC=4792.2   AICc=4792.46   BIC=4809.6
##
## Training set error measures:
##                       ME     RMSE      MAE        MPE    MAPE      MASE
## Training set -18.20524 5127.997 4022.603 -0.3492687 4.91179 0.4917955
##                     ACF1
## Training set 0.00255897
```

```
plot_grid(autoplot(USElecGen_NG_deseasoned_ARIMA101$residuals, col="orange"),
  autoplot(Acf(USElecGen_NG_deseasoned_ARIMA101$residuals,lag.max=40, plot = FALSE), col="blue"),
  autoplot(Pacf(USElecGen_NG_deseasoned_ARIMA101$residuals,lag.max=40, plot = FALSE), col="darkgreen"),
  nrow=1)
```




```
Comparing_AICs <- data.frame(USElecGen_NG_deseasoned_ARIMA110$aic,
                             USElecGen_NG_deseasoned_ARIMA111$aic,
                             USElecGen_NG_deseasoned_ARIMA211$aic,
```

```
                              USElecGen_NG_deseasoned_ARIMA112$aic,
                              USElecGen_NG_deseasoned_ARIMA101$aic)
Comparing_AICs
```

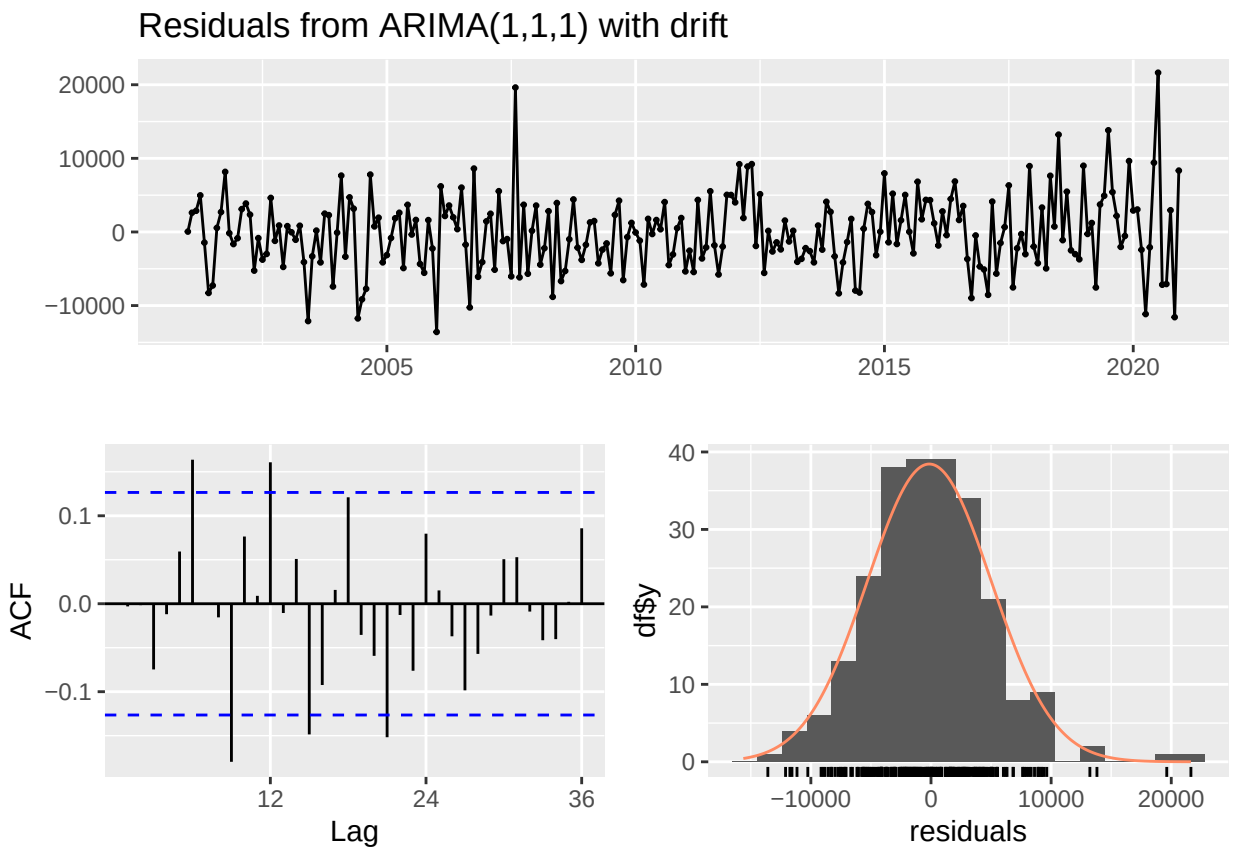
```
##   USElecGen_NG_deseasoned_ARIMA110.aic USElecGen_NG_deseasoned_ARIMA111.aic
## 1                              4799.075                              4774.213
##   USElecGen_NG_deseasoned_ARIMA211.aic USElecGen_NG_deseasoned_ARIMA112.aic
## 1                              4776.212                              4776.212
##   USElecGen_NG_deseasoned_ARIMA101.aic
## 1                              4792.202
```

Analysis: The best model fit is ARIMA (1,1,1) as it has the lowest AIC. I began by testing ARIMA (1,1,0) as the original deseseasoned data had a declining ACF over time indicating that an AR component would be a good fit. However the result did have some significant spikes in the ACF in initial years, indicating the possible need for an MA component. Thus I added that and tested ARIMA (1,1,1) next which did take out those significant ACF spikes leaving only one at lag 12 which can be ignored. I also tested ARIMA (2,1,1) and ARIMA (1,1,2) to see if an additional AR or MA component would make it more robust, but the AIC increased and there was no significant change in the ACF and PACF plots so those components are not required. I also tested the model without differencing ARIMA(1,0,1) which give the highest AIC of all, indicating that differencing is required as the series is not stationary. Thus the correct model fit is ARIMA (1,1,1).

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 the best fitted model ARIMA(1,1,1)
checkresiduals(USElecGen_NG_deseasoned_ARIMA111)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,1) with drift
## Q* = 48.356, df = 22, p-value = 0.0009736
##
## Model df: 2.   Total lags used: 24
```

Analysis: Yes, the residuals almost look like white noise, apart from a couple of points that could be outliers.This is supported by the fact that the residuals are fitted by a normal distribution indicating that they are iid.

Modeling the original series (with seasonality)

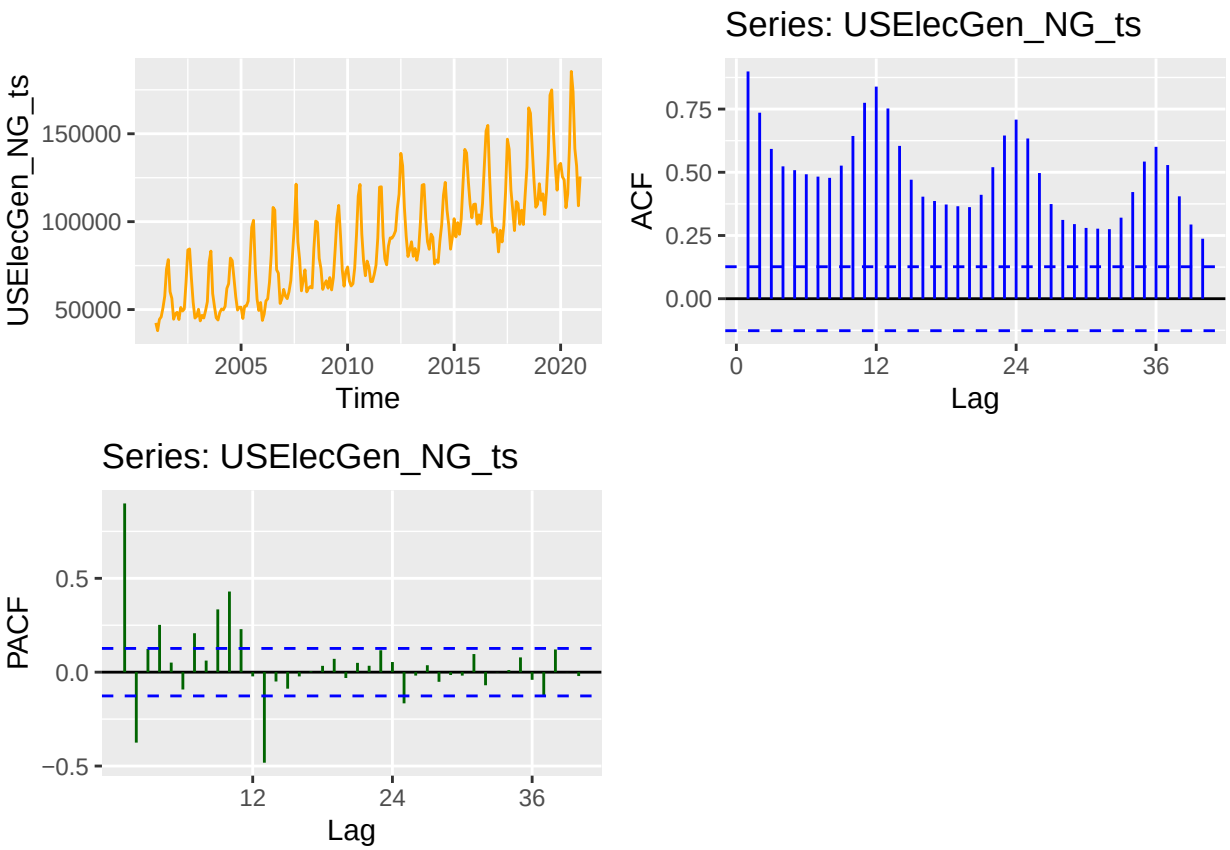
Q7

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.

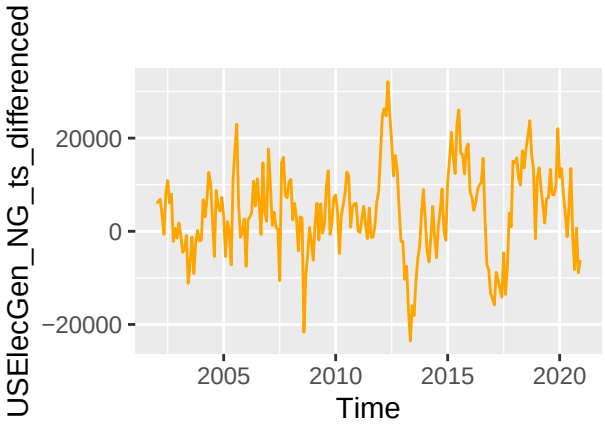
```
#Checking if seasonal differencing is required using nsdiffs()
nsdiffs(USElecGen_NG_ts)
```

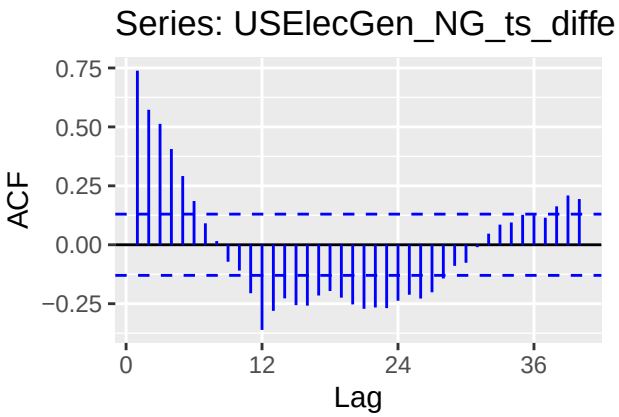
```
## [1] 1
```

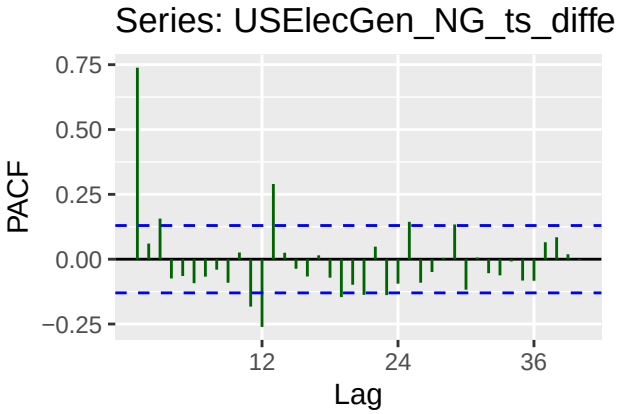
```r
#Checking ACF and PACF of original data
plot_grid(
  autoplot(USElecGen_NG_ts, col="orange"),
  autoplot(Acf(USElecGen_NG_ts, lag.max=40, plot=FALSE), col="blue"),
  autoplot(Pacf(USElecGen_NG_ts, lag.max=40, plot=FALSE), col="darkgreen"))
```




```r
#The ACF and PACF show a seasonal trend at lag 12, so differencing that
USElecGen_NG_ts_differenced <- diff(USElecGen_NG_ts, lag=12, differences = 1)
plot_grid(autoplot(USElecGen_NG_ts_differenced, col="orange"),
          autoplot(Acf(USElecGen_NG_ts_differenced, lag.max=40, plot=FALSE),col="blue"),
          autoplot(Pacf(USElecGen_NG_ts_differenced, lag.max=40, plot=FALSE), col="darkgreen"))
```



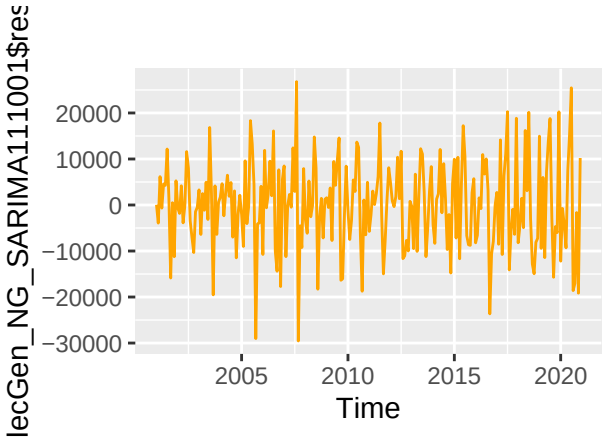


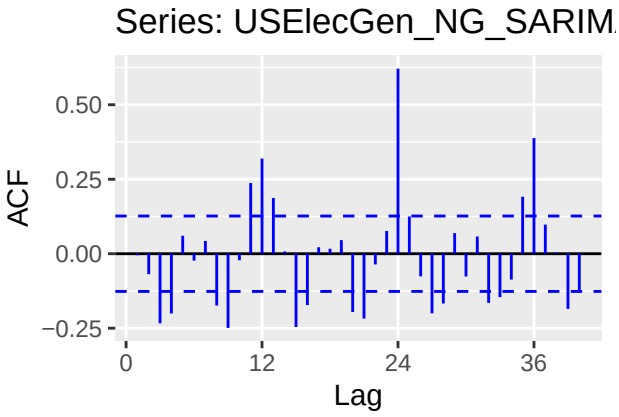


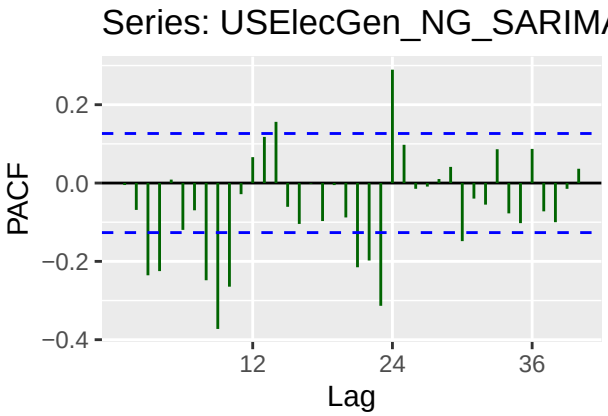

```
#To fit the SARIMA model, considering p=1, d=1, q=1 since that was the best fit result
#in the previous questions. For the seasonal component, since P+Q <=1, starting with Q=1,
#since the autocorrelation at the seasonal period is negative
USElecGen_NG_SARIMA111001 <- Arima(USElecGen_NG_ts, order=c(1,1,1), seasonal=c(0,0,1), include.drift=TRU
print(summary(USElecGen_NG_SARIMA111001))
```

```
## Series: USElecGen_NG_ts
## ARIMA(1,1,1)(0,0,1)[12] with drift
##
## Coefficients:
##           ar1     ma1    sma1      drift
##       -0.1571  0.4116  0.5880   285.6343
## s.e.   0.2678  0.2475  0.0444  1181.6771
##
## sigma^2 = 93887447:  log likelihood = -2533.43
## AIC=5076.86   AICc=5077.11   BIC=5094.24
##
## Training set error measures:
##                     ME     RMSE      MAE        MPE    MAPE      MASE
## Training set 8.412678 9588.089 7523.902 -0.4688216 8.66903 0.9198574
##                      ACF1
## Training set -0.005039261
```

```
plot_grid(autoplot(USElecGen_NG_SARIMA111001$residuals, col="orange"), autoplot(Acf(USElecGen_NG_SARIMA
```







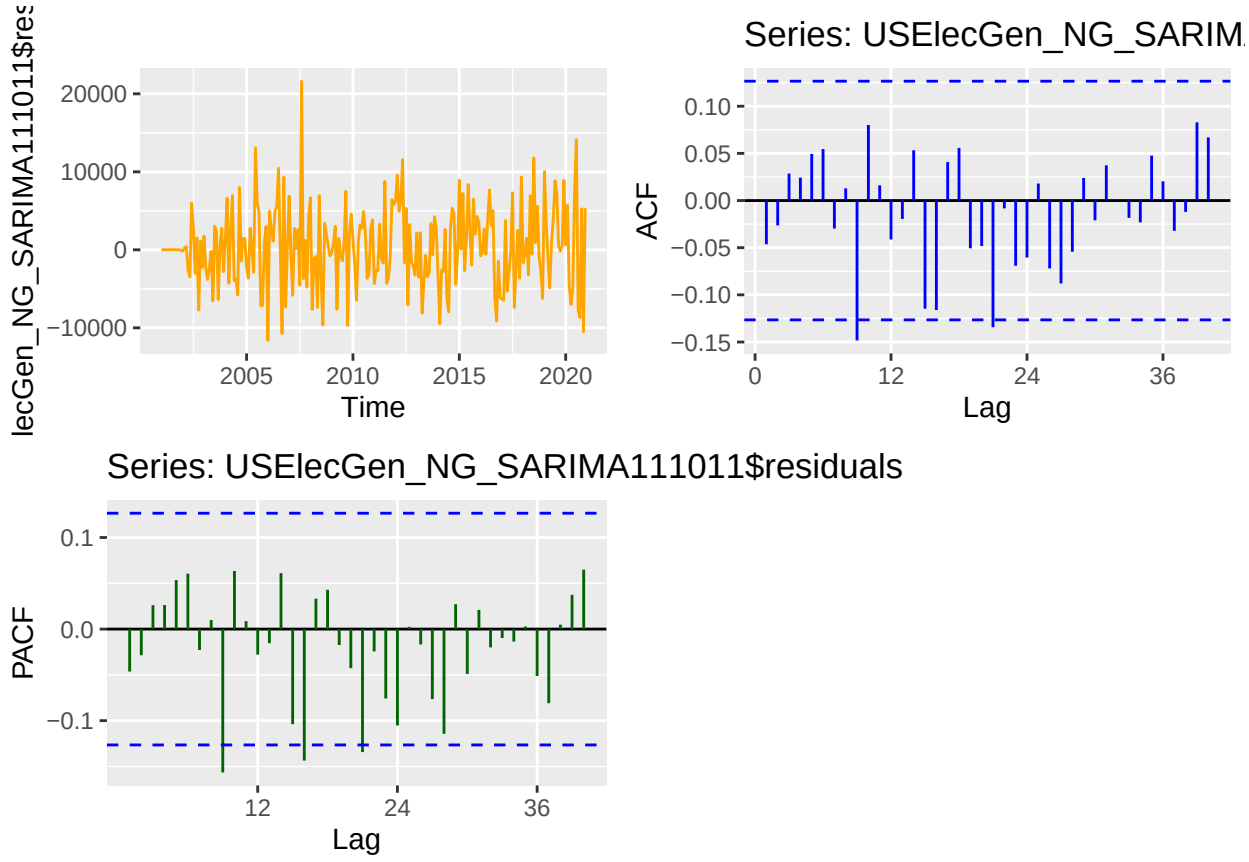

```
#The ACF still shows seasonal spikes so adding a D=1 component to difference that

USElecGen_NG_SARIMA111011 <- Arima(USElecGen_NG_ts, order=c(1,1,1), seasonal=c(0,1,1),
                                   include.drift=TRUE)
```

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

```
print(summary(USElecGen_NG_SARIMA111011))
```

```
## Series: USElecGen_NG_ts
## ARIMA(1,1,1)(0,1,1)[12]
##
## Coefficients:
##          ar1      ma1     sma1
##       0.7323  -0.9819  -0.7017
## s.e.  0.0504   0.0183   0.0563
##
## sigma^2 = 27922085:  log likelihood = -2272.2
## AIC=4552.39   AICc=4552.57   BIC=4566.09
##
## Training set error measures:
##                     ME    RMSE      MAE        MPE    MAPE      MASE        ACF1
## Training set 388.6732 5104.96 3966.553 0.02702438 4.66492 0.4849429 -0.04631823
```

```
plot_grid(autoplot(USElecGen_NG_SARIMA111011$residuals, col="orange"),
          autoplot(Acf(USElecGen_NG_SARIMA111011$residuals, lag.max=40, plot=FALSE),col="blue"),
          autoplot(Pacf(USElecGen_NG_SARIMA111011$residuals, lag.max=40, plot=FALSE), col="darkgreen"))
```




```
#The ACF is now insignificant!

Comparing_AICs_Seasonal <- data.frame(USElecGen_NG_SARIMA111001$aic, USElecGen_NG_SARIMA111011$aic)
Comparing_AICs_Seasonal
```

```
##   USElecGen_NG_SARIMA111001.aic USElecGen_NG_SARIMA111011.aic
## 1                      5076.856                      4552.393
```

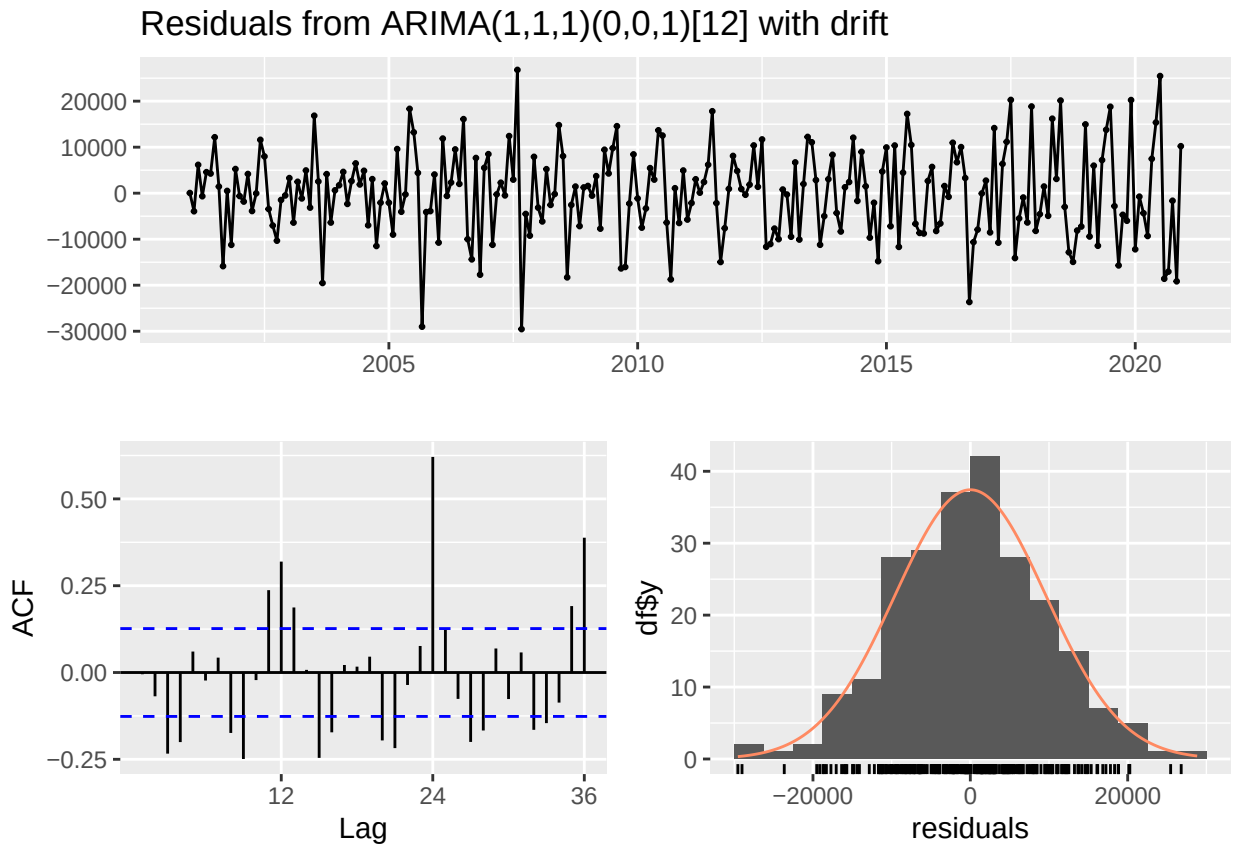
```
#The correct answer is SARIMA(1,1,1)(0,1,1).
```

Analysis: Differencing the original data at lag 12 indicated an ACF with many significant negative spikes, indicating the need for an SMA. Thus I ran a SARIMA(1,1,1)(0,0,1) which led to an ACF plot that still had some seasonal lags. Thus I tried adding a differencing component D=1 and ran SARIMA(1,1,1)(0,1,1) which differenced out that seasonality and yielded a lower AIC. Thus the answer is SARIMA(1,1,1)(0,1,1)

Q8

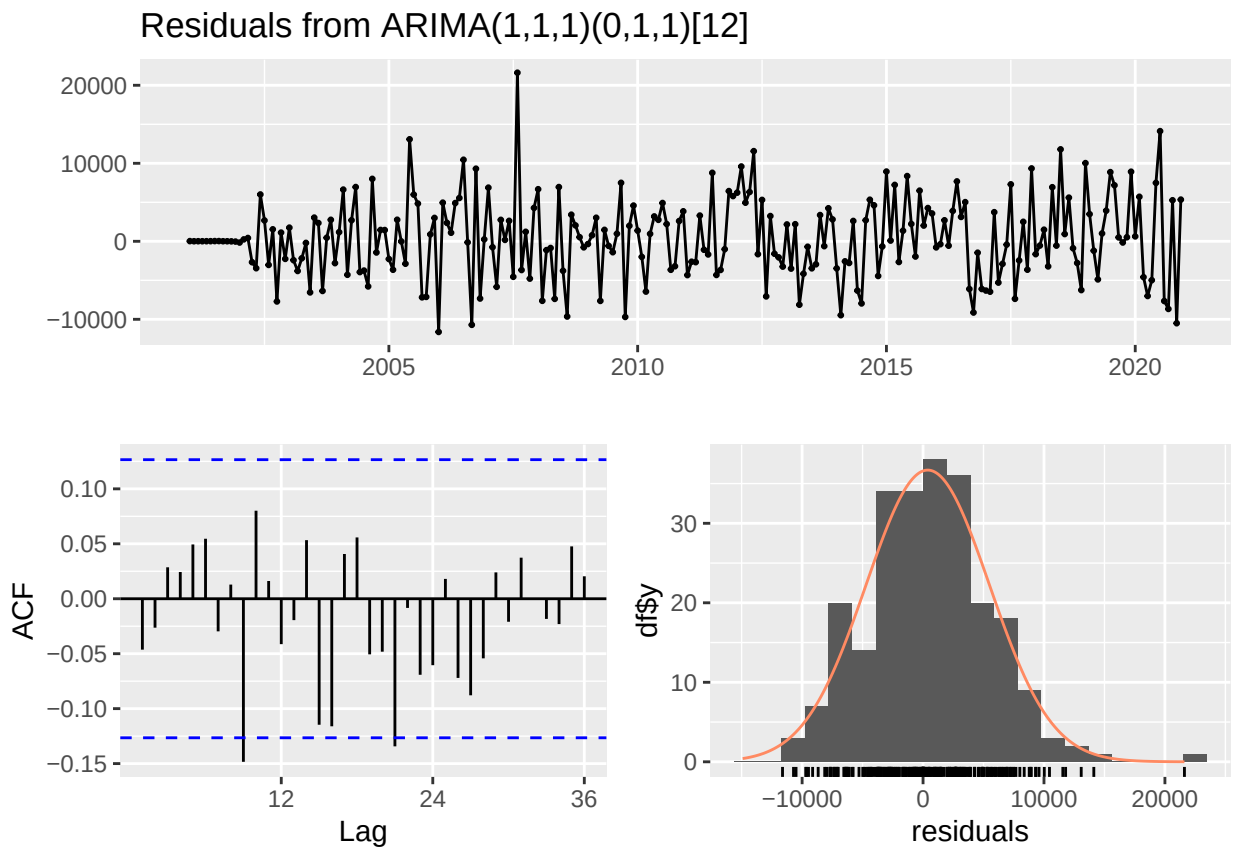
Compare the residual series for Q7 and Q6. Can you tell which ARIMA model is better representing the Natural Gas Series? Is that a fair comparison? Explain your response.

```
#Checking the residuals of SARIMA(1,1,1)(0,0,1)
checkresiduals(USElecGen_NG_SARIMA111001)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,1)(0,0,1)[12] with drift
## Q* = 250.75, df = 21, p-value < 2.2e-16
##
## Model df: 3.   Total lags used: 24
```

```
#Checking the residuals of SARIMA(1,1,1)(0,1,1)
checkresiduals(USElecGen_NG_SARIMA111011)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,1,1)(0,1,1)[12]
## Q* = 27.607, df = 21, p-value = 0.1516
##
## Model df: 3.   Total lags used: 24
```

Analysis: Model SARIMA(1,1,1)(0,0,1) has multiple significant ACF spikes in its residuals, and their plot has more deviations away from 0, when many points exceeding the +/- 1000 mark. On the other hand, SARIMA(1,1,1)(0,1,1) loses the signficant ACF spikes and the residuals are a lot more centred aroudn 0 with most points within the +/- 1000 threshold. Both these factors help us determine which model is doing a better job of fitting the data.

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?

```
print(auto.arima(USElecGen_NG_deseasoned))
```

```
## Series: USElecGen_NG_deseasoned
## 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
```

```
#This answer matches!
```

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?

```
print(auto.arima(USElecGen_NG_ts))
```

```
## Series: USElecGen_NG_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
```

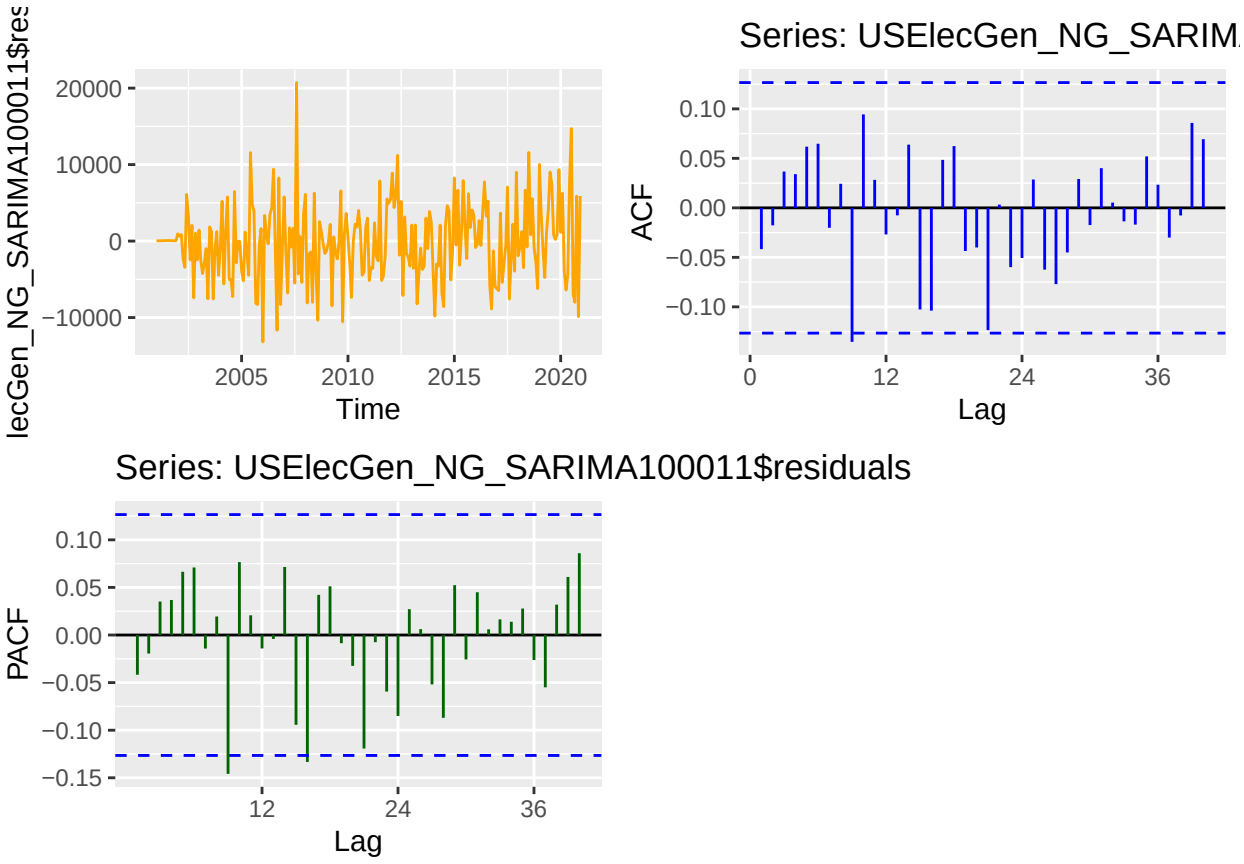
Analysis: the seasonal parameters match my answer P=0, D=1, Q=1 but the mistake I made was that I retained the order of p,d,q based on the answers of qn 4 - which in retrospect is wrong because those orders are based on deseseasoned data, which this time it is not. Running the correct orders below to see the correct result.

```
#Running SARIMA(1,0,0)(0,1,1)

USElecGen_NG_SARIMA100011 <- Arima(USElecGen_NG_ts, order=c(1,0,0), seasonal=c(0,1,1),
                                   include.drift=TRUE)
print(summary(USElecGen_NG_SARIMA100011))
```

```
## Series: USElecGen_NG_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
##
## Training set error measures:
##                      ME     RMSE      MAE        MPE     MAPE      MASE
## Training set -97.32578 5083.901 3950.295 -0.7114711 4.673706 0.4829553
##                      ACF1
## Training set -0.04171074
```

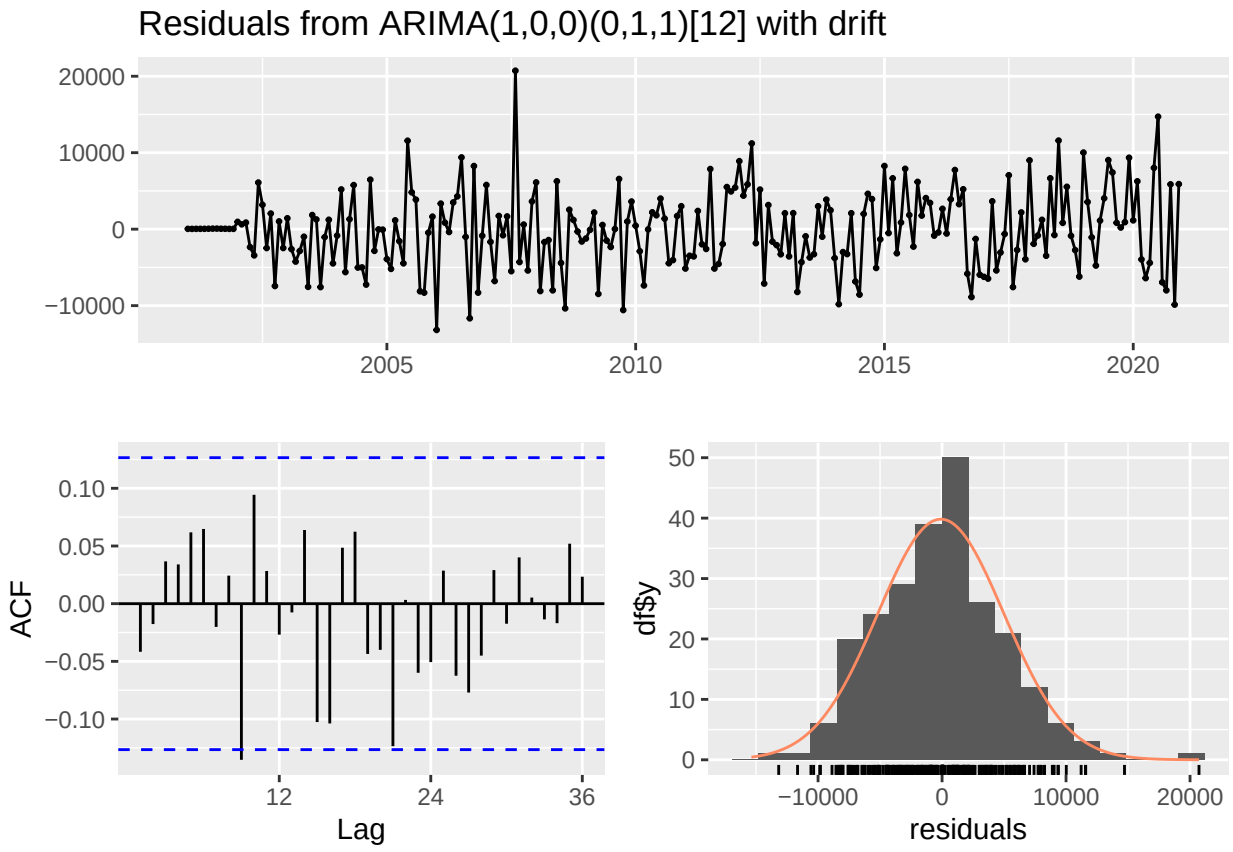
```
plot_grid(autoplot(USElecGen_NG_SARIMA100011$residuals, col="orange"),
          autoplot(Acf(USElecGen_NG_SARIMA100011$residuals, lag.max=40, plot=FALSE),col="blue"),
          autoplot(Pacf(USElecGen_NG_SARIMA100011$residuals, lag.max=40, plot=FALSE), col="darkgreen"))
```




```
print(USElecGen_NG_SARIMA100011$aic)
```

```
## [1] 4567.082
```

```
#interesting to see that the AIC of this model is higher than that of SARIMA(1,1,1)(0,1,1)
#which was 4552

checkresiduals(USElecGen_NG_SARIMA100011)
```




```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(1,0,0)(0,1,1)[12] with drift
## Q* = 25.414, df = 22, p-value = 0.2777
##
## Model df: 2.   Total lags used: 24
```