Homework 7

Zahlen Zbinden

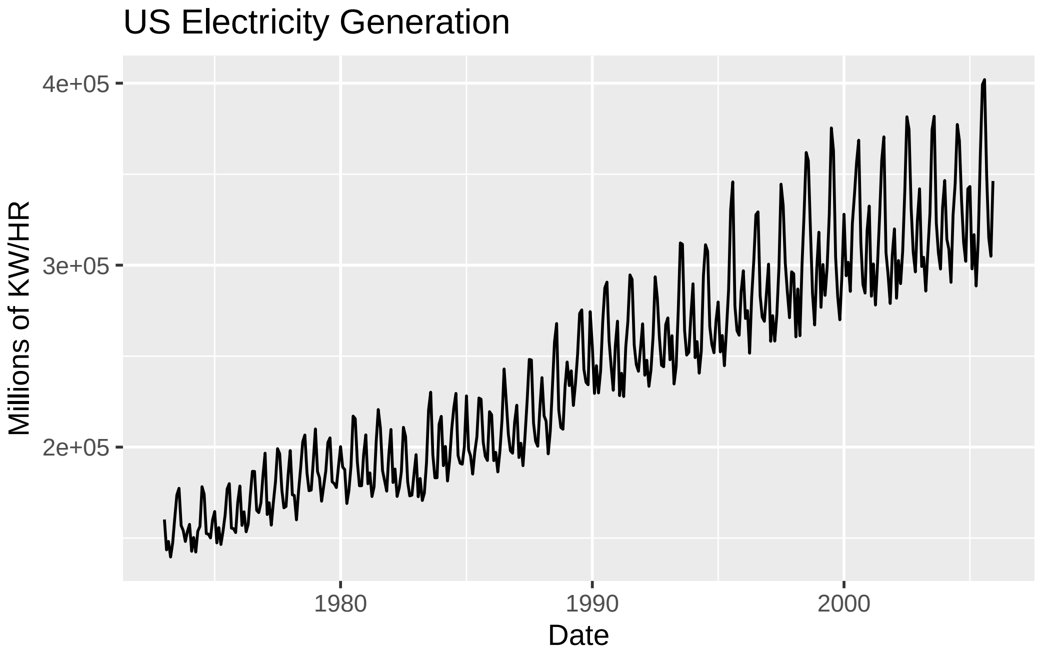
2023-09-24

```
# import data set
data(electricity)
l_elec <- log(electricity)
```

```
start_date <- as.Date("1973-1-1")
end_date <- as.Date("2005-12-31")
```

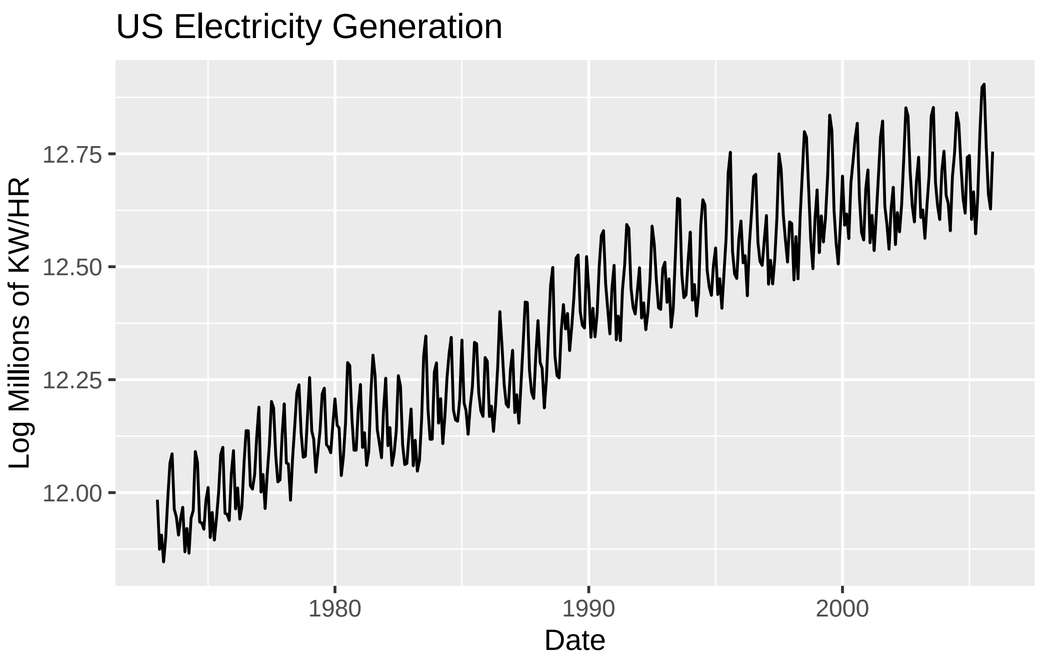
a) Display both the time series plot of the original data and the log transformed series.

```
# plot of original time series
electricity |>
  as_tibble() |>
  mutate(
    date = seq(
      start_date,
      end_date,
      by = "month"
    )
  ) |>
  ggplot(aes(x = date, y = electricity)) +
    geom_line() +
    labs(
      title = "US Electricity Generation",
      x = "Date",
      y = "Millions of KW/HR"
    )
```



We can see from the plot of the original time series that there is increasing variability as time increases, this suggests that we need to make a transformation.

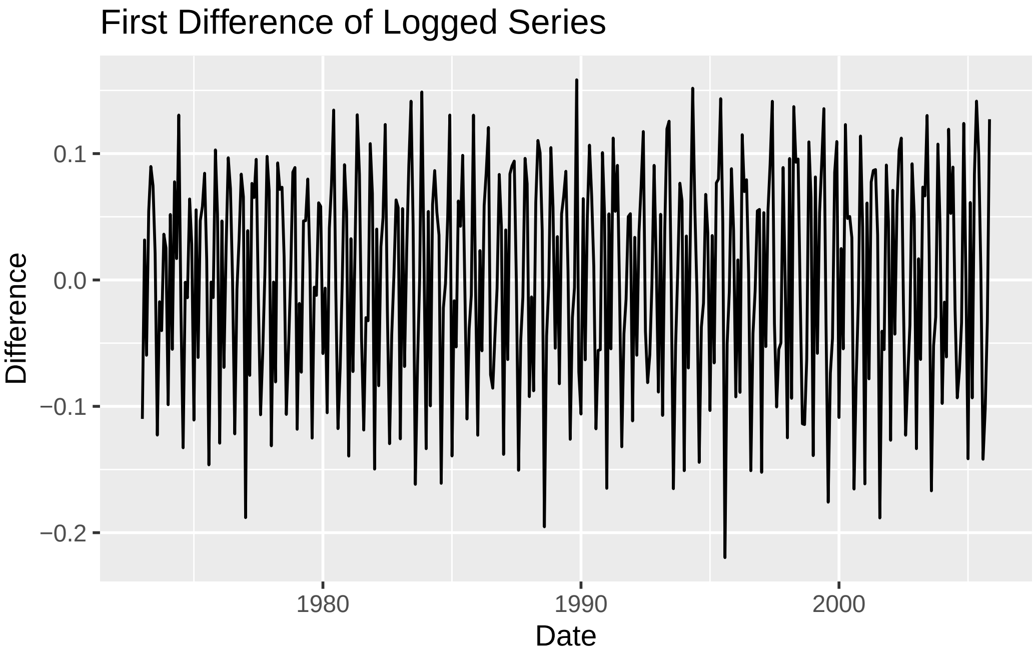
```
l_elec |>
  as_tibble() |>
  mutate(
    date = seq(
      start_date,
      end_date,
      by = "month"
    )
  ) |>
  ggplot(aes(x = date, y = electricity)) +
    geom_line() +
    labs(
      title = "US Electricity Generation",
      x = "Date",
      y = "Log Millions of KW/HR"
    )
```



We can see from the plot of the log transformation that there is much less of an increase in the variability as time increases, this would suggest that taking the log transformation is appropriate for out data.

b) Display and interpret the time series plots of the first difference of the logged series.

```
# first difference of log(electricity)
diff_lelec <- diff(l_elec)

diff_lelec |>
  as_tibble() |>
  mutate(
    date = seq(
      start_date,
      end_date %m-% months(1),
      by = "month"
    )
  ) |>
  ggplot(aes(x = date, y = electricity)) +
    geom_line() +
    labs(
      title = "First Difference of Logged Series",
      x = "Date",
      y = "Difference"
    )
```

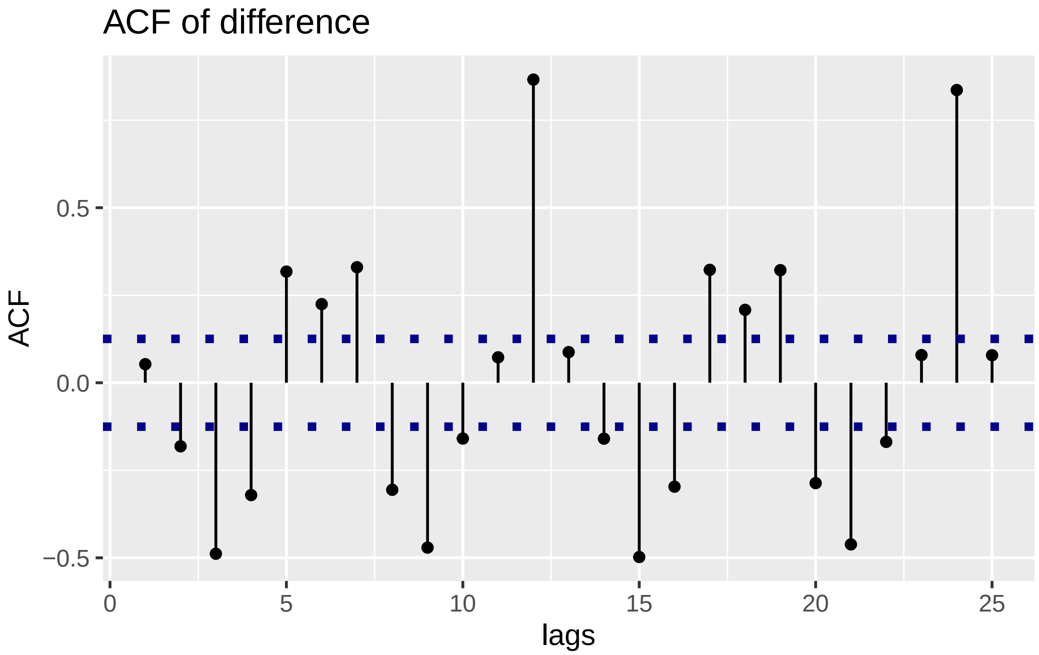


We can see from the first order difference of the log transformed time series that all the values seem to be centered around 0, and it also looks stationary.

If we look at the ACF plot of the series we can uncover more about the data structure.

```
acf_dle <- acf(diff_lelec, max.lag = 36, plot = FALSE)
acf_ci <- qnorm((1 - .05) / 2) / sqrt(length(acf_dle$n.used))
```

```
acf_dle$acf |>
  as_tibble() |>
  mutate(
    lag = 1:length(acf_dle$acf)
  ) |>
  ggplot(aes(x = lag, y = V1)) +
    geom_point() +
    geom_segment(
      aes(
        x = lag,
        xend = lag,
        y = 0,
        yend = V1
      )
    ) +
    geom_hline(
      yintercept = 2 * acf_ci,
      linetype = 3,
      linewidth = 1.5,
      col = "darkblue"
    ) +
    geom_hline(
      yintercept = -2 * acf_ci,
      linetype = 3,
      linewidth = 1.5,
      col = "darkblue"
    ) +
    labs(
      title = "ACF of difference",
      x = "lags",
      y = "ACF"
    )
```



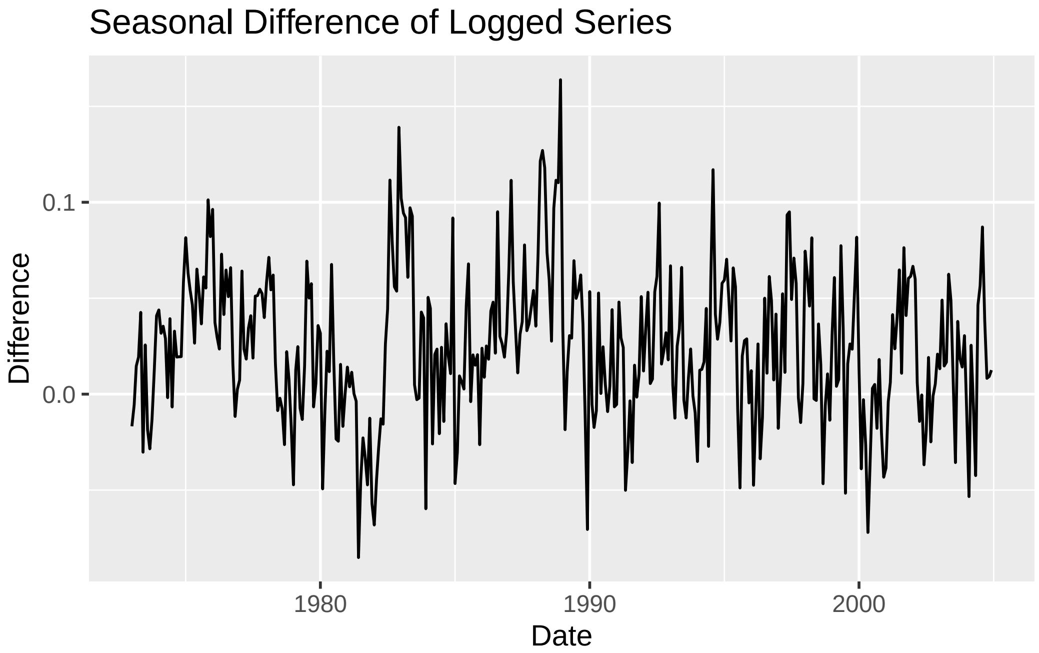
We can see form this ACF that the difference of the data is not white noise. This doesn't imply that the data is not stationary at this point, only that we can't say that it is stationary because it is white noise.

c) Display and iterpret the time series plot of the season difference of the first differnce of the logged series.

I will first look at just doing a seasonal difference and then look at doing a seasonal difference of the first order difference. If we are able to create a stationary time series that is comparable with less differencing (not including the first order difference and only the seasonal differences) we may may want to use that data as it is the less complex of the two.

```
# season difference only
sdiff_lelec <- l_elec |>
                diff(lag = 12)
```

```
sdiff_lelec |>
  as_tibble() |>
  mutate(
    date = seq(
      start_date,
      end_date %m-% months(12),
      by = "month"
    )
  ) |>
  ggplot(aes(x = date, y = electricity)) +
    geom_line() +
    labs(
      title = "Seasonal Difference of Logged Series",
      x = "Date",
      y = "Difference"
    )
```

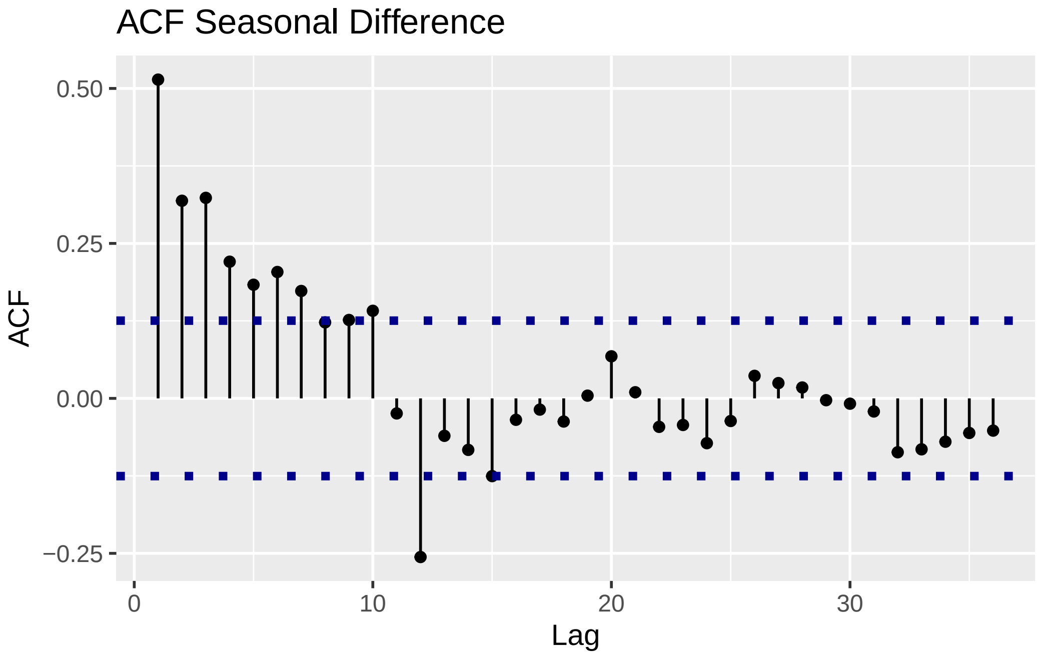


We can see from the plot of the seasonal difference that the data doesn't appear to be better suited at first glance, lets look at the ACF to see if anything else can be uncovered.

```
acf_sdle <- acf(sdiff_lelec, lag.max = 36, plot = FALSE)
acf_ci <- qnorm((1 - .05) / 2) / sqrt(length(acf_sdle$n.used))
```

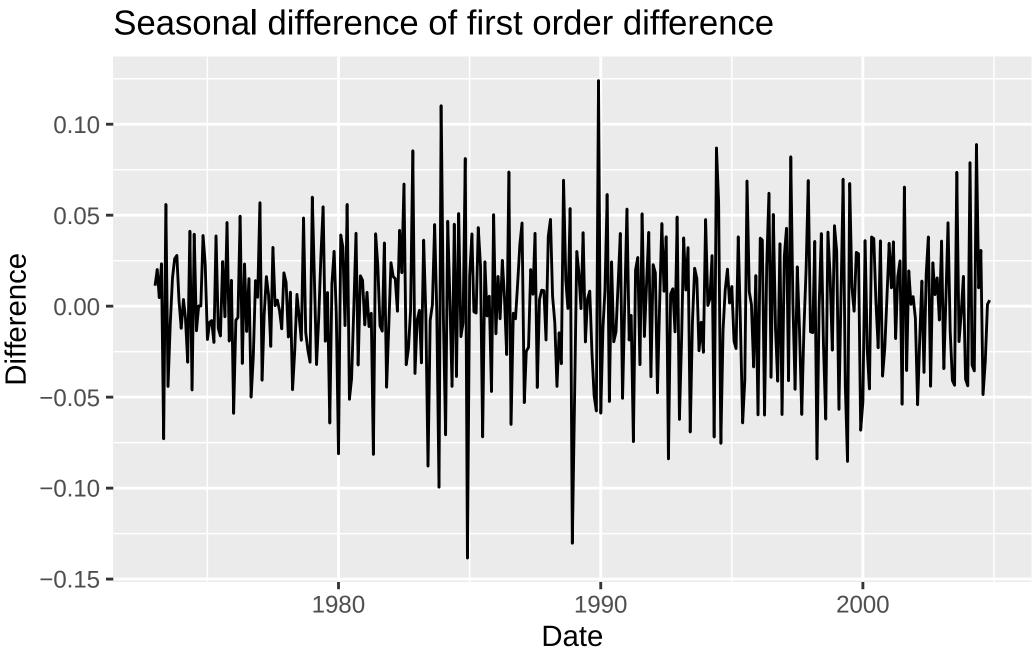
```
acf_sdle$acf |>
  as_tibble() |>
  mutate(
    lag = 1: length(acf_sdle$acf)
```

```r
) |>
ggplot(aes(x = lag, y = V1)) +
  geom_point() +
  geom_segment(
    aes(
      x = lag,
      xend = lag,
      y = 0,
      yend = V1
    )
  ) +
  geom_hline(
    yintercept = 2 * acf_ci,
    linetype = 3,
    linewidth = 1.5,
    col = "darkblue"
  ) +
  geom_hline(
    yintercept = -2 * acf_ci,
    linetype = 3,
    linewidth = 1.5,
    col = "darkblue"
  ) +
  labs(
    title = "ACF Seasonal Difference",
    x = "Lag",
    y = "ACF"
  )
```



The ACF doesn't point out to the data being a white noise process, for our purposes just taking the seasonal difference didn't provide us with better results than just taking a first order differnce so we will take a seasonal difference of the first order difference and compare that against just our first order difference.

```r
# seasonal difference of first order difference
s2diff_lelec <- diff(diff_lelec, lag = 12)

s2diff_lelec |>
  as_tibble() |>
  mutate(
    date = seq(
      start_date,
      end_date %m-% months(13),
      by = "month"
    )
  ) |>
  ggplot(aes(x = date, y = electricity)) +
    geom_line() +
    labs(
      title = "Seasonal difference of first order difference",
      x = "Date",
      y = "Difference"
    )
```



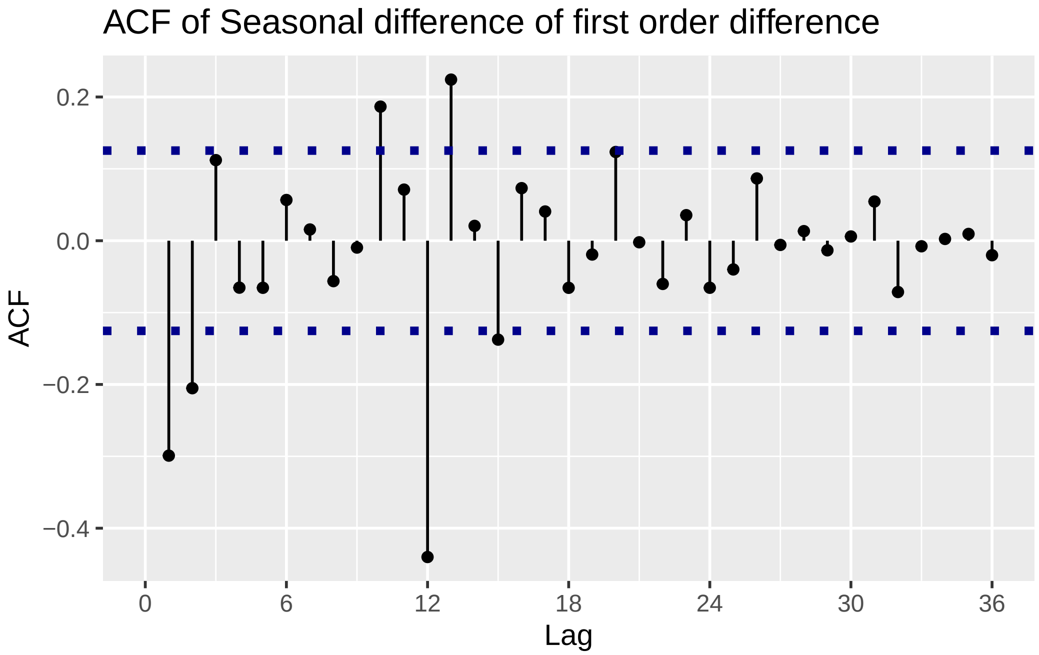
We can see from the plot that the data could be a little better suited for our purpose as there are much less spikes in the model and appears to be more centered around 0.

d)

Lets take a look at the ACF to see if it provides any more insight

```
acf_s2diff <- acf(s2diff_lelec, lag.max = 36, plot = FALSE)
acf_ci <- qnorm((1 - .05) / 2) / sqrt(length(acf_s2diff$n.used))
```

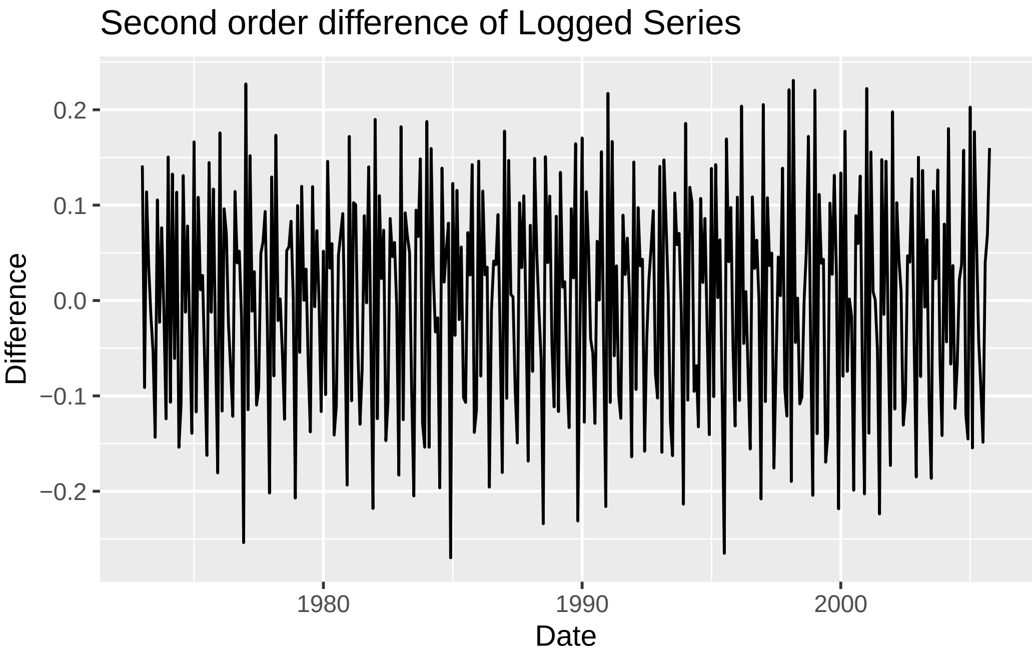
```
acf_plot_s2diff <- acf_s2diff$acf |>
                    as_tibble() |>
                    mutate(
                      lag = 1: length(acf_s2diff$acf)
                    ) |>
                    ggplot(aes(x = lag, y = V1)) +
                      geom_point() +
                      geom_segment(
                        aes(
                          x = lag,
                          xend = lag,
                          y = 0,
                          yend = V1
                        )
                      ) +
                      geom_hline(
                        yintercept = 2 * acf_ci,
                        linetype = 3,
                        linewidth = 1.5,
                        col = "darkblue"
                      ) +
                      geom_hline(
                        yintercept = -2 * acf_ci,
                        linetype = 3,
                        linewidth = 1.5,
                        col = "darkblue"
                      ) +
                      scale_x_continuous(limits = c(0, 36), breaks = seq(0, 36, 6)) +
                      labs(
                        title = "ACF of Seasonal difference of first order difference",
                        x = "Lag",
                        y = "ACF"
                      )

acf_plot_s2diff
```



Compared to the ACF's of the other transformed data sets this one appears to have the closest ACF to that of a white noise process, with much fewer significant spikes as lags increase. It also doesn't have any of the hallmarks of non-stationary data such as significant lag decay.

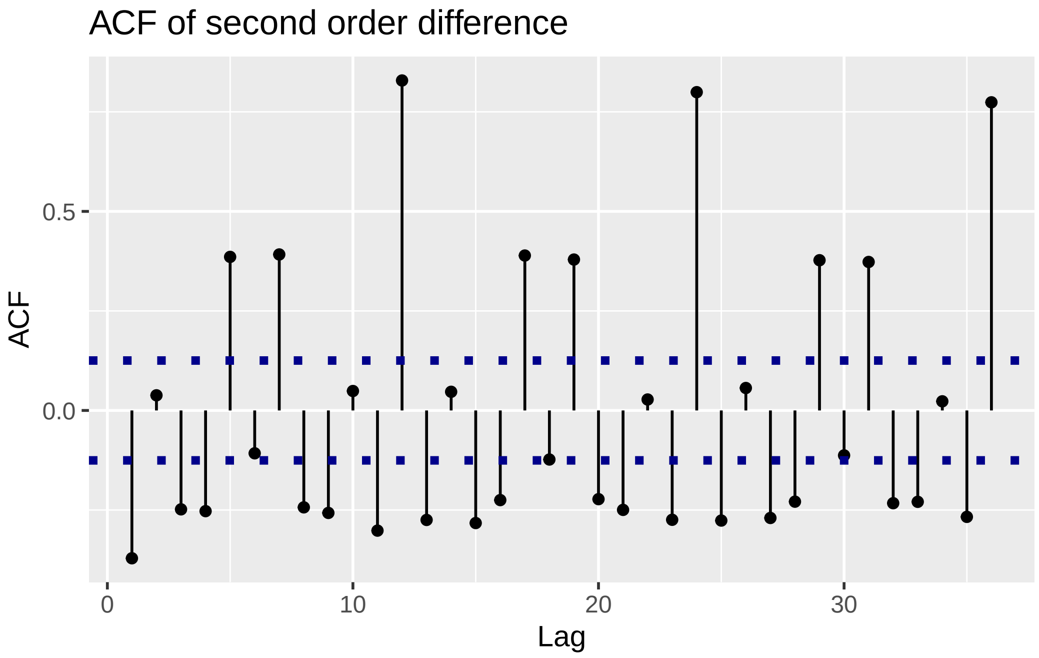
Before we move on, lets take a quick detour and check on the second order difference of the log transformed data set.

```r
diff2 <- diff(l_elec, differences = 2)

l_elec |>
  diff(differences = 2) |>
  as_tibble() |>
  mutate(
    date = seq(
      start_date,
      end_date %m-% months(2),
      by = "month"
    )
  ) |>
  ggplot(aes(x = date, y = electricity)) +
    geom_line() +
    labs(
      title = "Second order difference of Logged Series",
      x = "Date",
      y = "Difference"
    )
```



At first glance this data set looks comparable to that of just the first order difference, however lets investigate the acf.

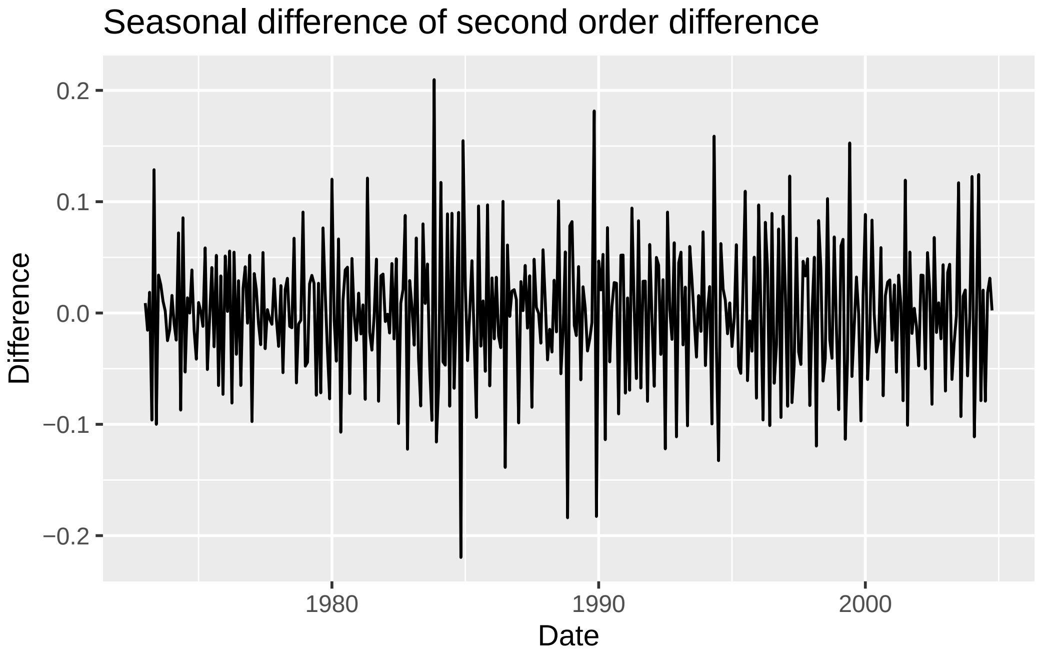
```
acf_2diff <- acf(diff(l_elec, differences = 2), lag.max = 36, plot = FALSE)
acf_ci <- qnorm((1 - .05) / 2) / sqrt(length(acf_s2diff$n.used))

acf_2diff$acf |>
  as_tibble() |>
  mutate(
    lag = 1: length(acf_2diff$acf)
  ) |>
  ggplot(aes(x = lag, y = V1)) +
    geom_point() +
    geom_segment(
      aes(
        x = lag,
        xend = lag,
        y = 0,
        yend = V1
      )
    ) +
    geom_hline(
      yintercept = 2 * acf_ci,
      linetype = 3,
      linewidth = 1.5,
      col = "darkblue"
    ) +
    geom_hline(
      yintercept = -2 * acf_ci,
      linetype = 3,
      linewidth = 1.5,
      col = "darkblue"
    ) +
    labs(
      title = "ACF of second order difference",
      x = "Lag",
      y = "ACF"
    )
```



This ACF doesn't look anymore promising than anything we have considered so far, lets also take a look at the seasonal lag of the second order difference.

```
s22_diff <- diff(diff2, lag = 12)
```

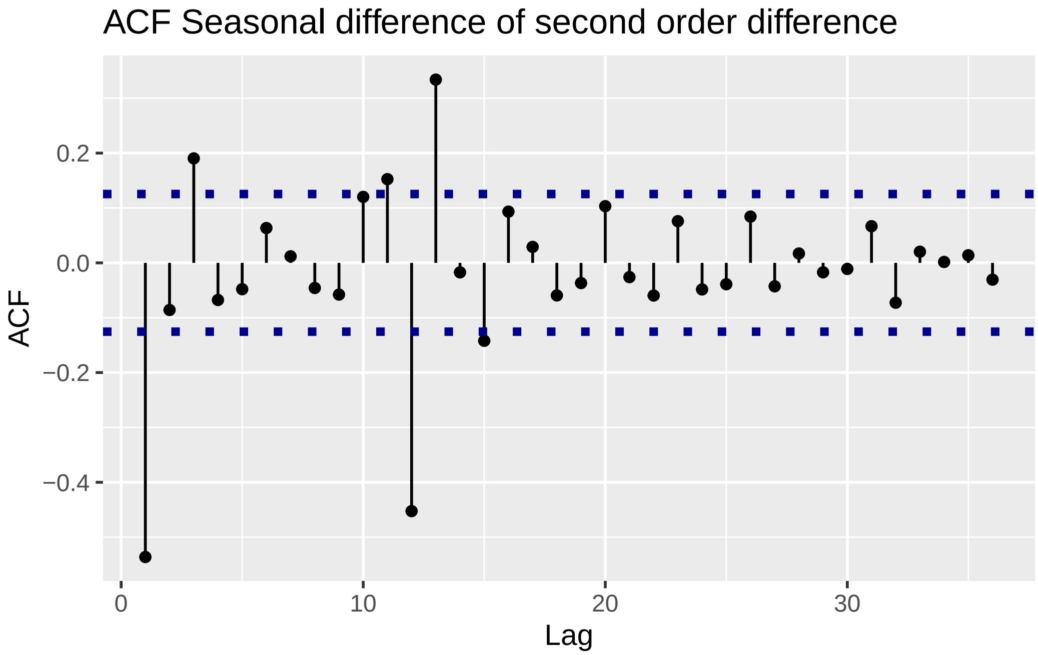
```
s22_diff |>
  as_tibble() |>
  mutate(
    date = seq(
      start_date,
      end_date %m-% months(14),
      by = "month"
    )
  ) |>
  ggplot(aes(x = date, y = electricity)) +
    geom_line() +
    labs(
      title = "Seasonal difference of second order difference",
      x = "Date",
      y = "Difference"
    )
```



This plot is about the same as our previous plots, however the data is more condensed around 0 with less spikes, less also take a look at the ACF of this dataset.

```
acf_s22diff <- acf(s22_diff, lag.max = 36, plot = FALSE)
acf_ci <- qnorm((1 - .05) / 2) / sqrt(length(acf_s22diff$n.used))
```

```
acf_s22diff$acf |>
  as_tibble() |>
  mutate(
    lag = 1: length(acf_s22diff$acf)
  ) |>
  ggplot(aes(x = lag, y = V1)) +
    geom_point() +
    geom_segment(
      aes(
        x = lag,
        xend = lag,
        y = 0,
        yend = V1
      )
    ) +
    geom_hline(
      yintercept = 2 * acf_ci,
      linetype = 3,
      linewidth = 1.5,
      col = "darkblue"
    ) +
    geom_hline(
      yintercept = -2 * acf_ci,
      linetype = 3,
      linewidth = 1.5,
      col = "darkblue"
    ) +
    labs(
      title = "ACF Seasonal difference of second order difference",
      x = "Lag",
      y = "ACF"
    )
```

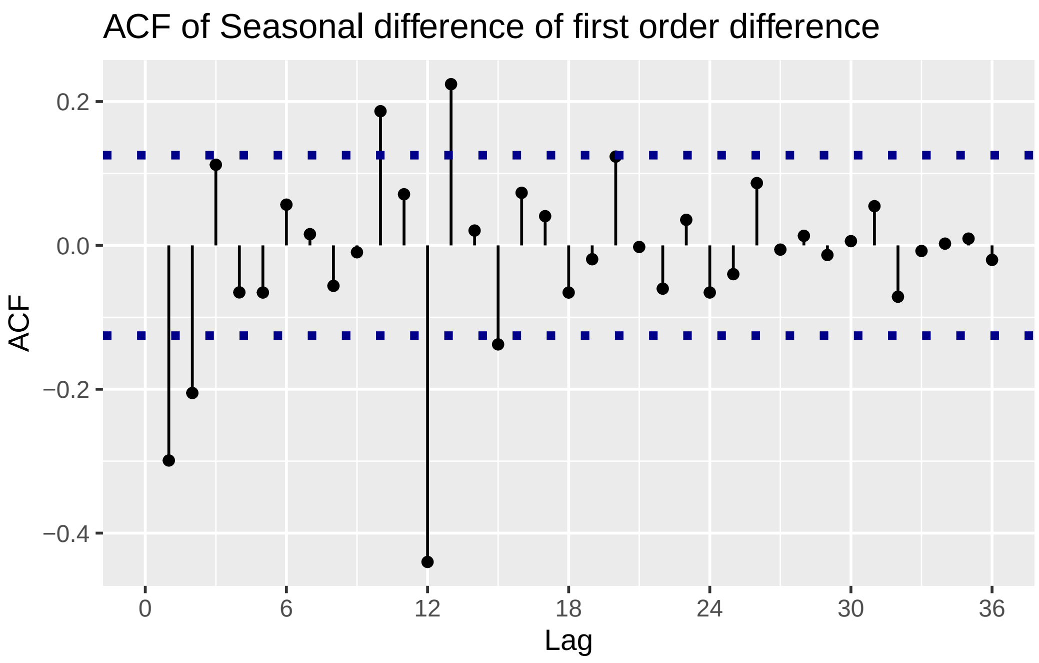


This ACF also doesn't show any hallmark traits of being white noise, we should pick the less complicated data set as we continue to avoid any sort of overdifferencing.

e) Fit a multiplicative season ARIMA model to the logged series. What model do you choose and why?

First recall the ACF plot of the seasonal difference of the first order difference of the log transformed time series.

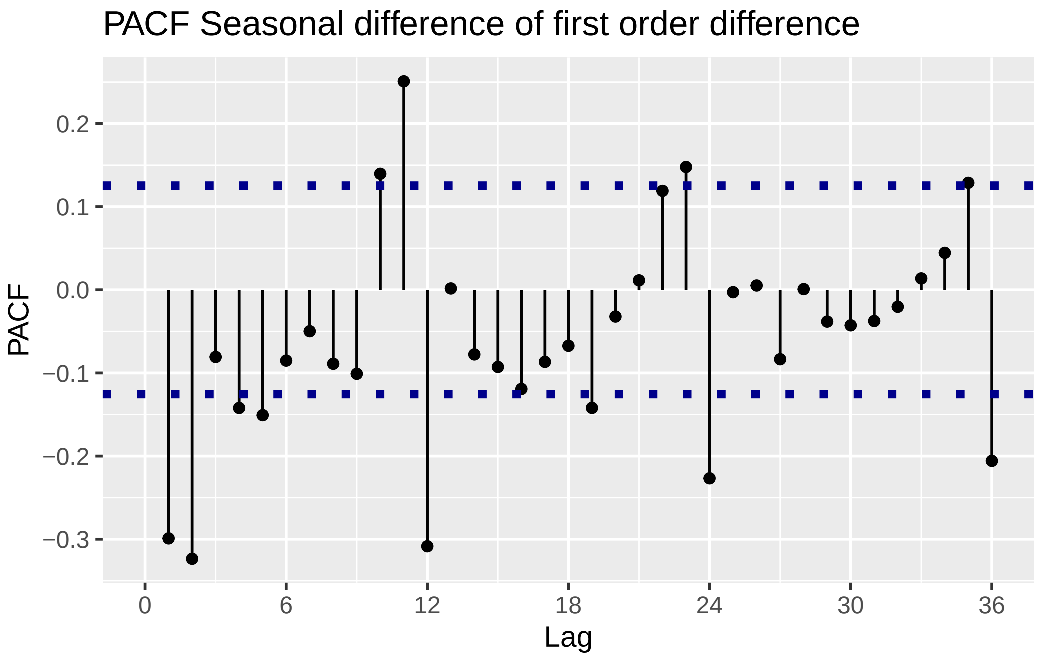
```
acf_plot_s2diff
```



Now lets take a look at the PACF for this data set as well

```
pacf_s2diff <- pacf(s2diff_lelec, lag.max = 36, plot = FALSE)
pacf_ci <- qnorm((1 - .05) / 2) / sqrt(length(pacf_s2diff$n.used))
```

```
pacf_s2diff$acf |>
  as_tibble() |>
  mutate(
    lag = 1: length(pacf_s2diff$acf)
  ) |>
  ggplot(aes(x = lag, y = V1)) +
    geom_point() +
    geom_segment(
      aes(
        x = lag,
        xend = lag,
        y = 0,
        yend = V1
      )
    ) +
    geom_hline(
      yintercept = 2 * pacf_ci,
      linetype = 3,
      linewidth = 1.5,
      col = "darkblue"
    ) +
    geom_hline(
      yintercept = -2 * pacf_ci,
      linetype = 3,
      linewidth = 1.5,
      col = "darkblue"
    ) +
    scale_x_continuous(limits = c(0, 36), breaks = seq(0, 36, 6)) +
    labs(
      title = "PACF Seasonal difference of first order difference",
      x = "Lag",
      y = "PACF"
    )
```



Lets first consider the seasonal part of our model.

We can see from the ACF at the easonal lags (12, 24, 36) there is a spike at lag 12, and a cutoff after that. That would suggest that we could pick an MA(1) model for the seasonal portion of our SARIMA model.

When looking at the seasonal lags of the PACF we see significant lags all the way out to 36 which may suggest that an AR(3) or if we looked into higher values of lags even an AR(4) or AR(5) model, however that would create a lot of parameters in our model and we don't want to over complicate it unecessarily.

Now lets consider the non-seasonal part of our model. (lags < 12)

For the ACF we have significant lags at 2, and 10, this may suggest that an MA(2) model could be a good candidate.

For the PACF we see significant lags at 2, and perhaps something at 10, but that may just be from a type 1 error rate. The PACF is more conclusive to me for the non-seasonal part of our model, AR(2)

From this I will try a couple different models. For all models chosen d & D will be 1 as we took one first order difference of the data, and one seasonal difference of the data

The first model that I will select is a SARIMA(2,1,0)(0,1,1), this is the most simple model that can be gleened form the insight of the ACF and PACF alone.

I will also look at a couple of different models that I will show below just to conduct a good search.

SARIMA(2,1,1)(0,1,1)

SARIMA(2,1,1)(1,1,1)

I have chosed to search an additional seasonal AR(p) model becuase the PACF isn't conclusive as to what would be a good order for an AR model in this situtation.

We will also use the auto.arima() function to determine if there is another model that may be suitable for our investigation.

```
auto.arima(l_elec, stepwise = FALSE, approximation = FALSE)
```

```
Series: l_elec
ARIMA(2,0,1)(0,1,1)[12] with drift

Coefficients:
         ar1      ar2      ma1     sma1   drift
      1.3935  -0.4094  -0.8325  -0.8526  0.0021
s.e.  0.0762   0.0708   0.0476   0.0309  0.0002

sigma^2 = 0.0006796:  log likelihood = 851.84
AIC=-1691.69   AICc=-1691.46   BIC=-1667.98
```

From this function, we have chosen to do an extensive search with stepwise = FALSE and approximation = FALSE. This should perform a large number of calculations to choose a model that has the best AICc for out data set. There is no need to put the differenced dataset into the function as it will search for the best differences to apply.

We can see from that it has chosen an SARIMA(2,0,1)(0,1,1)12 model, with an AICc of -1691.46 and log likelihood of 851.84.

f) Investigate diagnostics for this model, including autocorrelation and normality of the residuals.

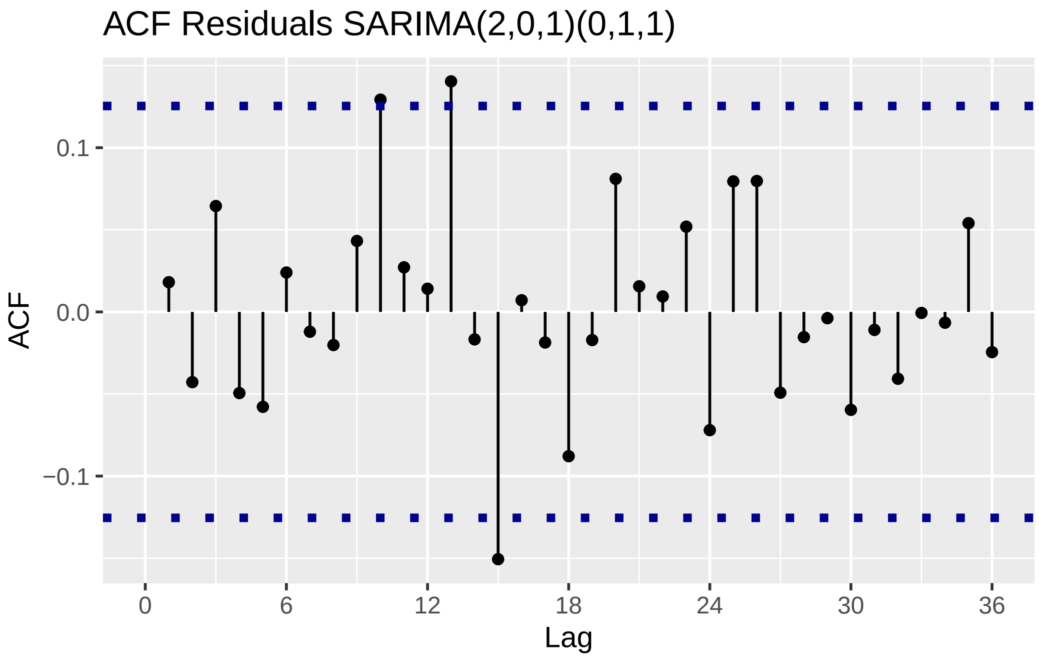
SARIMA(2,0,1)(0,1,1)

```
# fit auto model
fit_sarima201_011 <- auto.arima(l_elec, stepwise = FALSE, approximation = FALSE)
```

```
res_201011 <- fit_sarima201_011$residuals
```

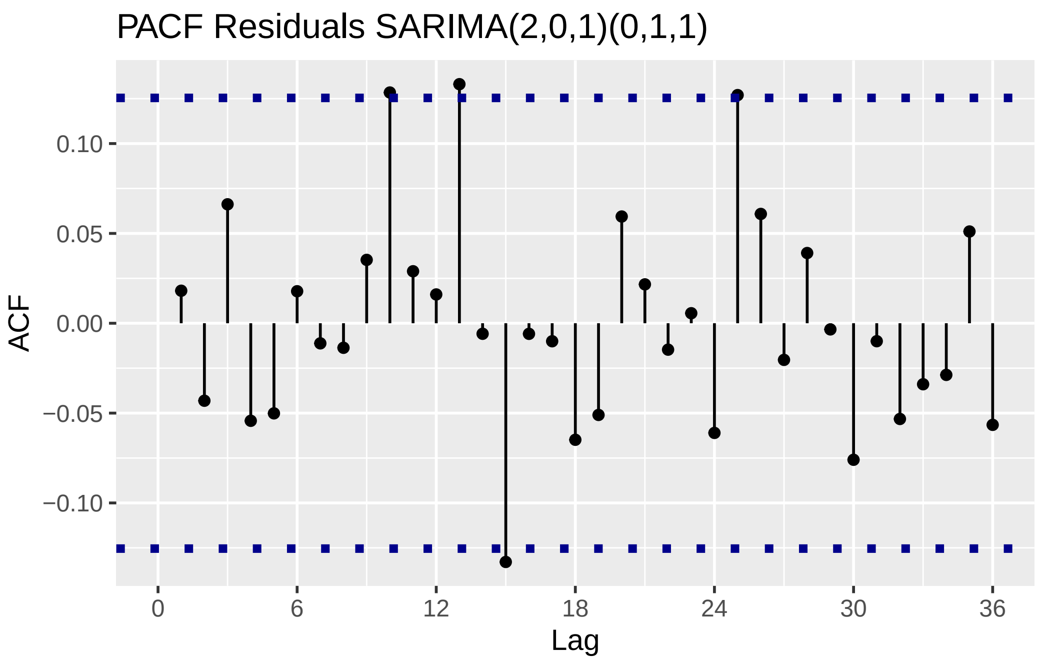
```
# check acf of residuals
res_acf1 <- acf(res_201011, lag.max = 36, plot = FALSE)
acf_ci1 <- qnorm((1 - .05) / 2) / sqrt(length(res_acf1$n.used))
```

```
res_acf1$acf |>
  as_tibble() |>
  mutate(
    lag = 1: length(res_acf1$acf)
  ) |>
  ggplot(aes(x = lag, y = V1)) +
    geom_point() +
    geom_segment(
      aes(
        x = lag,
        xend = lag,
        y = 0,
        yend = V1
      )
    ) +
    geom_hline(
      yintercept = 2 * acf_ci1,
      linetype = 3,
      linewidth = 1.5,
      col = "darkblue"
    ) +
    geom_hline(
      yintercept = -2 * acf_ci1,
      linetype = 3,
      linewidth = 1.5,
      col = "darkblue"
    ) +
    scale_x_continuous(limits = c(0, 36), breaks = seq(0, 36, 6)) +
    labs(
      title = "ACF Residuals SARIMA(2,0,1)(0,1,1)",
      x = "Lag",
      y = "ACF"
    )
```



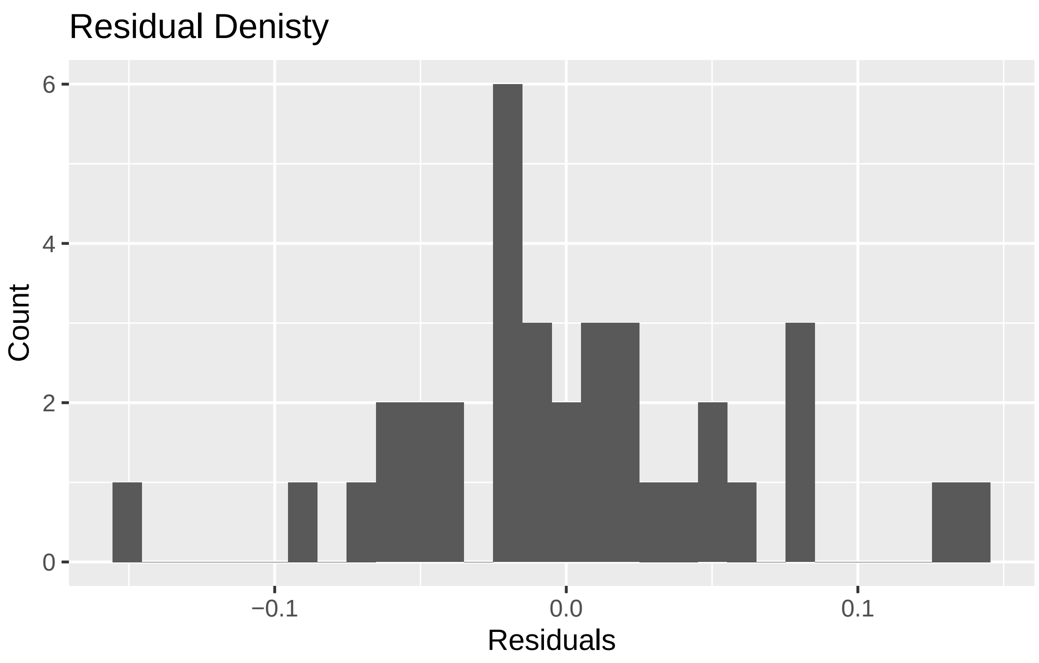
```
# check acf of residuals
res_pacf1 <- pacf(res_201011, lag.max = 36, plot = FALSE)
pacf_ci1 <- qnorm((1 - .05) / 2) / sqrt(length(res_pacf1$n.used))
```

```r
res_pacf1$acf |>
  as_tibble() |>
  mutate(
    lag = 1: length(res_pacf1$acf)
  ) |>
  ggplot(aes(x = lag, y = V1)) +
    geom_point() +
    geom_segment(
      aes(
        x = lag,
        xend = lag,
        y = 0,
        yend = V1
      )
    ) +
    geom_hline(
      yintercept = 2 * pacf_ci1,
      linetype = 3,
      linewidth = 1.5,
      col = "darkblue"
    ) +
    geom_hline(
      yintercept = -2 * pacf_ci1,
      linetype = 3,
      linewidth = 1.5,
      col = "darkblue"
    ) +
    scale_x_continuous(limits = c(0, 36), breaks = seq(0, 36, 6)) +
    labs(
      title = "PACF Residuals SARIMA(2,0,1)(0,1,1)",
      x = "Lag",
      y = "ACF"
    )
```



```r
res_acf1$acf |>
  as_tibble() |>
  ggplot(aes(x = V1)) +
    geom_histogram() +
    labs(
      title = "Residual Denisty",
      x = "Residuals",
      y = "Count"
    )
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



We can see from the plots of the ACF and PACF that the residuals do not show any trend as lags increase, this suggests that they are indeed random. We can also see from the historgram that the density is approximately Normal.

This model behaves well.

SARIMA(2,1,1)X(0,1,1)

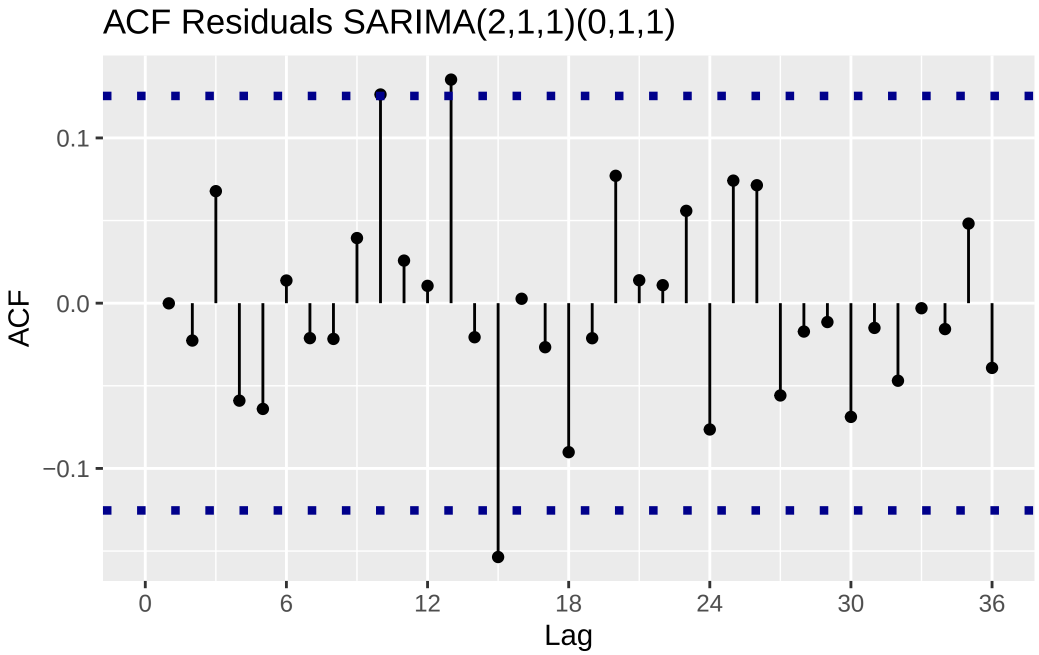
```
# create model
fit_sarima211_011 <- Arima(
  l_elec,
  order = c(2,1,1),
  seasona = c(0,1,1)
)
```

```
# pull out residuals
res_211011 <- fit_sarima211_011$residuals
```

```
# check acf of residuals
res_acf2 <- acf(res_211011, lag.max = 36, plot = FALSE)
```

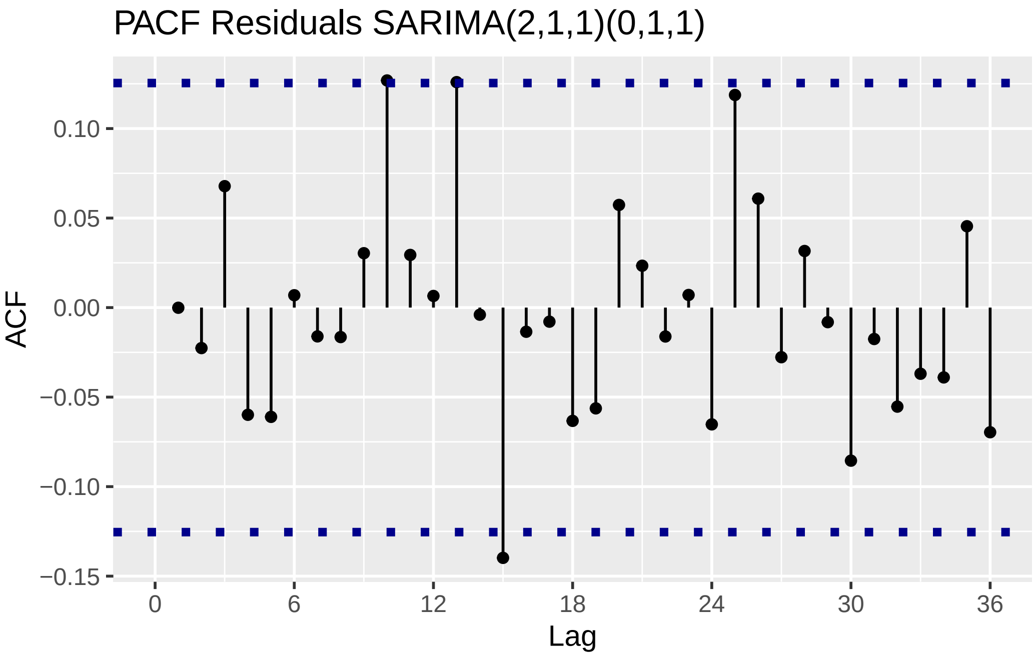
```
acf_ci2 <- qnorm((1 - .05) / 2) / sqrt(length(res_acf2$n.used))
```

```
res_acf2$acf |>
  as_tibble() |>
  mutate(
    lag = 1: length(res_acf2$acf)
  ) |>
  ggplot(aes(x = lag, y = V1)) +
    geom_point() +
    geom_segment(
      aes(
        x = lag,
        xend = lag,
        y = 0,
        yend = V1
      )
    ) +
    geom_hline(
      yintercept = 2 * acf_ci2,
      linetype = 3,
      linewidth = 1.5,
      col = "darkblue"
    ) +
    geom_hline(
      yintercept = -2 * acf_ci2,
      linetype = 3,
      linewidth = 1.5,
      col = "darkblue"
    ) +
    scale_x_continuous(limits = c(0, 36), breaks = seq(0, 36, 6)) +
    labs(
      title = "ACF Residuals SARIMA(2,1,1)(0,1,1)",
      x = "Lag",
      y = "ACF"
    )
```



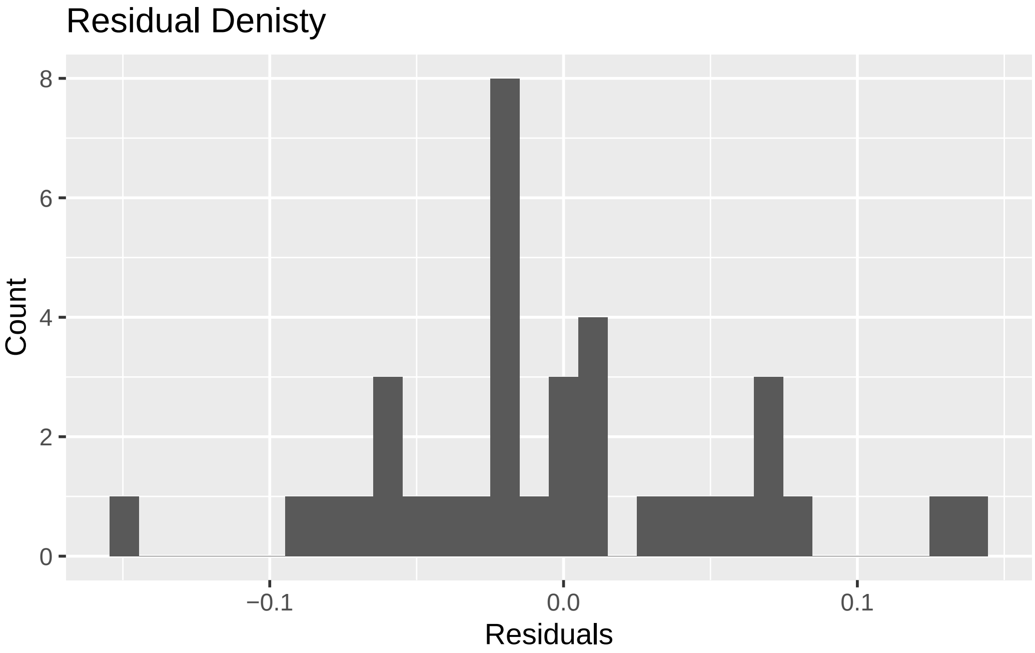
```
# check acf of residuals
res_pacf2 <- pacf(res_211011, lag.max = 36, plot = FALSE)
pacf_ci2 <- qnorm((1 - .05) / 2) / sqrt(length(res_pacf2$n.used))
```

```
res_pacf2$acf |>
  as_tibble() |>
  mutate(
    lag = 1: length(res_pacf2$acf)
  ) |>
  ggplot(aes(x = lag, y = V1)) +
    geom_point() +
    geom_segment(
      aes(
        x = lag,
        xend = lag,
        y = 0,
        yend = V1
      )
    ) +
    geom_hline(
      yintercept = 2 * pacf_ci2,
      linetype = 3,
      linewidth = 1.5,
      col = "darkblue"
    ) +
    geom_hline(
      yintercept = -2 * pacf_ci2,
      linetype = 3,
      linewidth = 1.5,
      col = "darkblue"
    ) +
    scale_x_continuous(limits = c(0, 36), breaks = seq(0, 36, 6)) +
    labs(
      title = "PACF Residuals SARIMA(2,1,1)(0,1,1)",
      x = "Lag",
      y = "ACF"
    )
```



```
res_acf2$acf |>
  as_tibble() |>
  ggplot(aes(x = V1)) +
    geom_histogram() +
      labs(
        title = "Residual Denisty",
        x = "Residuals",
        y = "Count"
      )
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



We can see from the plots of the ACF and PACF that the residuals do not show any trend as lags increase, this suggests that they are indeed random. We can also see from the historgram that the density is not as "normal" as the density from the previous model, with a large amount of the residuals being slightly negative, although it is still normal, just a higher concentration.

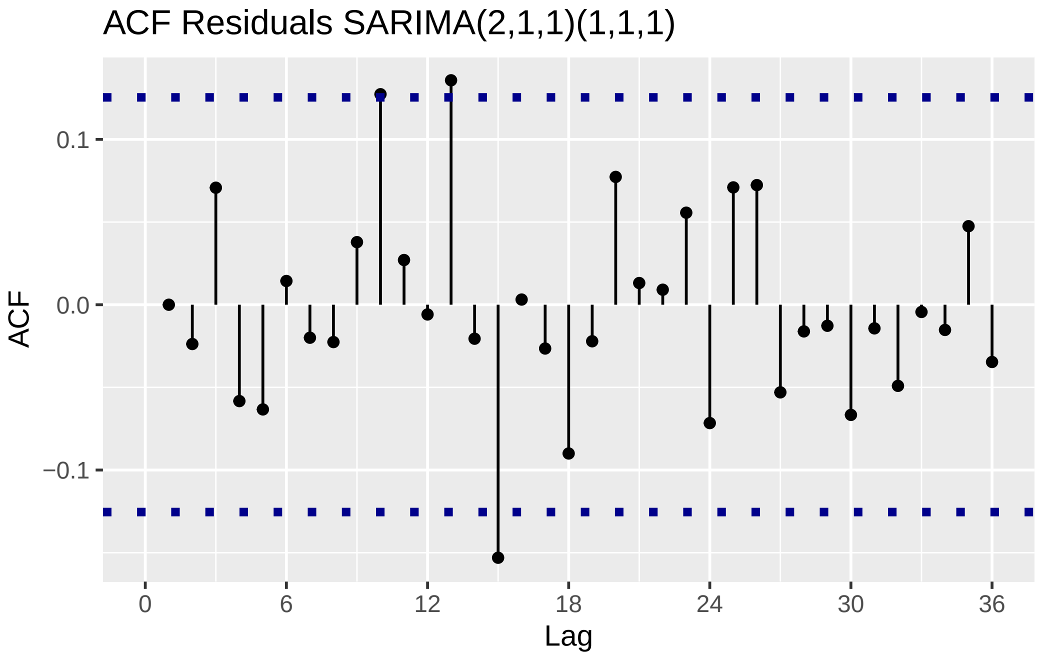
SARIMA(2,1,1)(1,1,1)

```
# create model
fit_sarima211_111 <- Arima(
  l_elec,
  order = c(2,1,1),
  seasona = c(1,1,1)
)
```

```
# pull out residuals
res_211111 <- fit_sarima211_111$residuals
```

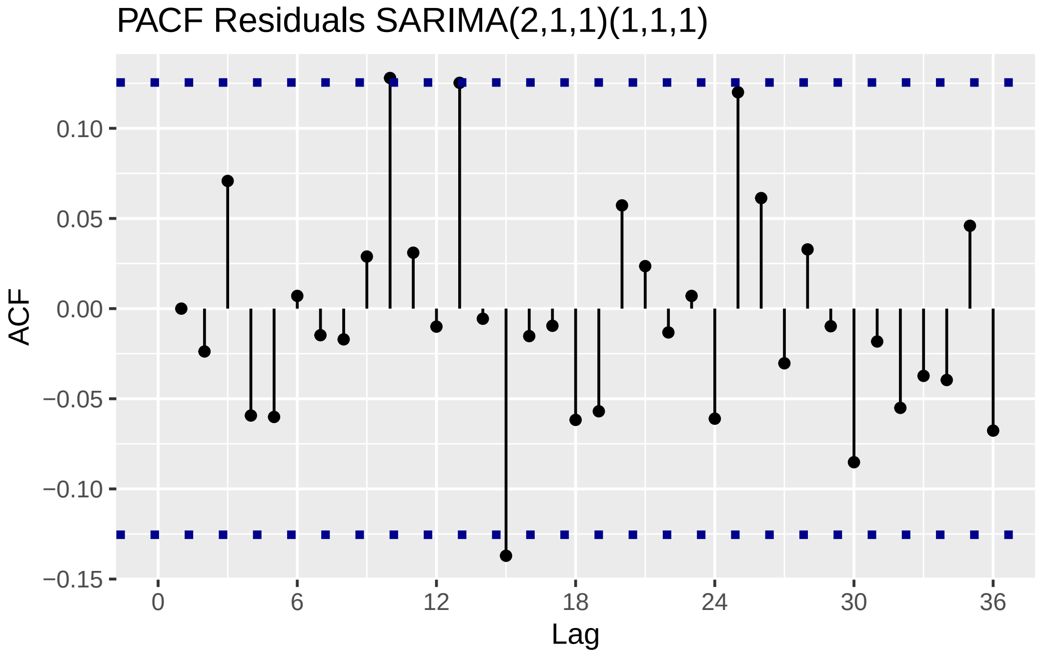
```
# check acf of residuals
res_acf3 <- acf(res_211111, lag.max = 36, plot = FALSE)
acf_ci3 <- qnorm((1 - .05) / 2) / sqrt(length(res_acf3$n.used))
```

```
res_acf3$acf |>
  as_tibble() |>
  mutate(
    lag = 1: length(res_acf3$acf)
  ) |>
  ggplot(aes(x = lag, y = V1)) +
    geom_point() +
    geom_segment(
      aes(
        x = lag,
        xend = lag,
        y = 0,
        yend = V1
      )
    ) +
    geom_hline(
      yintercept = 2 * acf_ci3,
      linetype = 3,
      linewidth = 1.5,
      col = "darkblue"
    ) +
    geom_hline(
      yintercept = -2 * acf_ci3,
      linetype = 3,
      linewidth = 1.5,
      col = "darkblue"
    ) +
    scale_x_continuous(limits = c(0, 36), breaks = seq(0, 36, 6)) +
    labs(
      title = "ACF Residuals SARIMA(2,1,1)(1,1,1)",
      x = "Lag",
    y = "ACF"
  )
```



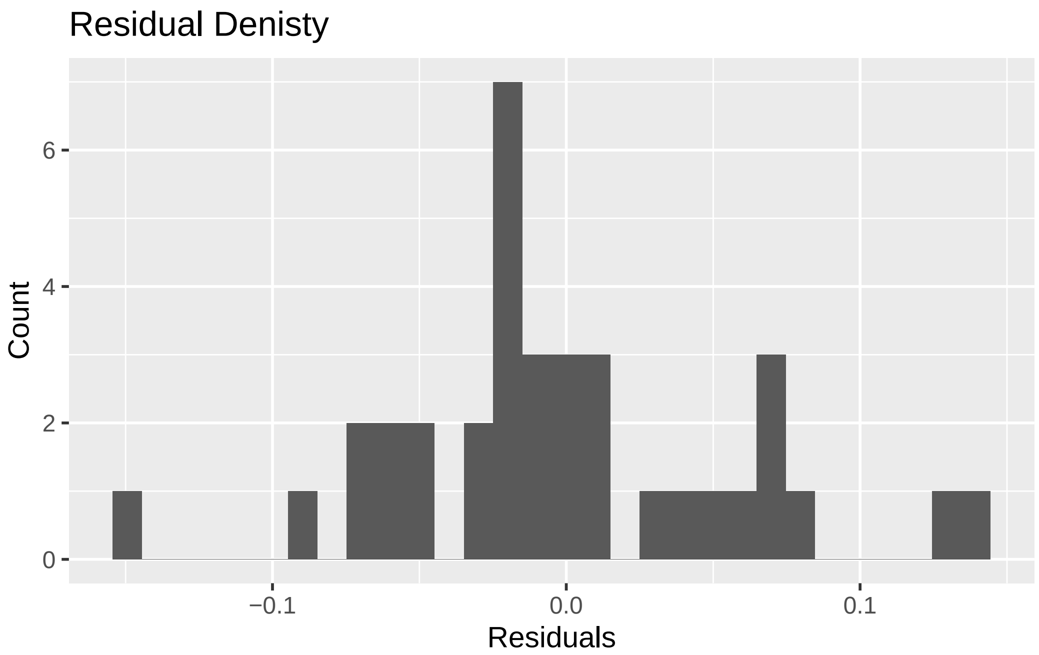
```
# check acf of residuals
res_pacf3 <- pacf(res_211111, lag.max = 36, plot = FALSE)
pacf_ci3 <- qnorm((1 - .05) / 2) / sqrt(length(res_pacf3$n.used))
```

```
res_pacf3$acf |>
  as_tibble() |>
  mutate(
    lag = 1: length(res_pacf3$acf)
  ) |>
  ggplot(aes(x = lag, y = V1)) +
    geom_point() +
    geom_segment(
      aes(
        x = lag,
        xend = lag,
        y = 0,
        yend = V1
      )
) +
geom_hline(
  yintercept = 2 * pacf_ci3,
  linetype = 3,
  linewidth = 1.5,
  col = "darkblue"
) +
geom_hline(
  yintercept = -2 * pacf_ci3,
  linetype = 3,
  linewidth = 1.5,
  col = "darkblue"
) +
scale_x_continuous(limits = c(0, 36), breaks = seq(0, 36, 6)) +
labs(
  title = "PACF Residuals SARIMA(2,1,1)(1,1,1)",
  x = "Lag",
  y = "ACF"
)
```



```r
res_acf3$acf |>
  as_tibble() |>
  ggplot(aes(x = V1)) +
    geom_histogram() +
    labs(
      title = "Residual Denisty",
      x = "Residuals",
      y = "Count"
    )
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



We can see from the ACF and PACF that the residuals do not have any correlation and appear to be random. We can see from the histogram that the residuals are approximately normal.

So far nothing disqualifies any of our models. We will need to rely on the AICc test to really determine which of the models is performing the best.

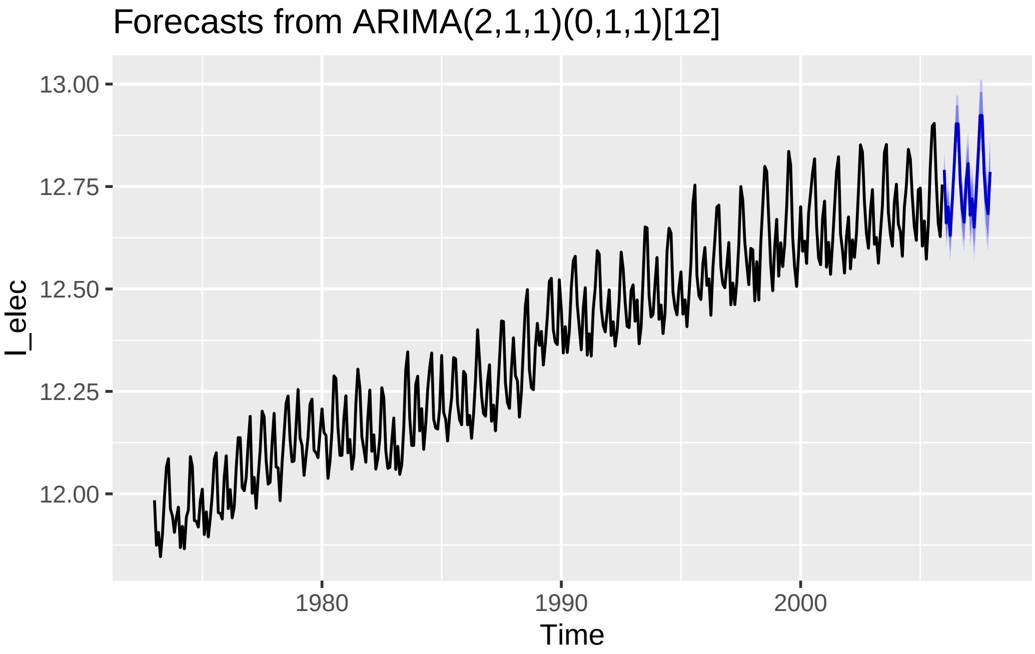
Fit AIC Results

model	AIC
SARIMA(2,0,1)(0,1,1)	-1691.460
SARIMA(2,1,1)(0,1,1)	-1682.843
SARIMA(2,1,1)(1,1,1)	-1682.843

From the AICc values we will choose to go with the SARIMA(2,1,1)(0,1,1) model as it has the least number of parameters and therefor is the model with the lowest complexity. There was nothing to suggest that it shouldn't be considered from the study of the residuals, as they look to be random and normally distributed.

g) product forecasts for this eries with a lead time of two years. Be sure to include forecast limits.

```r
# forecast with the forecast function
autoplot(forecast(fit_sarima211_011, h = 24))
```

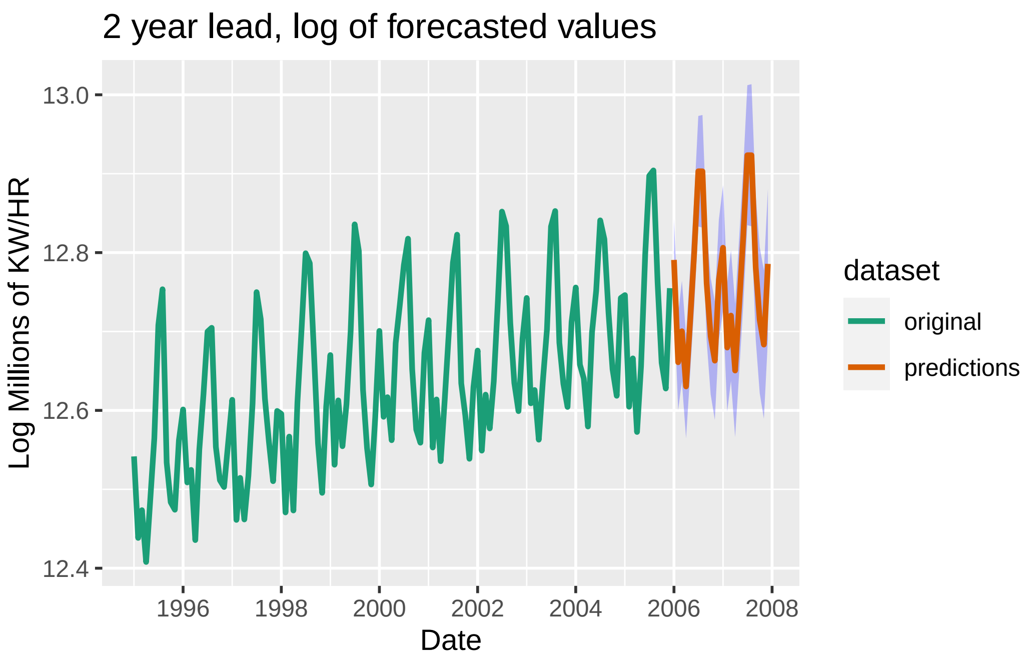


```r
# forecast with predict
pred <- predict(fit_sarima211_011, n.ahead = 24)
```

```r
pred_start_date <- as.Date("2006-01-01")
pred_end_date <- as.Date("2007-12-31")
```

```r
# plot forecast with predict

pred_plot <- data.frame("values" = as.matrix(pred$pred)) |>
          mutate(
            dataset = "predictions",
            date = seq(
              pred_start_date,
              pred_end_date,
              by = "month"
            )
          )

original_data <- l_elec |>
                  as_tibble() |>
                  mutate(
                    dataset = "original",
                    date = seq(
                      start_date,
                      end_date,
                      by = "month"
                    )
                  ) |>
                  rename("values" = "electricity")

plot_data <- rbind(original_data, pred_plot) |>
              filter(date >= "1995-01-01")

prediction_error <- data.frame(
  "se" = as.matrix(pred$se),
  "date" = seq(
    pred_start_date,
    pred_end_date,
    by = "month"
  )
) |>
  mutate(
    error_max = 2 * se + pred_plot$values,
    error_min = -2 * se + pred_plot$values
  )
```

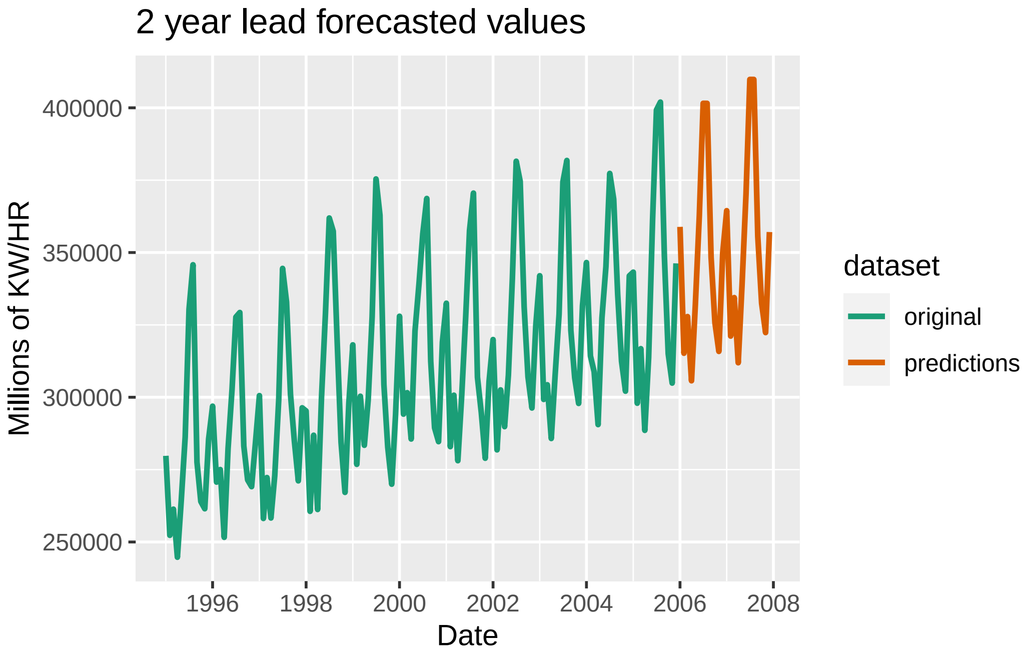
```
ggplot(plot_data) +
  geom_ribbon(
    data = prediction_error,
    aes(
      x = date,
      ymin = error_min,
      ymax = error_max,
    ),
    fill = "blue",
    alpha = .25
  ) +
  geom_line(
    aes(
      x = date,
      y = values,
      col = dataset
    ),
    linewidth = 1
  ) +
  scale_color_brewer(
    type = "qual",
    palette = 2
  ) +
  scale_x_date(date_breaks = "2 years", date_labels = "%Y") +
  labs(
    title = "2 year lead, log of forecasted values",
    x = "Date",
    y = "Log Millions of KW/HR"
  )
```



```
# undo the log transformation

pred_plot <- data.frame("values" = as.matrix(exp(pred$pred))) |>
            mutate(
              dataset = "predictions",
              date = seq(
                pred_start_date,
                pred_end_date,
                by = "month"
              )
            )

original_data <- l_elec |>
                    as_tibble() |>
                    mutate(
                      dataset = "original",
                      date = seq(
                        start_date,
                        end_date,
                        by = "month"
                      ),
                      electricity = exp(electricity)
```

```
                  ) |>
                  rename("values" = "electricity")

plot_data <- rbind(original_data, pred_plot) |>
              filter(date >= "1995-01-01")

prediction_error <- data.frame(
  "se" = as.matrix(pred$se),
  "date" = seq(
    pred_start_date,
    pred_end_date,
    by = "month"
  )
) |>
  mutate(
    error_max = exp(2 * se) + pred_plot$values,
    error_min = exp(-2 * se) + pred_plot$values
  )
```

```
ggplot(plot_data) +
  geom_ribbon(
    data = prediction_error,
    aes(
      x = date,
      ymin = error_min,
      ymax = error_max,
    ),
    fill = "blue",
    alpha = .25
  ) +
  geom_line(
    aes(
      x = date,
      y = values,
      col = dataset
    ),
    linewidth = 1
  ) +
  scale_color_brewer(
    type = "qual",
    palette = 2
  ) +
  scale_x_date(date_breaks = "2 years", date_labels = "%Y") +
  labs(
    title = "2 year lead forecasted values",
    x = "Date",
    y = "Millions of KW/HR"
  )
```


2 year lead forecasted values
Millions of KW/HR
Date
dataset
original
predictions