

Time Series Analysis

Homework3 201811526 이은주

Data

```
data = read.csv('loadregr.csv') #find a proper directory for your data.
(data_ts = ts(data$MKw, start=c(1970, 1), end=c(1979, 12), frequency=12 ))
```

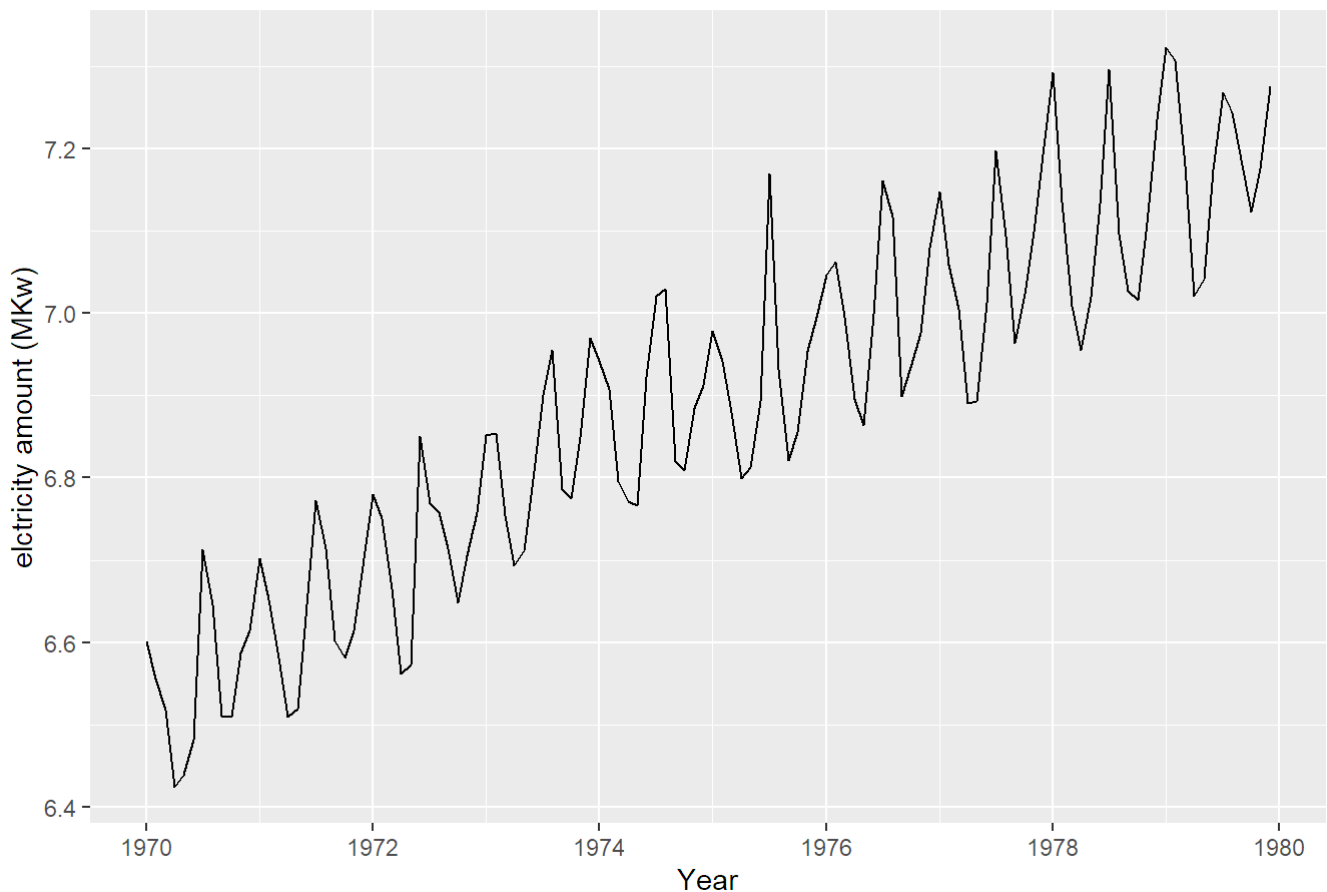
##	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep
## 1970	6.60110	6.55576	6.51810	6.42498	6.43868	6.48173	6.71284	6.64639	6.51100
## 1971	6.70224	6.65245	6.58104	6.50932	6.51981	6.64591	6.77194	6.71382	6.60164
## 1972	6.77956	6.75018	6.66494	6.56197	6.57278	6.85069	6.76831	6.75746	6.71462
## 1973	6.85132	6.85432	6.75464	6.69287	6.71209	6.79801	6.90086	6.95501	6.78619
## 1974	6.94209	6.90815	6.79581	6.77106	6.76630	6.92218	7.02025	7.02929	6.81991
## 1975	6.97866	6.94202	6.87909	6.79965	6.81259	6.89272	7.16972	6.93213	6.82103
## 1976	7.04580	7.06250	6.99577	6.89578	6.86417	6.99927	7.16027	7.11666	6.89878
## 1977	7.14766	7.05902	7.00532	6.89115	6.89297	7.01257	7.19739	7.09346	6.96310
## 1978	7.29256	7.13518	7.00859	6.95518	7.01947	7.13228	7.29528	7.09632	7.02693
## 1979	7.32306	7.30641	7.17916	7.02096	7.04144	7.17437	7.26813	7.24400	7.18675
##	Oct	Nov	Dec						
## 1970	6.51021	6.58769	6.61386						
## 1971	6.58158	6.61455	6.70228						
## 1972	6.64869	6.70601	6.75843						
## 1973	6.77511	6.84878	6.96915						
## 1974	6.80915	6.88408	6.91290						
## 1975	6.85578	6.95368	6.99520						
## 1976	6.93408	6.97616	7.07897						
## 1977	7.02267	7.09898	7.19206						
## 1978	7.01620	7.11002	7.23462						
## 1979	7.12274	7.17493	7.27526						

1970년 1월부터 1979년 12월까지의 Iowa state의 electricity data입니다.

1. Draw time series plot

```
data_ts %>%  
  autoplot() +  
  ggtitle("Time Series Plot of Electricity data") +  
  xlab("Year") + ylab("electricity amount (MKw)")
```

Time Series Plot of Electricity data



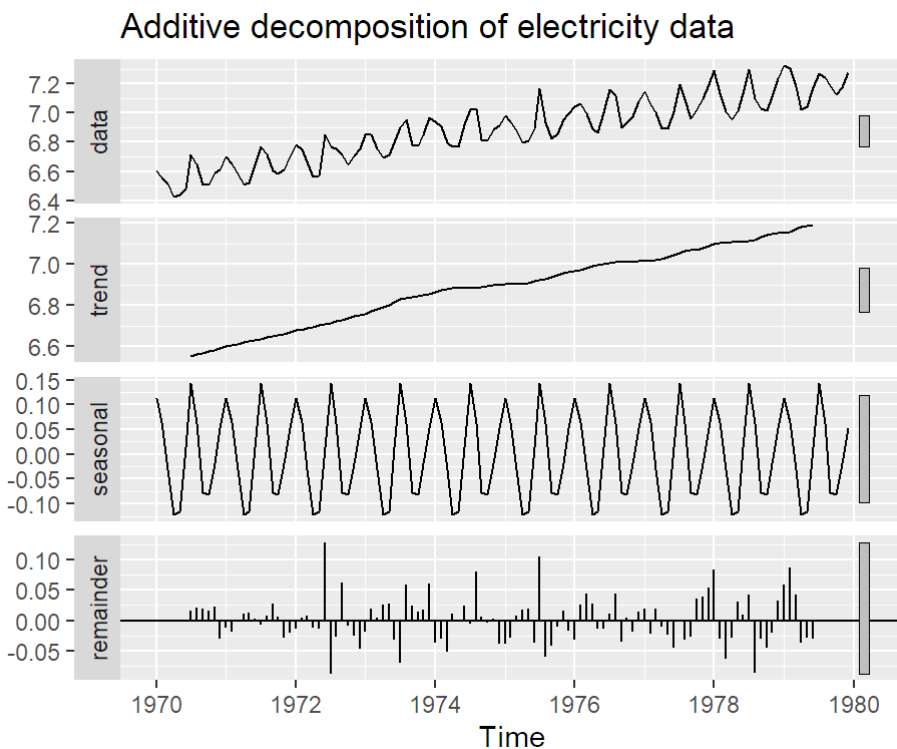
- 위의 time series plot을 보면 시간이 흐를수록 증가하는 추세를 볼 수 있고 일정 기간마다 상승 및 하강을 이루는 계절성을 보임을 알 수 있습니다. 이에 따라 추세성, 계절성, 잔차에 대해 decomposition해 볼 필요가 있음을 알 수 있습니다.

2. Fit decomposition model and draw a plot with the original data, seasonally adjusted, trend-cycle component all together.

- Fit decomposition model => 총 5가지 방법을 이용하여 decomposition을 한 후,
 - kpss.test를 통해 H_0 :residual is independent 를 검정한 결과
 - ACF, PACF가 유의수준 내에 존재하는 가
 두가지를 기준으로 고려하여 가장 정상성을 만족하는 모델을 최종모델로 선택하여 plot을 그렸습니다.

1. classical additive decomposition

```
dec_add = decompose(data_ts, type="additive")
dec_add %>%
  autoplot() +
  ggtitle("Additive decomposition of electricity data")
```

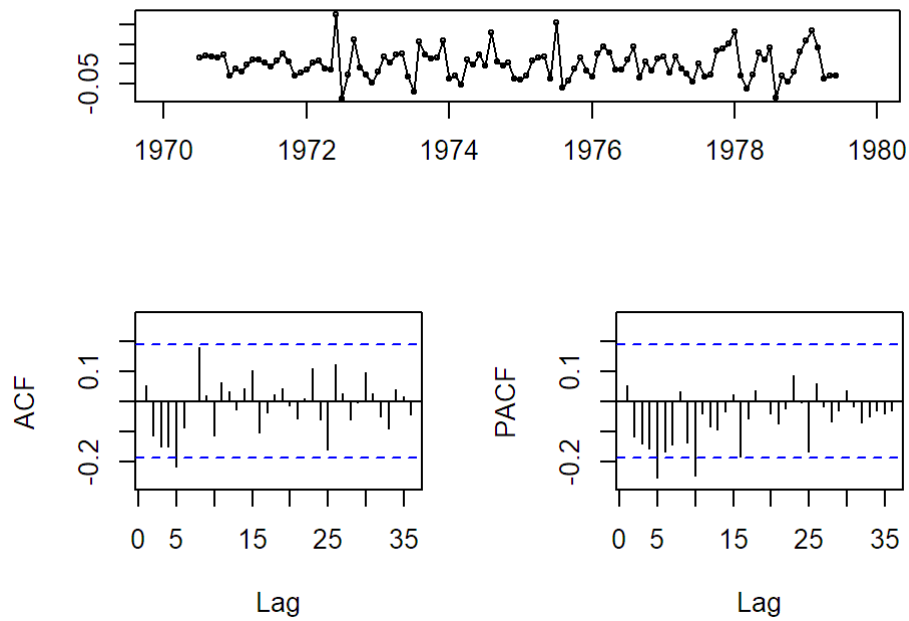


```
#diagnosis
tseries::kpss.test(dec_add$random, null="Level")
```

```
##
## KPSS Test for Level Stationarity
##
## data: dec_add$random
## KPSS Level = 0.00040235, Truncation lag parameter = 4, p-value = 0.1
```

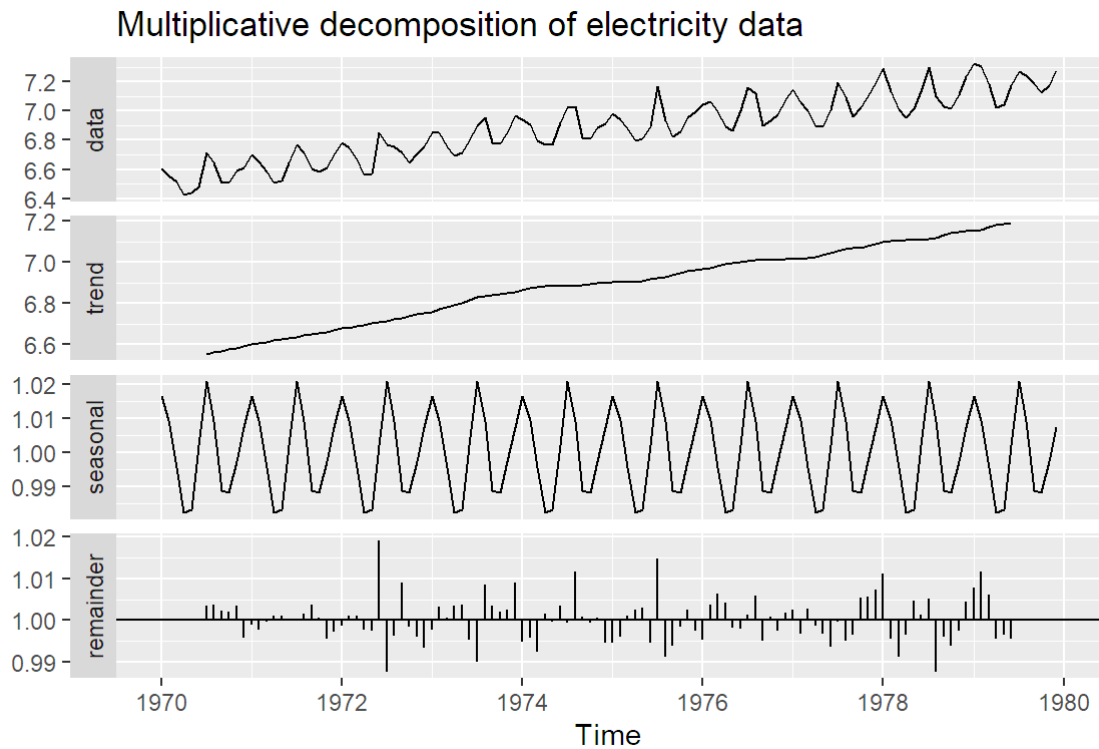
```
tsdisplay(dec_add$random, main="Random_Additive model")
```

Random_Additive model



2. classical multiplicative decomposition

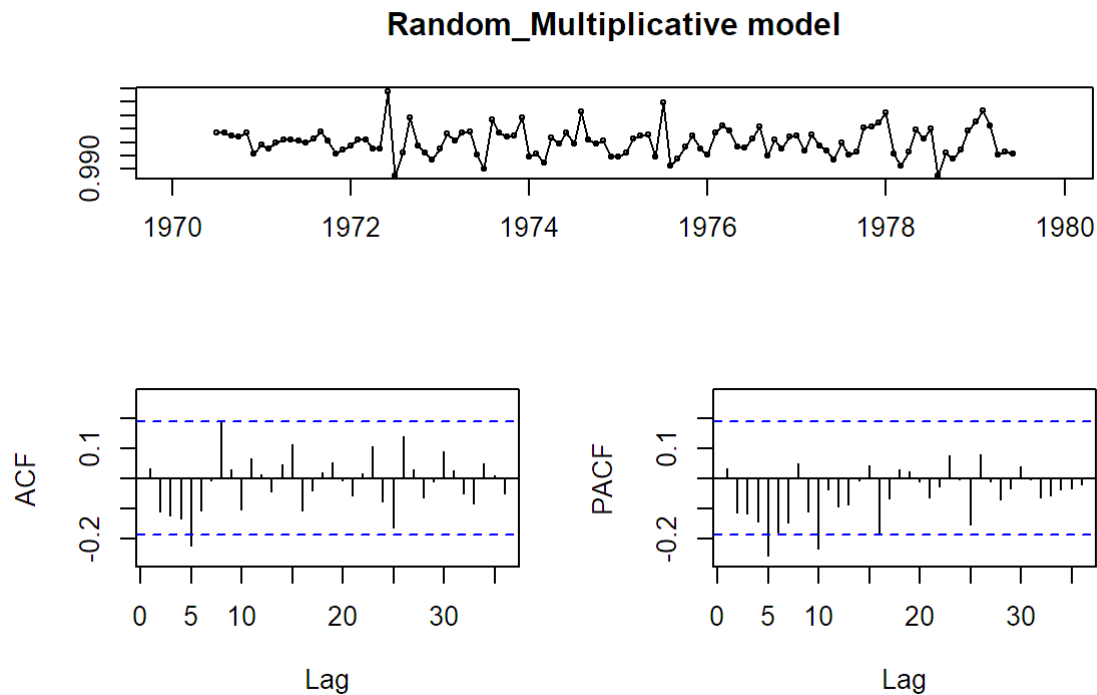
```
dec_mul = decompose(data_ts, type="multiplicative")
dec_mul %>%
  autoplot() +
  ggtitle("Multiplicative decomposition of electricity data")
```



```
#diagnosis
tseries::kpss.test(dec_mul$random, null="Level")
```

```
##
## KPSS Test for Level Stationarity
##
## data: dec_mul$random
## KPSS Level = 0.024423, Truncation lag parameter = 4, p-value = 0.1
```

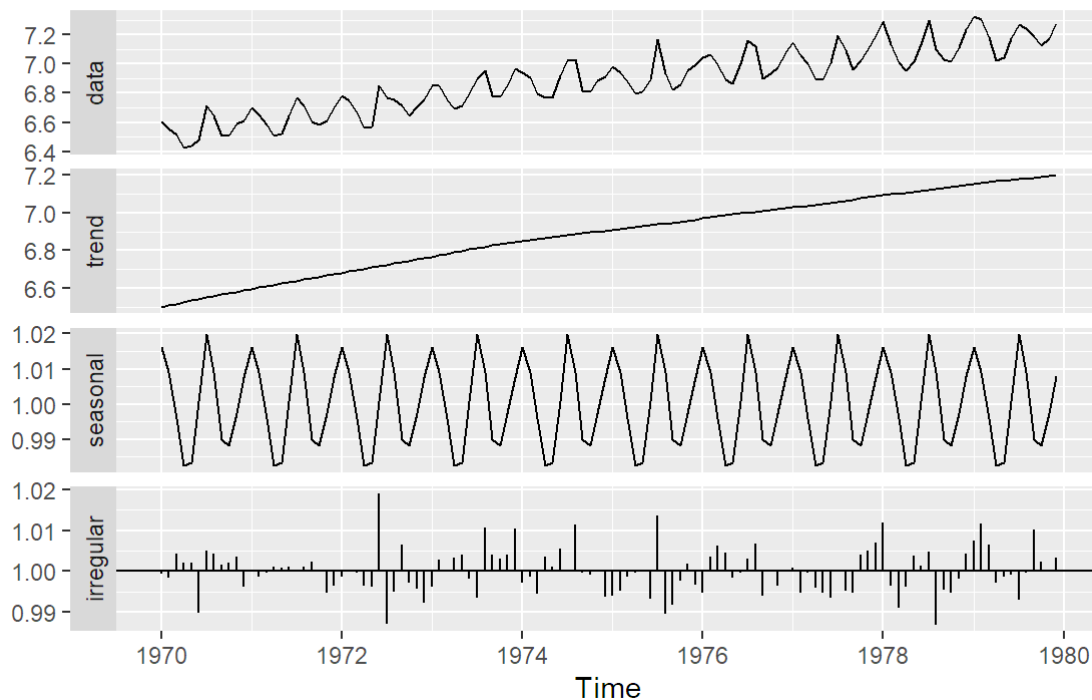
```
tsdisplay(dec_mul$random, main="Random_Multiplicative model")
```



3. SEATS decomposition

```
dec_seas = seas(data_ts)
dec_seas %>%
  autoplot() +
  ggtitle("SEATS decomposition of electiricity data")
```

SEATS decomposition of electricity data

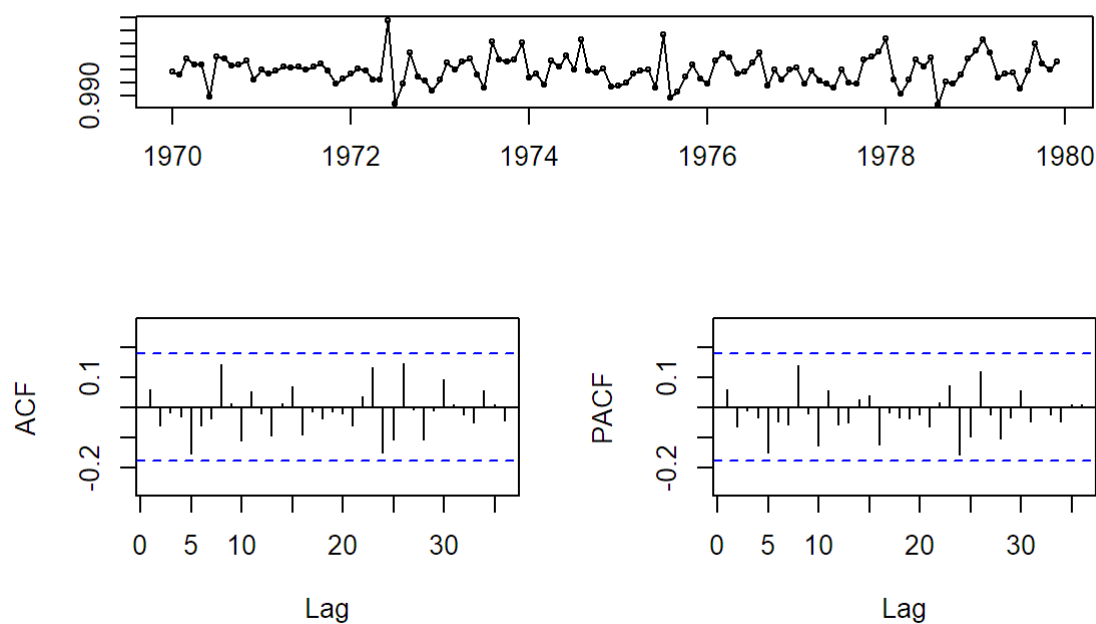


```
#diagnosis
tseries::kpss.test(dec_seas$data[, "irregular"], null="Level")
```

```
##
## KPSS Test for Level Stationarity
##
## data: dec_seas$data[, "irregular"]
## KPSS Level = 0.049134, Truncation lag parameter = 4, p-value = 0.1
```

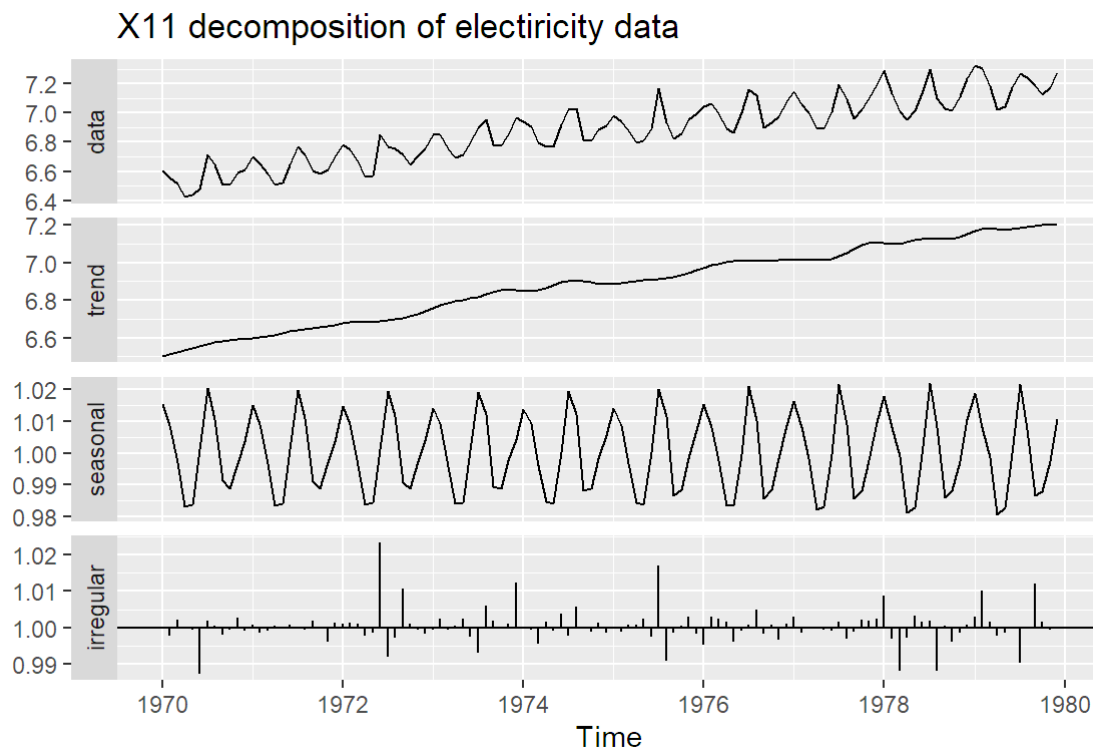
```
tsdisplay(dec_seas$data[, "irregular"], null="Level", main="Random_SEATS model")
```

Random_SEATS model



4. X11 decomposition

```
dec_x11 = seas(data_ts, x11="")
dec_x11 %>%
  autoplot() +
  ggtitle("X11 decomposition of electiricity data")
```

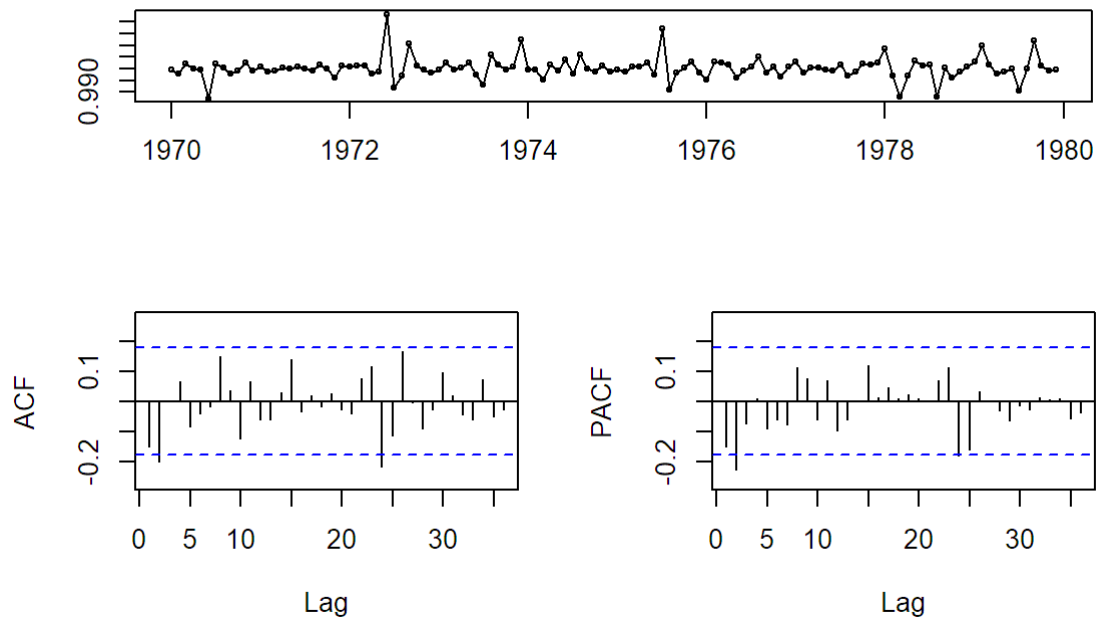


```
#diagonosis
tseries::kpss.test(dec_x11$data[, "irregular"], null="Level")
```

```
##
## KPSS Test for Level Stationarity
##
## data: dec_x11$data[, "irregular"]
## KPSS Level = 0.085333, Truncation lag parameter = 4, p-value = 0.1
```

```
tsdisplay(dec_x11$data[, "irregular"], null="Level", main="Random_X11 model")
```

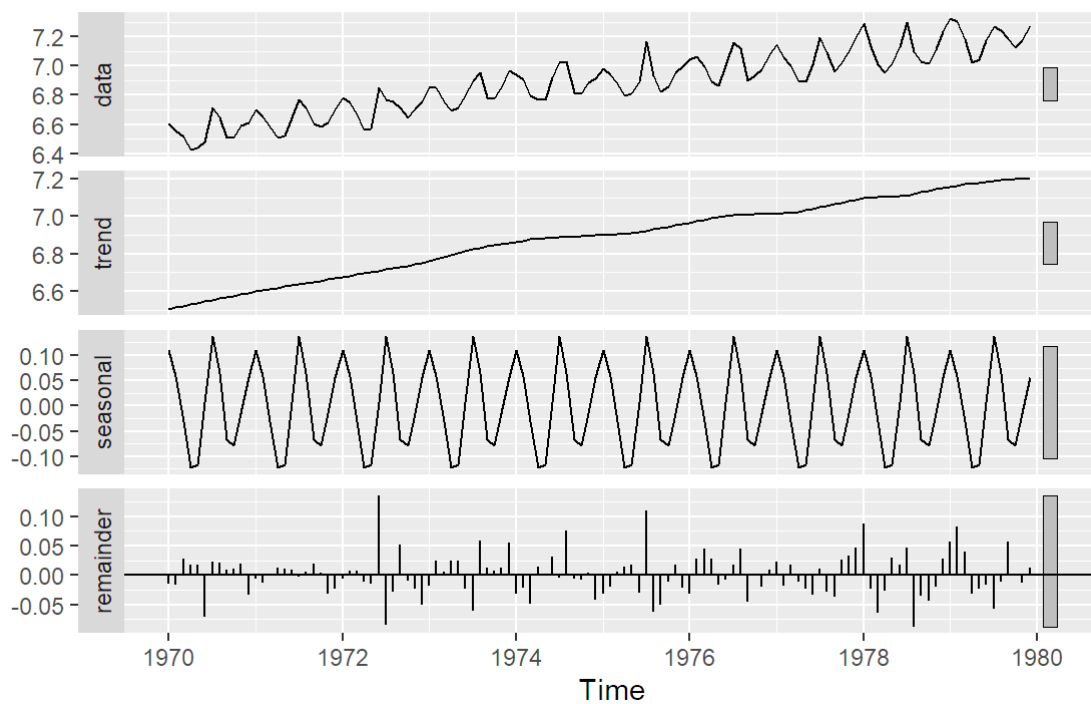
Random_X11 model



5. STL decomposition

```
dec_stl = stl(data_ts, "periodic")
dec_stl %>%
  autoplot() +
  ggtitle("STL decomposition of electricity data")
```

STL decomposition of electricity data



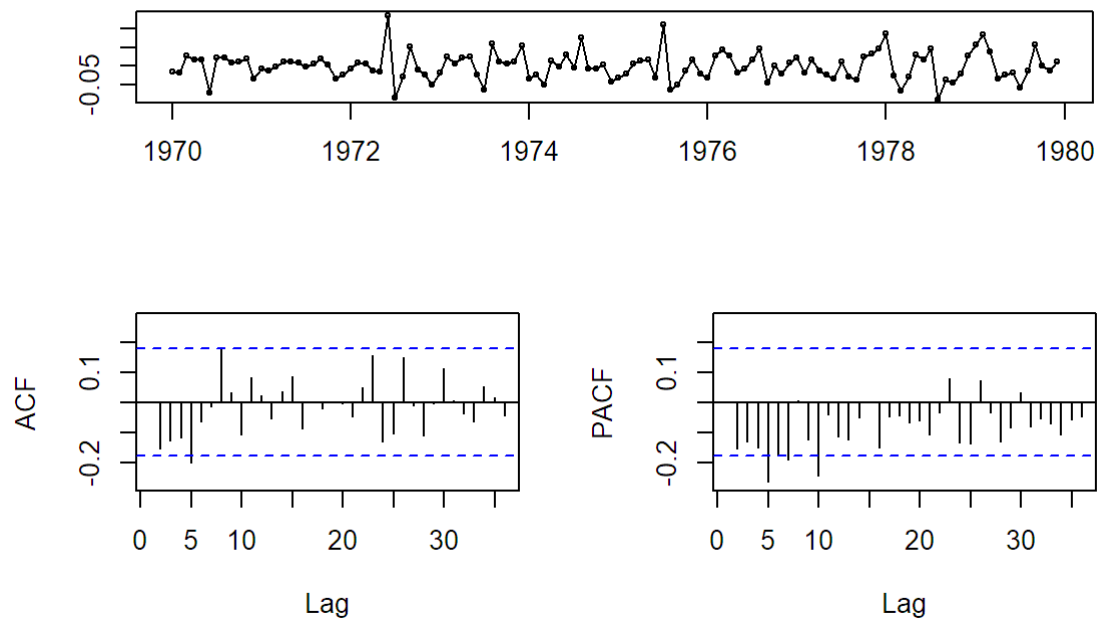
```
#diagnosis
tseries::kpss.test(dec_stl$time.series[, "remainder"], null="Level")
```



```
##  
## KPSS Test for Level Stationarity  
##  
## data: dec_stl$time.series[, "remainder"]  
## KPSS Level = 0.015404, Truncation lag parameter = 4, p-value = 0.1
```

```
tsdisplay(dec_stl$time.series[, "remainder"], null="Level", main="Random_STL model")
```

Random_STL model

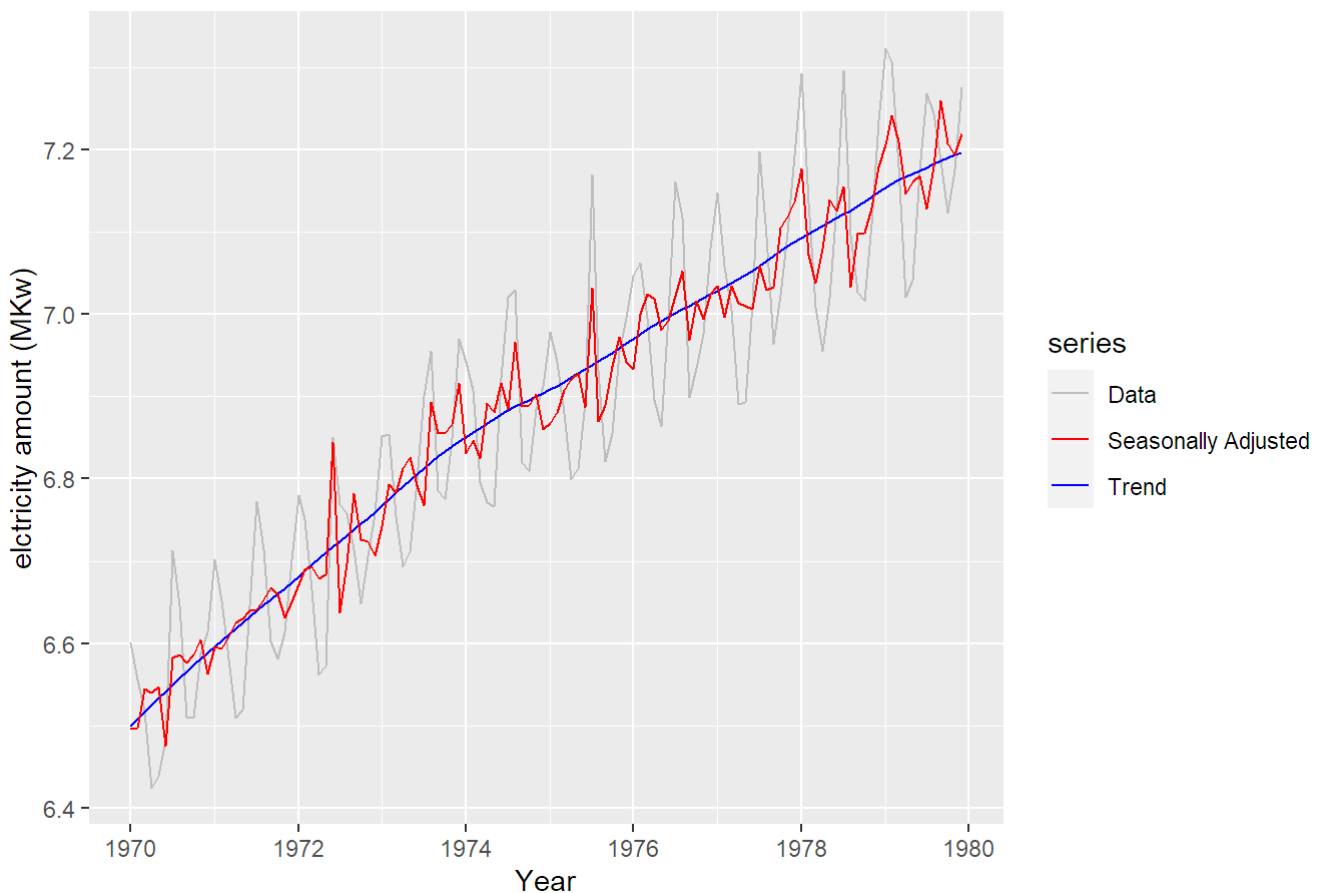


* plot with the original data, seasonally adjusted, trend-cycle component

=> 위의 5가지 모델은 모두 `kpss.test`에서 H_0 :residual is independent에 대해 p-value가 0.1로 귀무가설을 기각하지 못하였습니다. 즉, 모두 정상성을 어느정도 만족한다고 볼 수 있어 그 다음 기준인 ACF와 PACF가 모두 유의수준 내에 있는지에 따라 최종모델로 SEATS decomposition 방법을 이용하여 plot을 그렸습니다.

```
autoplot(data_ts, series="Data") +
  autolayer(trendcycle(dec_seas), series="Trend") +
  autolayer(seasadj(dec_seas), series="Seasonally Adjusted") +
  xlab("Year") + ylab("electricity amount (MKw)") +
  ggtitle("Compare original data, seasonally adjusted, trend-cycle") +
  scale_colour_manual(values=c("gray", "red", "blue"),
    breaks=c("Data", "Seasonally Adjusted", "Trend"))
```

Compare original data, seasonally adjusted, trend-cycle

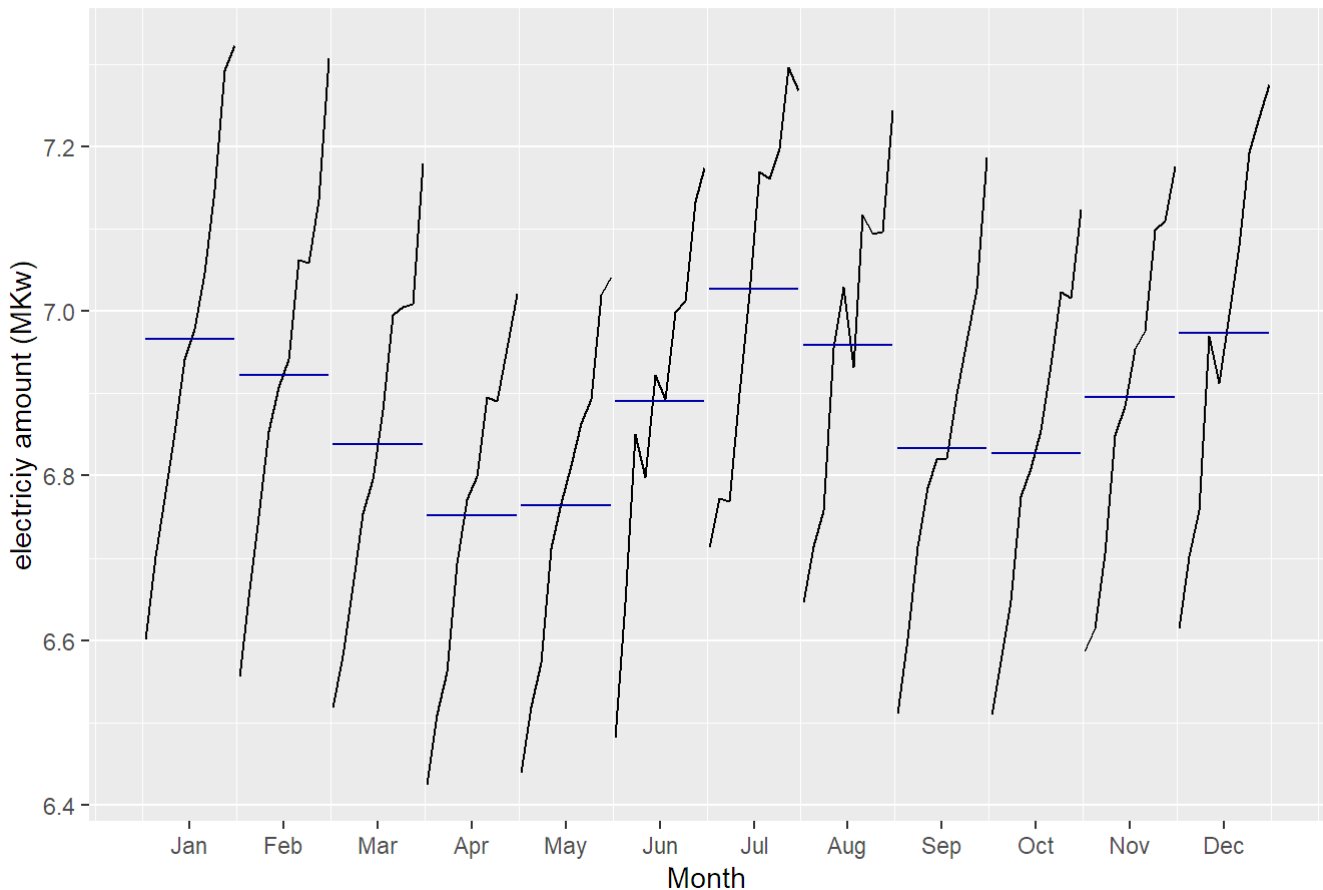


- blue line : electricity data의 trend를 나타내는 것으로 시간이 지날수록 상승세를 보입니다.
- red line : electricity data에서 계절성만을 제외하고, trend와 remainder만을 그린 것으로 gray line과 비교하였을 때, original data에서 계절성이 어느 정도 사라진 것을 알 수 있습니다.

3. Draw sub-series seasonal effect graph in each month.

```
data_ts %>%
  ggsubseriesplot() +
  ylab("electriciy amount (MKw)") +
  ggtitle("Seasonal Subseries Plot : Electricity data")
```

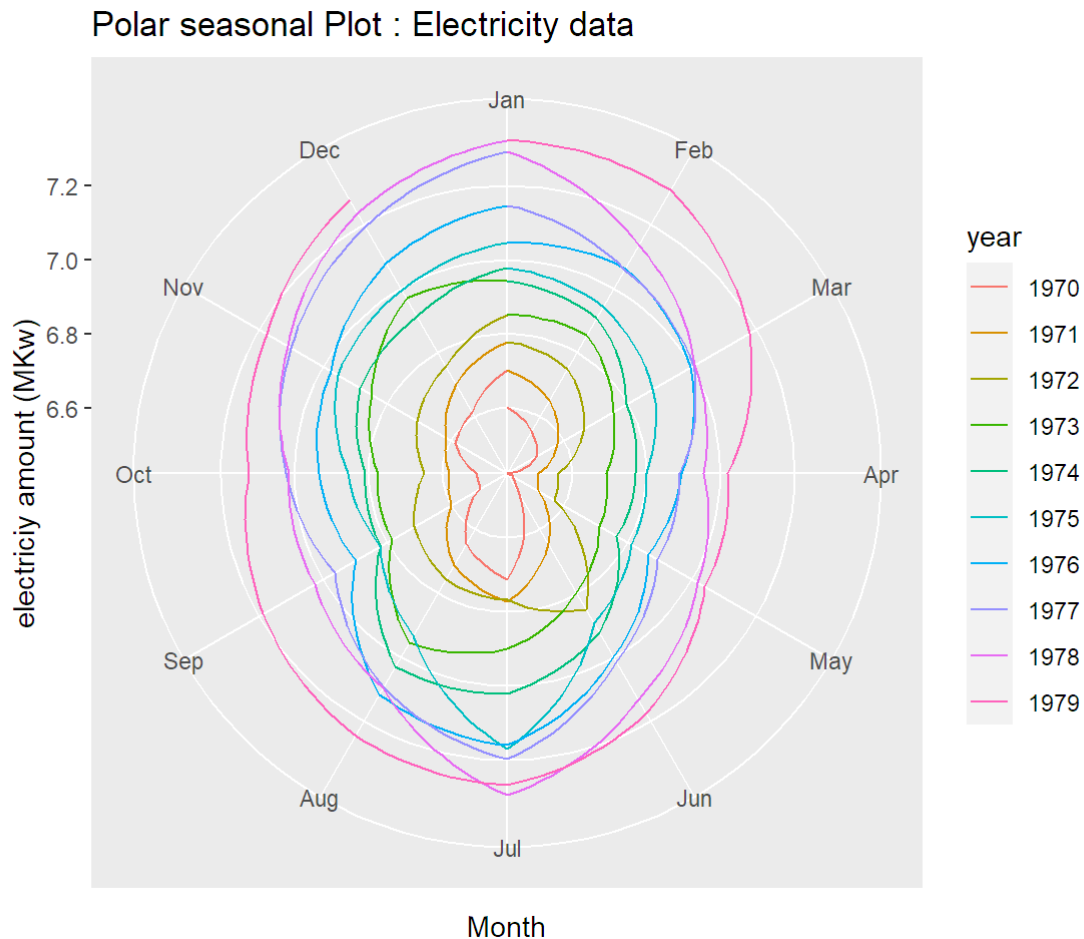
Seasonal Subseries Plot : Electricity data



- Seasonal Subseries Plot은 데이터의 계절성을 확인하기 위해 1970년부터 1979년까지의 데이터를 달(month)별로 함축한 것입니다.
위의 그래프에 따르면, 우선 12월~2월(겨울)에 전기량이 높은 것을 볼 수 있습니다.
이후 4~5월까지 전기량이 줄어들다가 다시 7~8월(여름)에는 전기량이 증가하는 것을 알 수 있습니다.
또한, 9~10월에 다시 전기량이 줄어들며 겨울이 다가오면서 또다시 전기량이 높아집니다.
즉, 달별로 수준의 차이가 있으며 이에 따라 계절성이 존재한다는 것을 알 수 있습니다.

4. Draw a polar seasonal plot.

```
data_ts %>%
  ggseasonplot(polar=TRUE) +
  ylab("electriciy amount (MKw)") +
  ggtitle("Polar seasonal Plot : Electricity data")
```



- Polar Seasonal Plot은 각 달별 전기량을 polar coordinate system(극좌표계)에 연도별로 나타낸 것입니다.
위의 그래프를 보면, 연도가 증가할수록 원형이 커지므로 전기사용량이 증가하는 것을 알 수 있습니다.
또한, 대체적인 원의 모양을 보면 12~2월, 6~8월에 전기사용량을 많으며, 3~5월, 9~11월에는 전기사용량이 적은 것을 알 수 있습니다.
즉, 연도가 증가할수록 전기사용량이 증가하는 추세이며 특히 여름과 겨울에 전기사용량이 높습니다.