Energy Usage Time Series

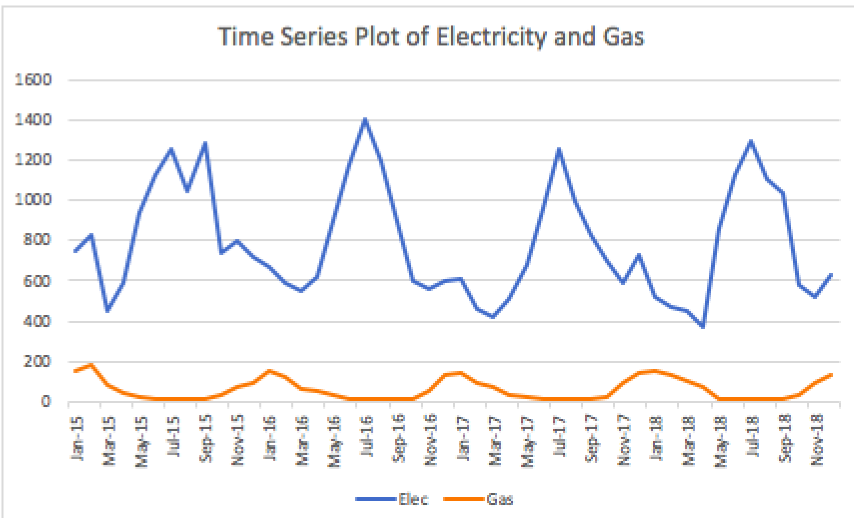
Project Summary

The purpose here is to detect any trends or other patterns in home energy consumption, and to enable me to identify baseline gas and electrical usage, as distinct from seasonally dependent usage. What usage statistics are predicted to occur in 2019?

Data Processing

#1

<u>Time series plot of the electric and gas series:</u>



For the electricity data, one pattern that stands out is that it is being used most during the months of July and least during the months of March. We can easily predict that because July is in the middle of summer, when everything is really hot, the home's electrical appliances such as the central air conditioning are being used most.

Whereas, for gas data, its peaks are always during January every year from 2015-2018. Those are the middle of winter's months and since it's really cold outside, the home might use most gas for heat and hot water.

Both series of data indicate significant seasonal patterns. Their trend, even though not significant, can exist as electricity usage has a slight drop and gas usage increases just a little over time.

#2

*Refer to the Appendix by the end of this project write-up

According to appendix, the naive forecasting method was most successful according to MAE criterion. Of which value is 175.8, smaller than that of overall forecasts at 257.5, and that of moving average order 3 forecasts at 243.2.

#3

Since the time series contains a strong seasonality component, we should use decomposition.

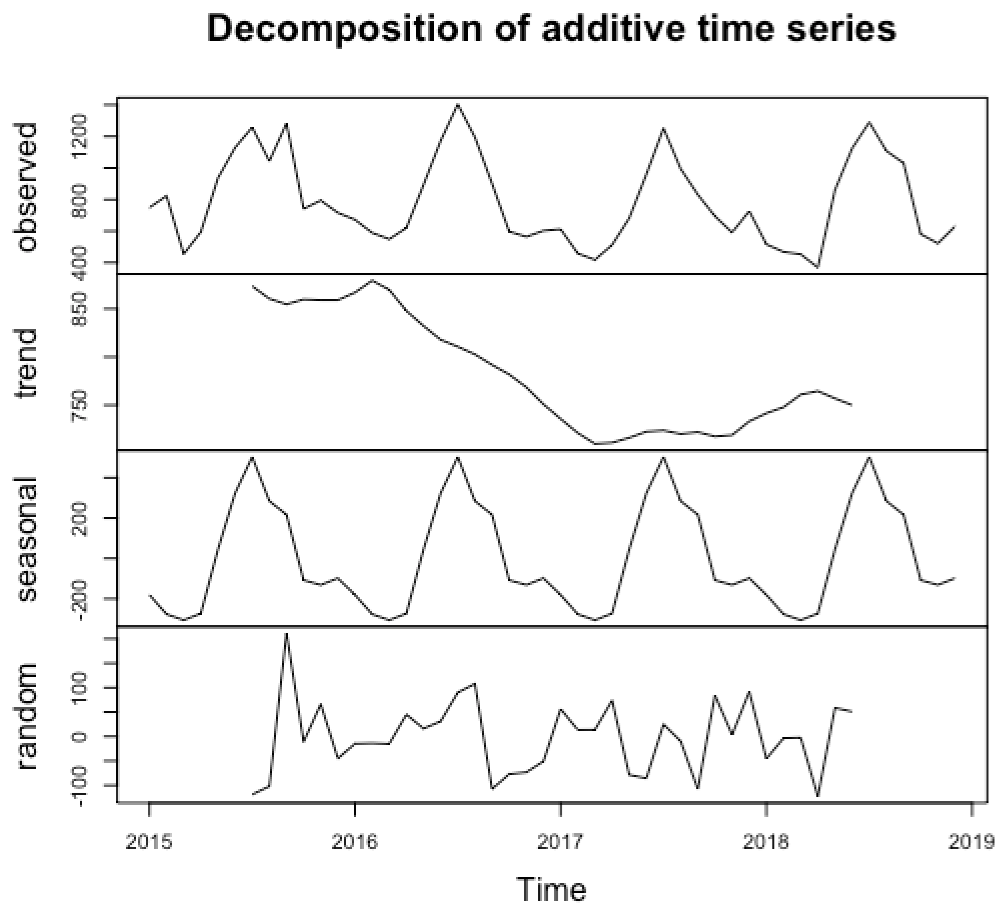
```
> pro4<- read.csv("Desktop/energy.csv")
```

Time series objects for electricity and gas:

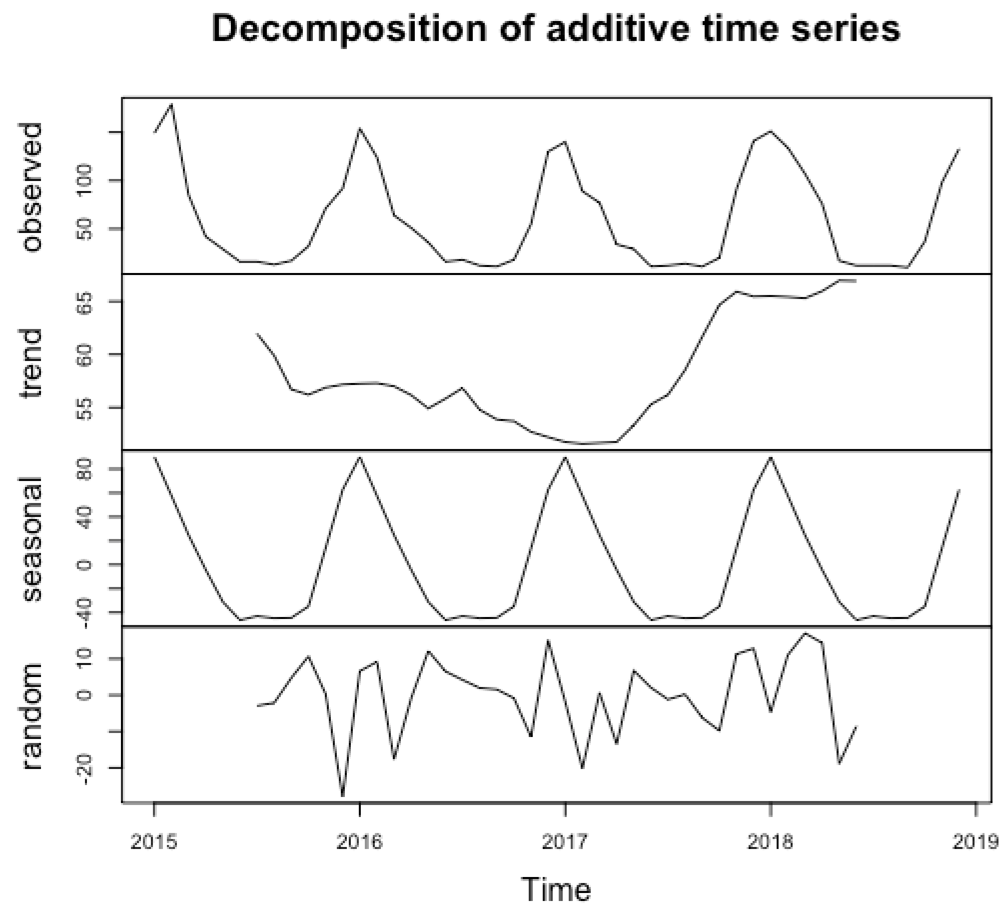
```
> elec <- ts(pro4$Elec, frequency=12, start=c(2015,1))
> gas <- ts(pro4$Gas, frequency = 12, start=c(2015,1))
```

Decomposition of each series and their graphs:

```
> install.packages("TTR")
> library("TTR")
> decompelec <- decompose(elec)
> plot(decompelec)
```




```
> decompgas <- decompose(gas)
> plot(decompgas)
```



<u>Regression of trend datas on time:</u>

```
> time <- c(1:48)
> elecmodel <- lm(decompelec$trend~time)
> summary(elecmodel)
```

```
Call:
lm(formula = decompelec$trend ~ time)

Residuals:
   Min      1Q  Median      3Q     Max
-60.63 -22.27    1.60   20.02   55.24

Coefficients:
            Estimate  Std. Error  t value   Pr(>|t|)
(Intercept) 897.4979   14.1449    63.450   < 2e-16 ***
time         -4.7048    0.5315    -8.851   2.41e-10 ***
---
```

```
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 33.13 on 34 degrees of freedom
  (12 observations deleted due to missingness)
Multiple R-squared:  0.6974,  Adjusted R-squared:  0.6885
F-statistic: 78.35 on 1 and 34 DF,  p-value: 2.409e-10

> gasmodel <- lm(decompgas$trend~time)
> summary(gasmodel)

Call:
lm(formula = decompgas$trend ~ time)

Residuals:
   Min     1Q  Median     3Q    Max
-7.355 -3.715  1.261  3.737  8.274

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 51.87695    1.86630  27.797  <2e-16 ***
time         0.25814    0.07013   3.681   8e-04 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 4.371 on 34 degrees of freedom
  (12 observations deleted due to missingness)
Multiple R-squared:  0.2849,  Adjusted R-squared:  0.2639
F-statistic: 13.55 on 1 and 34 DF,  p-value: 0.0007999
```

Looking at the above models, we can say that, without taking the seasonal aspect into account, there exists a trend on energy usage. Where for electricity, the trend is decreasing at 4.7 units over time and for gas, the trend is increasing at 0.3 units over time. Both trends are significant at the 0.05 level.

<u>Prediction on electrical and gas usage for the first three months of 2019:</u>
From our trend observation and patterns of seasonal component discussed in question 1, we are going to include both seasonal and trend aspects into our linear regression model on the original data:

```
> season <- factor(rep(c(1,2,3,4,5,6,7,8,9,10,11,12),4))
> season
 [1] 1  2  3  4  5  6  7  8  9  10 11 12 1  2  3  4  5  6  7  8  9  10 11 12 1  2  3  4  5  6  7  8  9  10 11
12 1
```

```
[38] 2  3  4  5  6  7  8  9  10 11 12
Levels: 1 2 3 4 5 6 7 8 9 10 11 12
> elecmodel <- lm(elec~time+season)
> summary(elecmodel)

Call:
lm(formula = elec ~ time + season)

Residuals:
    Min      1Q  Median      3Q     Max 
-156.613  -69.181   -0.975   61.656  199.838 

Coefficients:
            Estimate Std. Error t value  Pr(>|t|)    
(Intercept)  712.894     51.604  13.815  9.79e-16 ***
time          -3.981      1.028  -3.874  0.000449 ***
season2      -48.769     67.560  -0.722  0.475178    
season3     -160.787     67.584  -2.379  0.022941 *  
season4     -101.556     67.623  -1.502  0.142116    
season5      221.425     67.677   3.272  0.002407 ** 
season6      478.906     67.748   7.069  3.11e-08 ***
season7      687.638     67.833  10.137  5.93e-12 ***
season8      475.869     67.934   7.005  3.77e-08 ***
season9      406.100     68.051   5.968  8.52e-07 ***
season10      51.831     68.183   0.760  0.452237    
season11      20.313     68.330   0.297  0.768016    
season12      75.044     68.492   1.096  0.280710    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 95.53 on 35 degrees of freedom
Multiple R-squared:  0.9123,  Adjusted R-squared:  0.8823 
F-statistic: 30.35 on 12 and 35 DF,  p-value: 6.108e-15

> gasmodel <- lm(gas ~ time + season)
> summary(gasmodel)

Call:
lm(formula = gas ~ time + season)

Residuals:
    Min      1Q  Median      3Q     Max 
-43.142  -6.075   0.592   6.267  49.425 
```

Coefficients:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	146.4681	8.7597	16.721	< 2e-16	***
time	0.1069	0.1745	0.613	0.5438	
season2	-17.1069	11.4682	-1.492	0.1447	
season3	-65.4639	11.4721	-5.706	1.88e-06	***
season4	-98.0708	11.4788	-8.544	4.40e-10	***
season5	-121.1778	11.4881	-10.548	2.06e-12	***
season6	-135.2847	11.5000	-11.764	1.03e-13	***
season7	-134.6417	11.5145	-11.693	1.22e-13	***
season8	-136.4986	11.5317	-11.837	8.62e-14	***
season9	-137.1056	11.5515	-11.869	7.98e-14	***
season10	-122.7125	11.5738	-10.603	1.79e-12	***
season11	-70.8194	11.5988	-6.106	5.60e-07	***
season12	-25.6764	11.6263	-2.208	0.0339	*

Signif. codes: 0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 16.22 on 35 degrees of freedom
Multiple R-squared: 0.9263, Adjusted R-squared: 0.901
F-statistic: 36.64 on 12 and 35 DF, p-value: 3.17e-16

The p-values are significantly small and the R^2 values are huge, so we appear to have a good model.

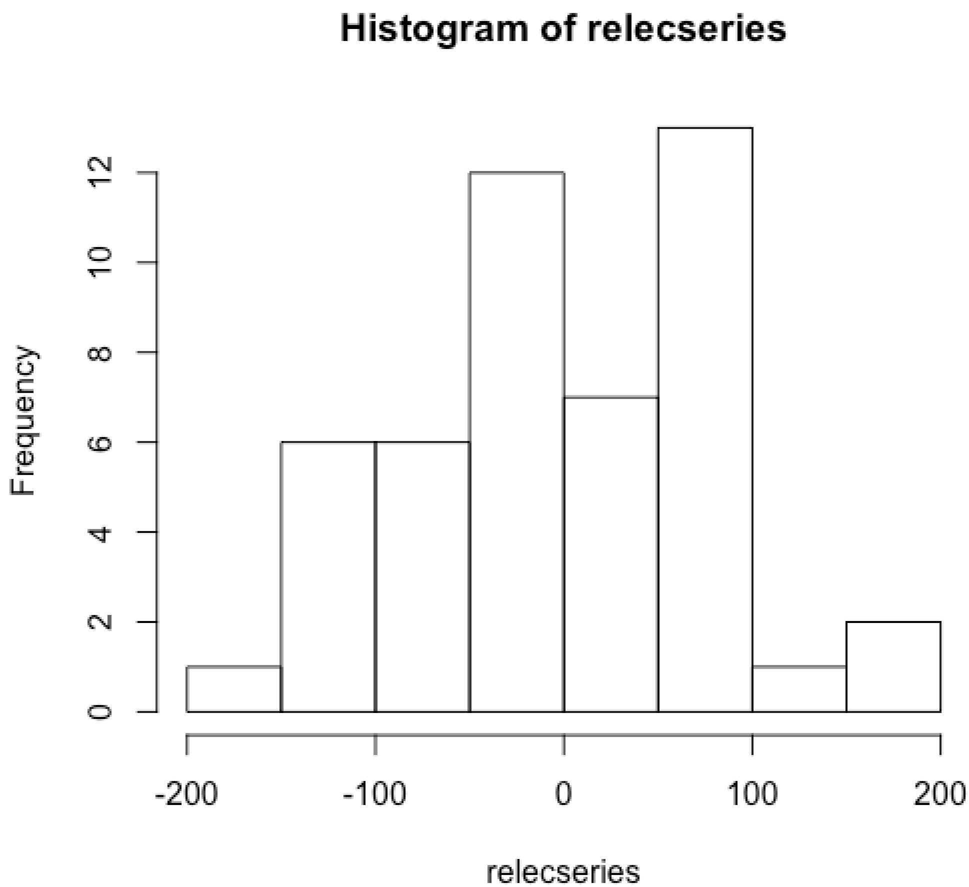
The first, second, and third month of 2019’s time would be 49, 50, 51; and their season variable would be season1 = 1, season2 = 1, and season3 = 1 accordingly while all the other season*i* variables are turned to zero.

- Predicted Electricity usage:
  - January: y = -3.981*49 + 712.894 = 517.8 (kwh)
  - February: y = -3.981*50 - 48.769*1 + 712.894 = 465.1 (kwh)
  - March: y = -3.981*51 - 160.787*1 + 712.894 = 349.1 (kwh)
- Predicted Gas usage:
  - January: y = 0.1069*49 + 146.4681 = 151.7 (therms)
  - February: y = 0.1069*50 - 17.1069*1 +146.4681 = 134.7 (therms)
  - March: y = 0.1069*51 - 65.4639*1 + 146.4681 = 86.5 (therms)

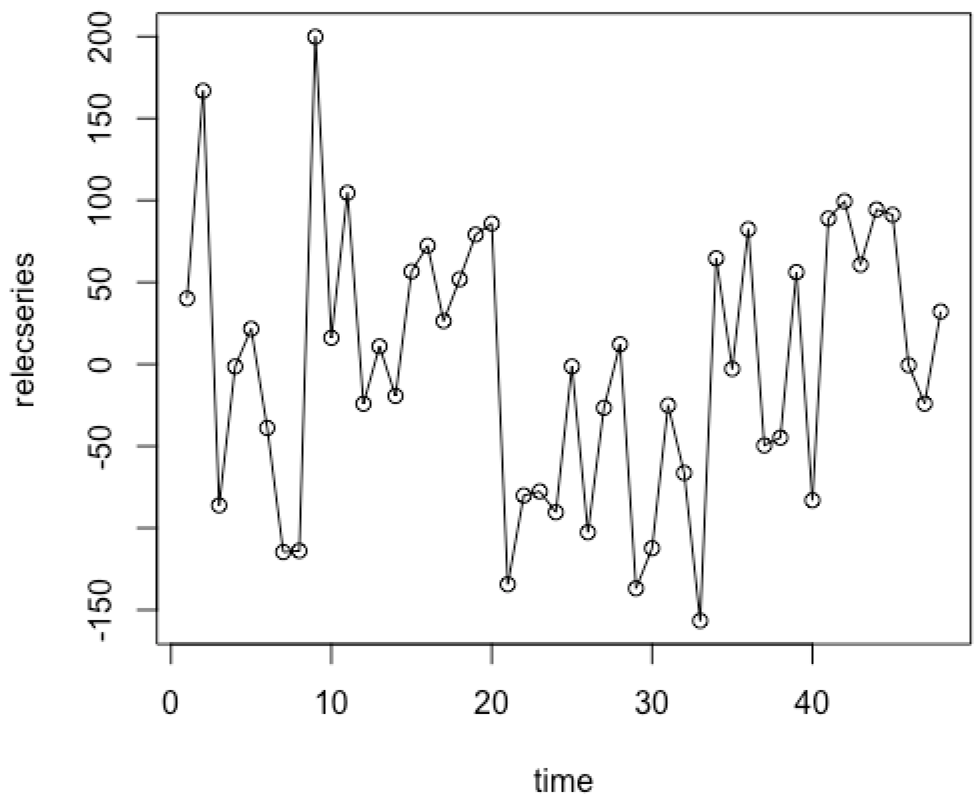
#4

<u>Random component of electricity series:</u>

```
> relecseries <- elecmodel$residuals
> hist(relecseries)
```

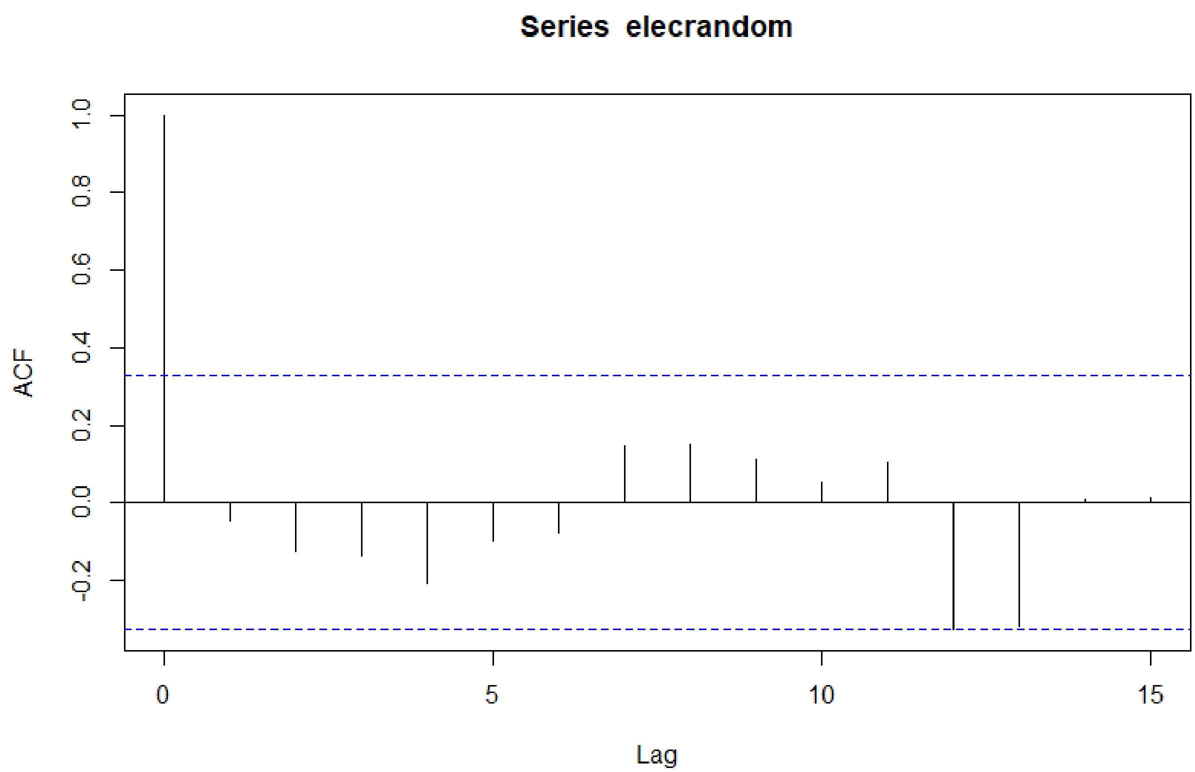



```
> plot(time,relecseries,type="n")
> lines(time,relecseries,type="o")
```

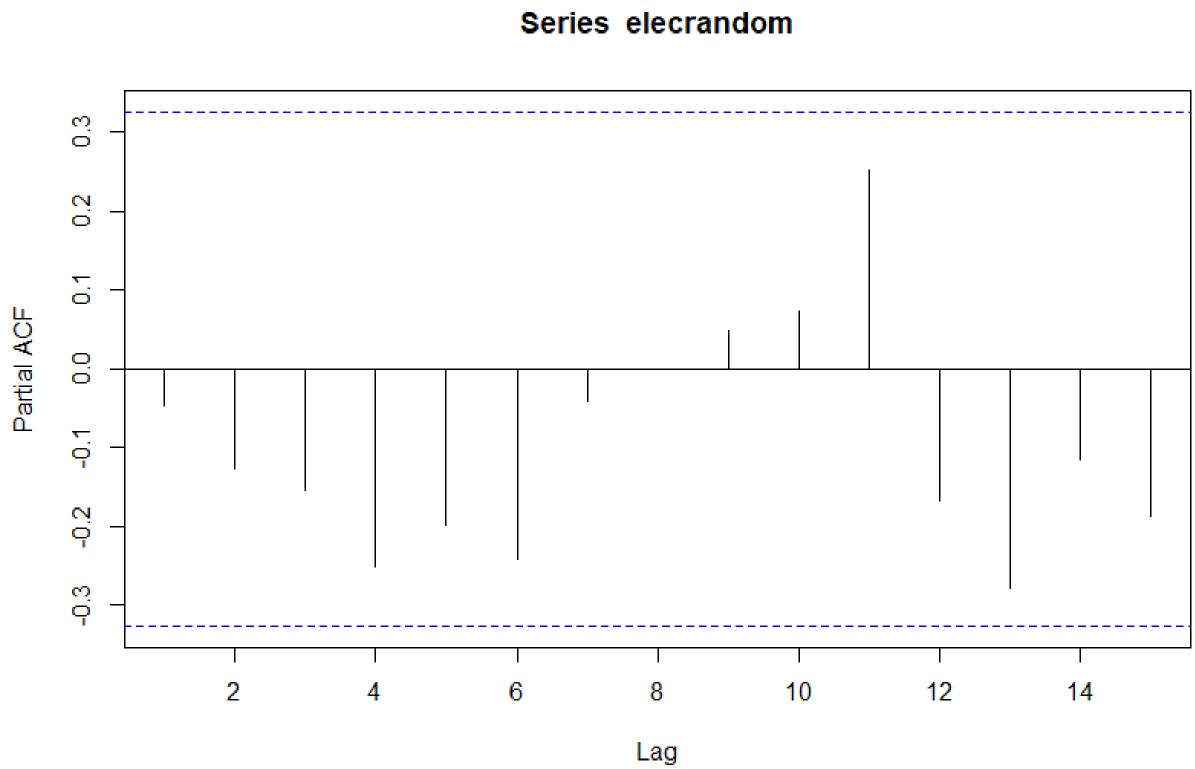



```
> elec1<-elecmodel$residuals[1:47]
> elec2<-elecmodel$residuals[2:48]

> cor.test(elec1, elec2)

        Pearson's product-moment correlation

data:  elec1 and elec2
t = 1.2119, df = 45, p-value = 0.2319
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.1152674  0.4423666
sample estimates:
      cor
0.1777842

> install.packages("tseries")
> library(tseries)
```

```
> elecrandom <- decompelec$random[7:42]
> acf(elecrandom)
```



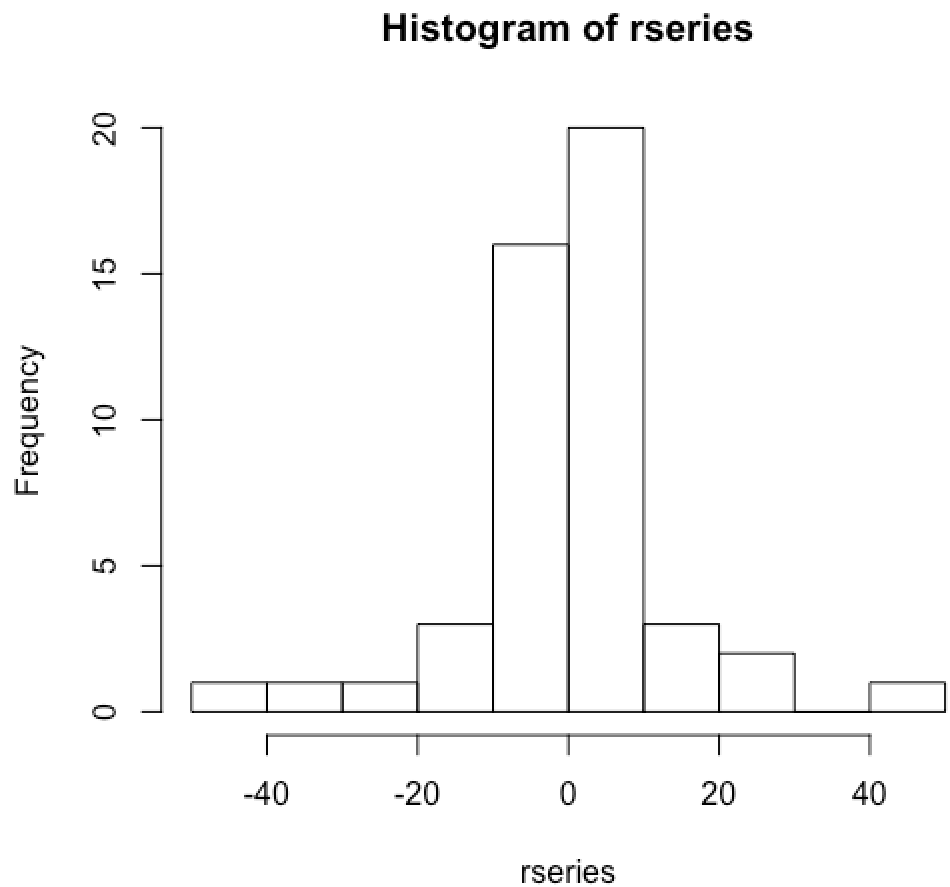

```
> pacf(elecrandom)
```



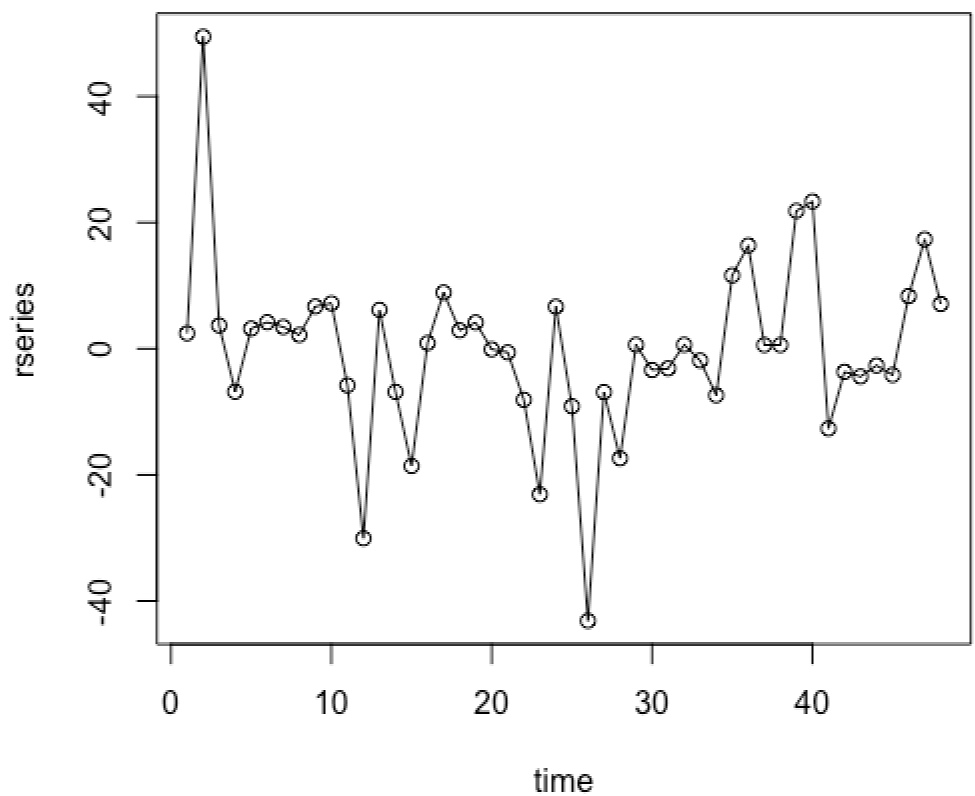
As can be seen, the residuals of the electricity series are not very normally distributed. However, it looks significantly random in time as shown in the lines graph. The big p-value of 0.2319 from the correlation test also makes it impossible to reject the hypothesis that the Xt and Xt+1 are correlated. This is illustrated in the acf test: at a lag of 0, there is perfect correlation, but there is not much correlation between the other lags. The function shows a confidence band at approximately 0.35, so we can see that there is not much significant lag in the previous steps. The pacf gives the same result with insignificance correlations at every value of lags (0.35 confidence band). Therefore, the random component of the electricity series is just white noise.

<u>Random component of gas series:</u>

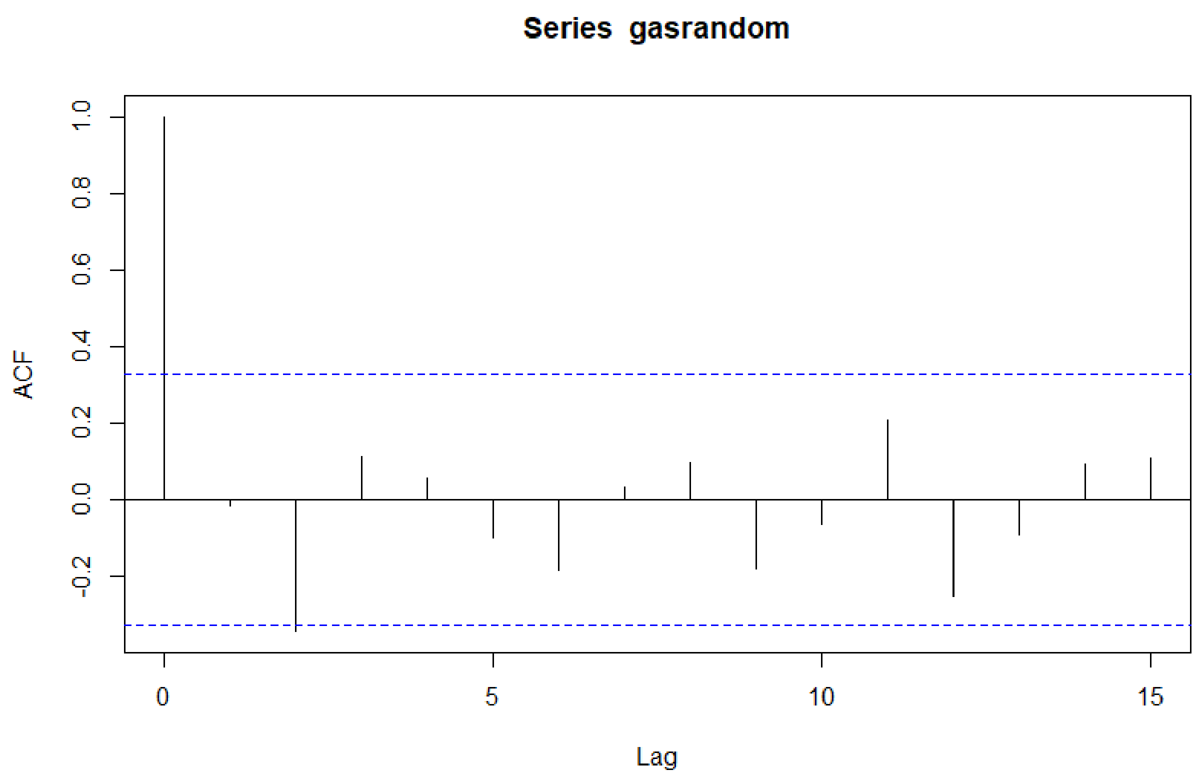
```
> rgasseries <- gasmodel$residuals
> hist(rgasseries)
```



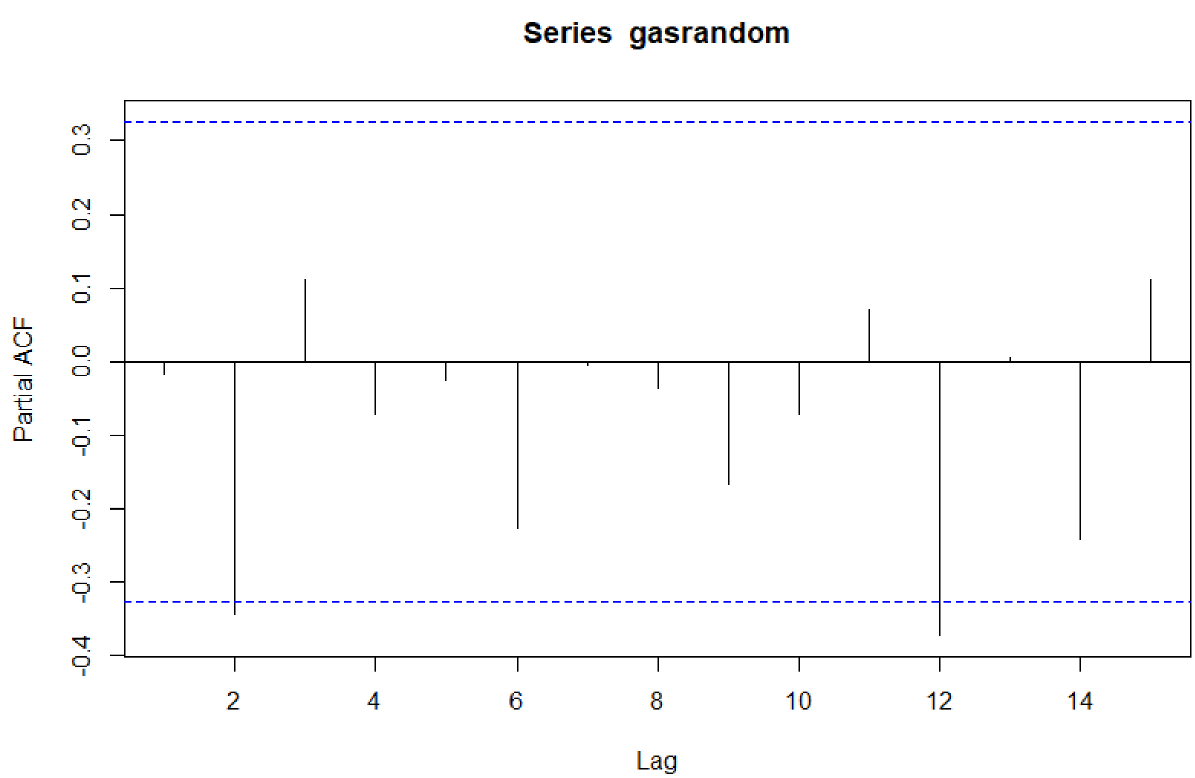

```
> plot(time,rgasseries,type="n")
> lines(time,rgasseries,type="o")
```




```
> gas1<-gasmodel$residuals[1:47]
> gas2<-gasmodel$residuals[2:48]

> cor.test(gas1,gas2)

        Pearson's product-moment correlation

data:  gas1 and gas2
t = 1.3972, df = 45, p-value = 0.1692
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.08844449  0.46390459
sample estimates:
      cor
0.2039011

> acf(gasrandom)
```



> pacf(gasrandom)



Similarly, the residuals of the gas series are highly normally distributed. As shown in the lines graph, it looks significantly random in time. The big p-value of 0.1692 from the correlation test also makes it impossible to reject the hypothesis that Xt and Xt+1 are correlated. Such is also illustrated in the acf test: at a lag of 0, there is perfect correlation, but there is not much correlation between the other lags. At a confidence band of approximately 0.35, we can see that there is not much significant lag in the previous steps. Therefore, the random component of the gas series is also just white noise.

Appendix

Elec	Naïve	Overall	MA(1)	Naïve Err	Overall Err	MA(1) Err
749						
823	749	749		74	74	
454	823	786		369	332	
594	454	675.33333	675.3333	140	81.33333	81.33333
936	594	655	623.6667	342	281	312.3333
1129	936	711.2	661.3333	193	417.8	467.6667
1258	1129	780.83333	886.3333	129	477.1667	371.6667
1043	1258	849	1107.667	215	194	64.66667
1283	1043	873.25	1143.333	240	409.75	139.6667
741	1283	918.77778	1194.667	542	177.7778	453.6667
794	741	901	1022.333	53	107	228.3333
716	794	891.27273	939.3333	78	175.2727	223.3333
672	716	876.66667	750.3333	44	204.6667	78.33333
589	672	860.92308	727.3333	83	271.9231	138.3333
549	589	841.5	659	40	292.5	110
620	549	822	603.3333	71	202	16.66667
893	620	809.375	586	273	83.625	307

1172	893	814.29412	687.3333	279	357.7059	484.6667
1404	1172	834.16667	895	232	569.8333	509
1195	1404	864.15789	1156.333	209	330.8421	38.66667
901	1195	880.7	1257	294	20.3	356
597	901	881.66667	1166.667	304	284.6667	569.6667
564	597	868.72727	897.6667	33	304.7273	333.6667
602	564	855.47826	687.3333	38	253.4783	85.33333
612	602	844.91667	587.6667	10	232.9167	24.33333
458	612	835.6	592.6667	154	377.6	134.6667
418	458	821.07692	557.3333	40	403.0769	139.3333
512	418	806.14815	496	94	294.1481	16
682	512	795.64286	462.6667	170	113.6429	219.3333
960	682	791.72414	537.3333	278	168.2759	422.6667
1252	960	797.33333	718	292	454.6667	534
995	1252	812	964.6667	257	183	30.33333
831	995	817.71875	1069	164	13.28125	238
694	831	818.12121	1026	137	124.1212	332
591	694	814.47059	840	103	223.4706	249
727	591	808.08571	705.3333	136	81.08571	21.66667
516	727	805.83333	670.6667	211	289.8333	154.6667
468	516	798	611.3333	48	330	143.3333
453	468	789.31579	570.3333	15	336.3158	117.3333
369	453	780.69231	479	84	411.6923	110
860	369	770.4	430	491	89.6	430
1124	860	772.58537	560.6667	264	351.4146	563.3333

1290	1124	780.95238	784.3333	166	509.0476	505.6667
1108	1290	792.7907	1091.333	182	315.2093	16.66667
1031	1108	799.95455	1174	77	231.0455	143
581	1031	805.08889	1143	450	224.0889	562
522	581	800.21739	906.6667	59	278.2174	384.6667
629	522	794.29787	711.3333	107	165.2979	82.33333
MAE				175.8298	257.5408	243.2074