ARIMA and SARIMA examples

Loading relevant packages

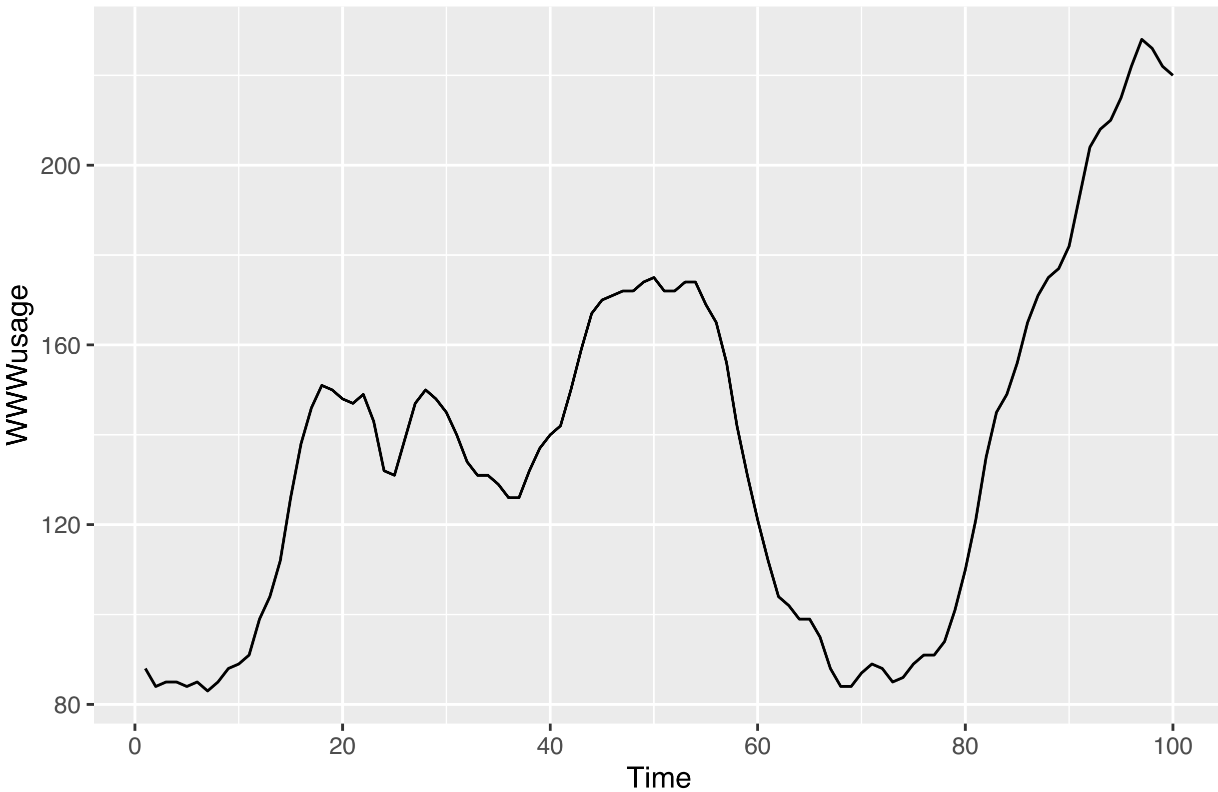
```r
library(tidyverse)
library(tidyquant)
library(gridExtra)
library(tibbletime)
library(forecast)
library(itsmr)
library(here)
library(fpp2)
library(tseries)
knitr::opts_chunk$set(comment=NA,tidy=FALSE)

#library(future) Not needed yet
#library(doFuture) Not needed yet
#library(rbenchmark) Not needed yet
```

Quick note

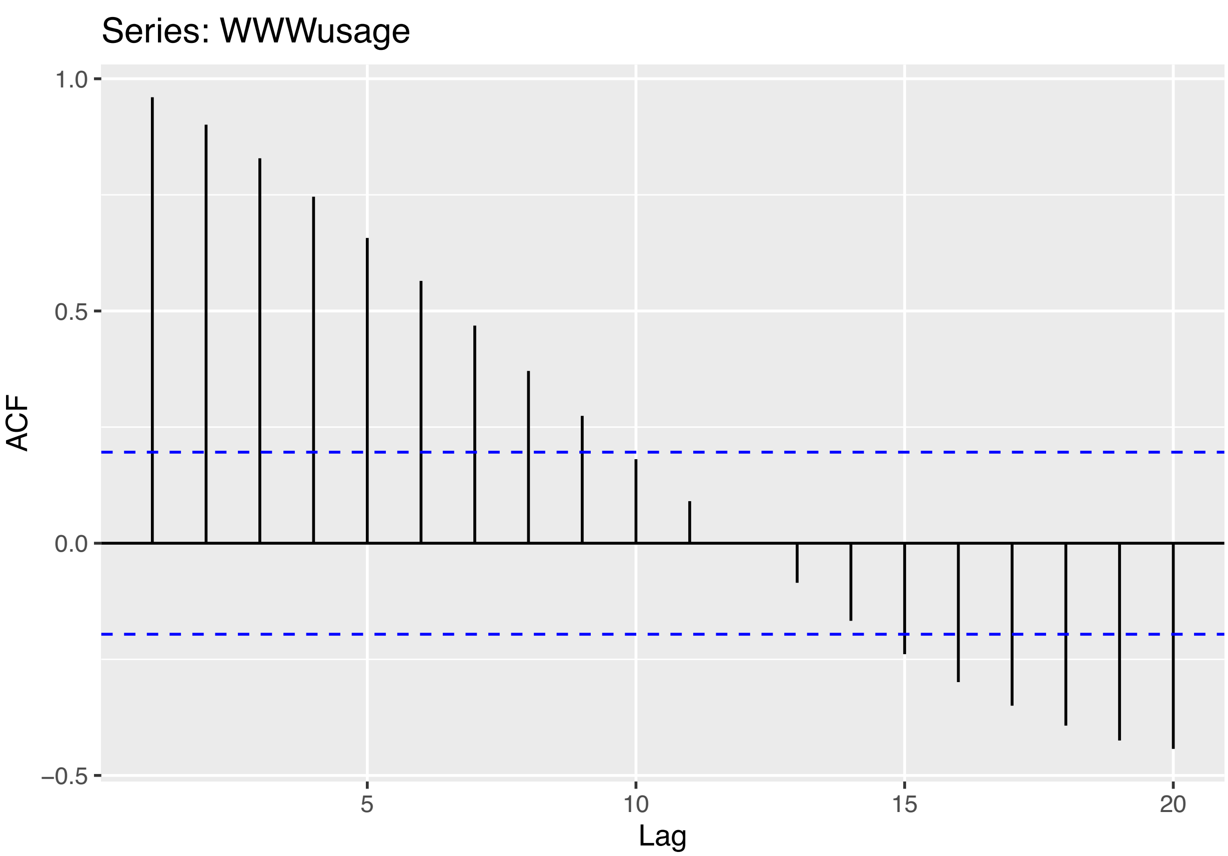
Note that the *auto.arima* function uses approximations by default when comparing models during its tests runs. I've set approximation to FALSE here in the code, but it will take longer to run in general. If your computer is slow, set all of those to TRUE.

WWWusage data

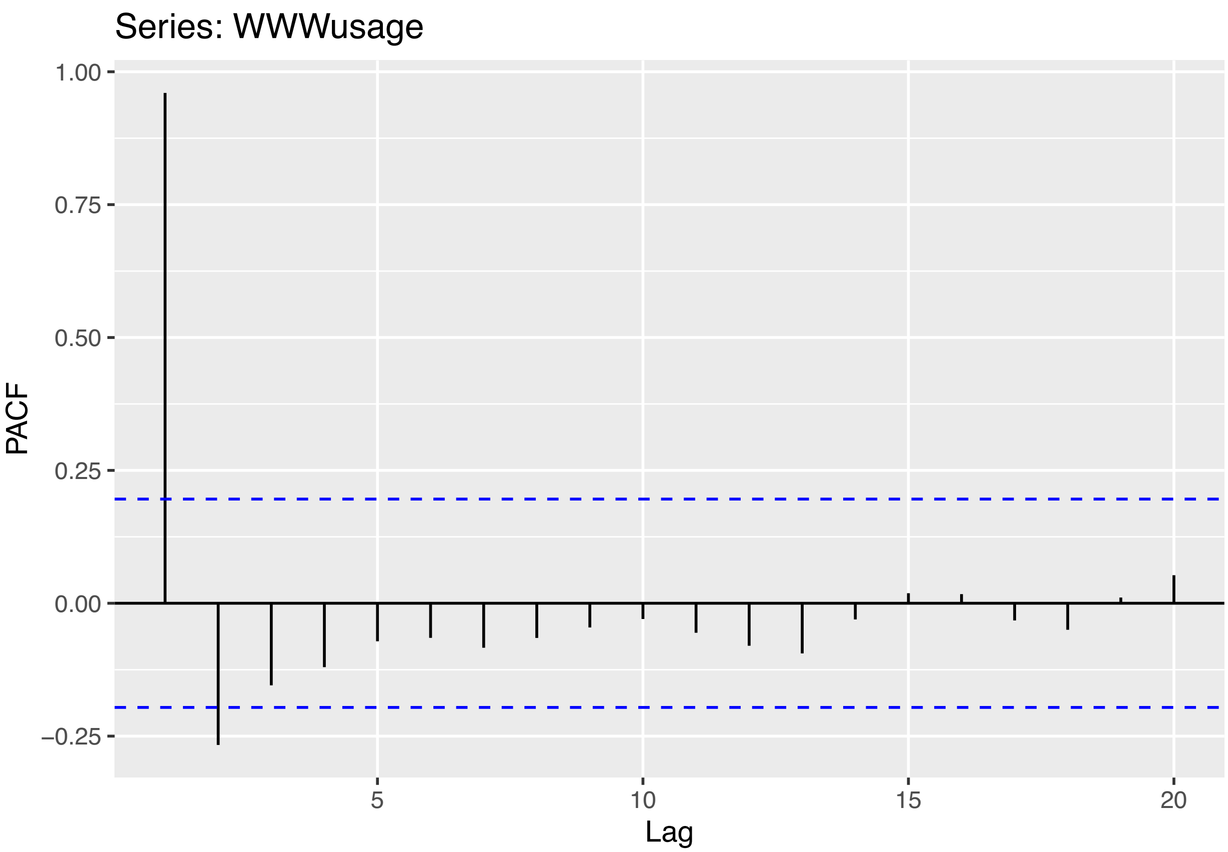
```r
autoplot(WWWusage)
```




```r
ggAcf(WWWusage)
```



```r
ggPacf(WWWusage)
```




```r
adf.test(WWWusage)
```

cannot reject
non stationarity

```
    Augmented Dickey-Fuller Test

data:  WWWusage
Dickey-Fuller = -2.6421, Lag order = 4, p-value = 0.3107
alternative hypothesis: stationary
```

```r
### Another test used for testing for stationarity
### Unilke the adf.test, this test has stationarity as null and rejects for non-stationarity
kpss.test(WWWusage)
```

modest evidence
against stationarity

```
    KPSS Test for Level Stationarity

data:  WWWusage
KPSS Level = 0.45424, Truncation lag parameter = 4, p-value = 0.05377
```

Note that the two tests agree that the data are likely to be non-stationary.

```r
WWWusage_diff=diff(WWWusage,1)
adf.test(WWWusage_diff)
```

```
    Augmented Dickey-Fuller Test
```

```
data:  WWWusage_diff
Dickey-Fuller = -2.5459, Lag order = 4, p-value = 0.3506
alternative hypothesis: stationary
```

```r
kpss.test(WWWusage_diff)
```

```
Warning in kpss.test(WWWusage_diff): p-value greater than printed p-value

	KPSS Test for Level Stationarity

data:  WWWusage_diff
KPSS Level = 0.2175, Truncation lag parameter = 3, p-value = 0.1
```

stationary

Note there that the two tests **disagree** about whether the 1st order differences are stationary. The ADF test still does not reject the non-stationary hypothesis, so cannot conclude that it is stationary. The KPSS test fails to reject a null stationary hypothesis, so cannot conclude that it is non-stationary.

Note that the non-augmented Dickey-Fuller test does rejects the hypothesis of a unit root:

```r
adf.test(WWWusage_diff,k=1)
```

```
Warning in adf.test(WWWusage_diff, k = 1): p-value smaller than printed p-value

	Augmented Dickey-Fuller Test

data:  WWWusage_diff
Dickey-Fuller = -4.1889, Lag order = 1, p-value = 0.01
alternative hypothesis: stationary
```

```r
adf.test(WWWusage_diff,k=2)
```

```
	Augmented Dickey-Fuller Test

data:  WWWusage_diff
Dickey-Fuller = -2.6766, Lag order = 2, p-value = 0.2965
alternative hypothesis: stationary
```

```r
adf.test(WWWusage_diff,k=3)
```

```
	Augmented Dickey-Fuller Test

data:  WWWusage_diff
Dickey-Fuller = -2.6106, Lag order = 3, p-value = 0.3238
alternative hypothesis: stationary
```

```r
adf.test(WWWusage_diff,k=4)
```

```
	Augmented Dickey-Fuller Test

data:  WWWusage_diff
Dickey-Fuller = -2.5459, Lag order = 4, p-value = 0.3506
alternative hypothesis: stationary
```

```r
adf.test(WWWusage_diff,k=5)
```

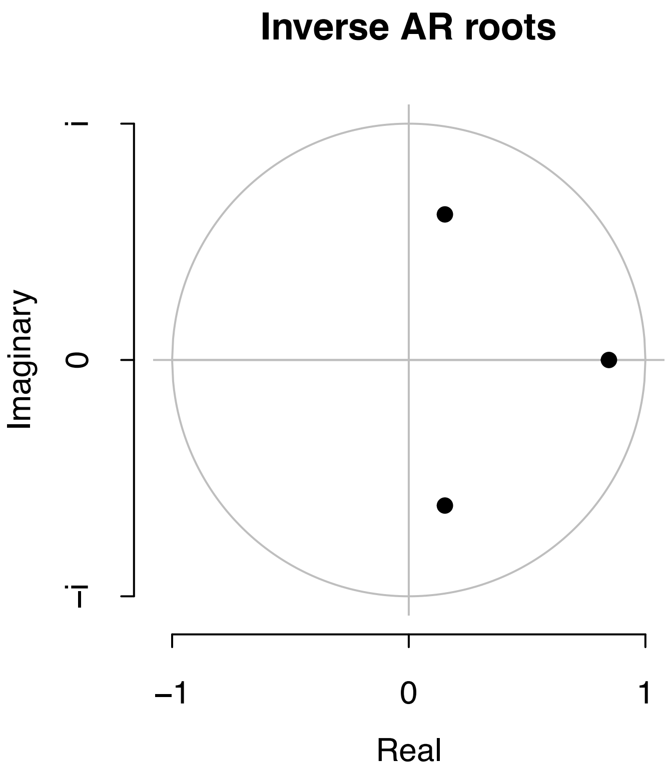
```
    Augmented Dickey-Fuller Test

data:  WWWusage_diff
Dickey-Fuller = -2.6744, Lag order = 5, p-value = 0.2974
alternative hypothesis: stationary
```

It could be the ADF test in this case is simply underpowered to reject against the hypothesis of nonstationarity… but similarly the KPSS test could be underpowered to reject the nonstationarity assumption as well. Therefore, we should go ahead and fit some models keeping both of these things in mind.

```r
WWW_arima = auto.arima(WWWusage,seasonal=FALSE,stepwise=FALSE,approximation=FALSE)
summary(WWW_arima)
```

```
Series: WWWusage
ARIMA(3,1,0)

Coefficients:
         ar1      ar2     ar3
      1.1513  -0.6612  0.3407
s.e.  0.0950   0.1353  0.0941

sigma^2 estimated as 9.656:  log likelihood=-252
AIC=511.99   AICc=512.42   BIC=522.37

Training set error measures:
                   ME     RMSE      MAE       MPE     MAPE      MASE
Training set 0.230588 3.044632 2.367157 0.2748377 1.890528 0.5230995
                     ACF1
Training set -0.003095066
```

```r
plot(WWW_arima)
```



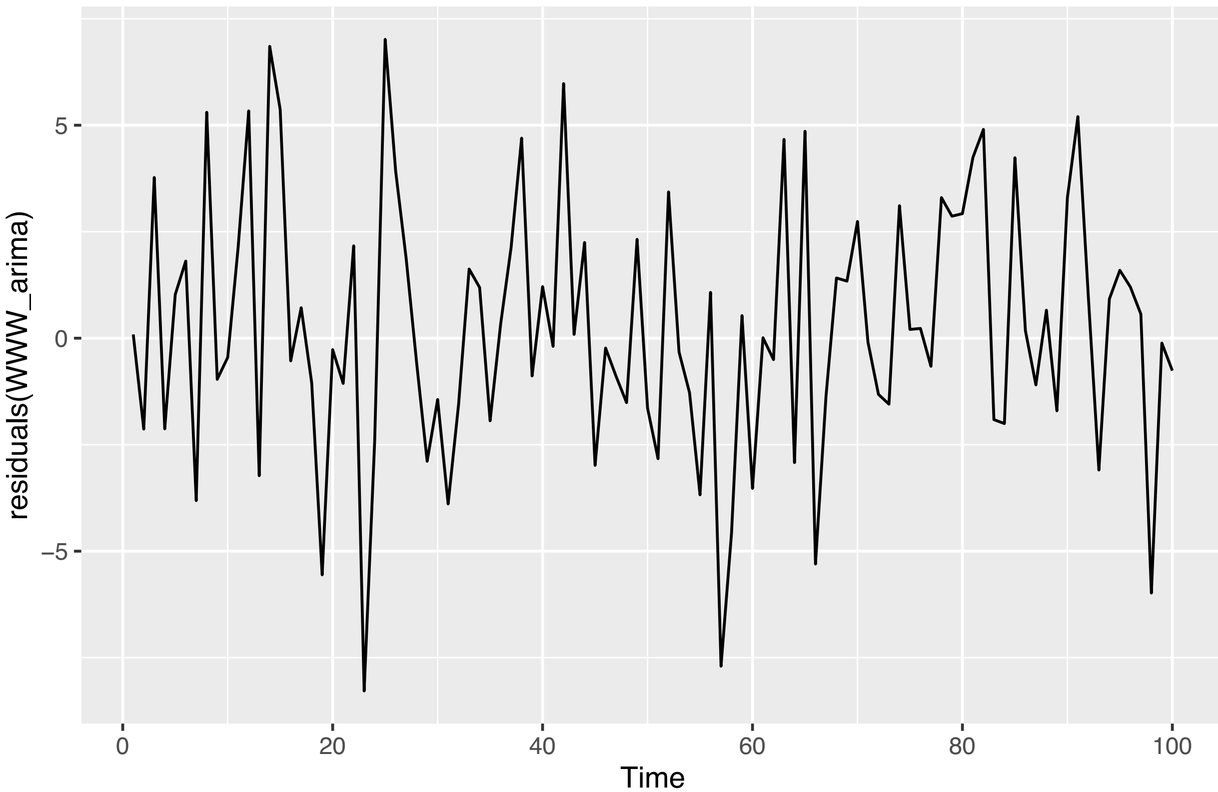

```
adf.test(residuals(WWW_arima))
```

```
	Augmented Dickey-Fuller Test

data:  residuals(WWW_arima)
Dickey-Fuller = -4.0315, Lag order = 4, p-value = 0.01052
alternative hypothesis: stationary
```

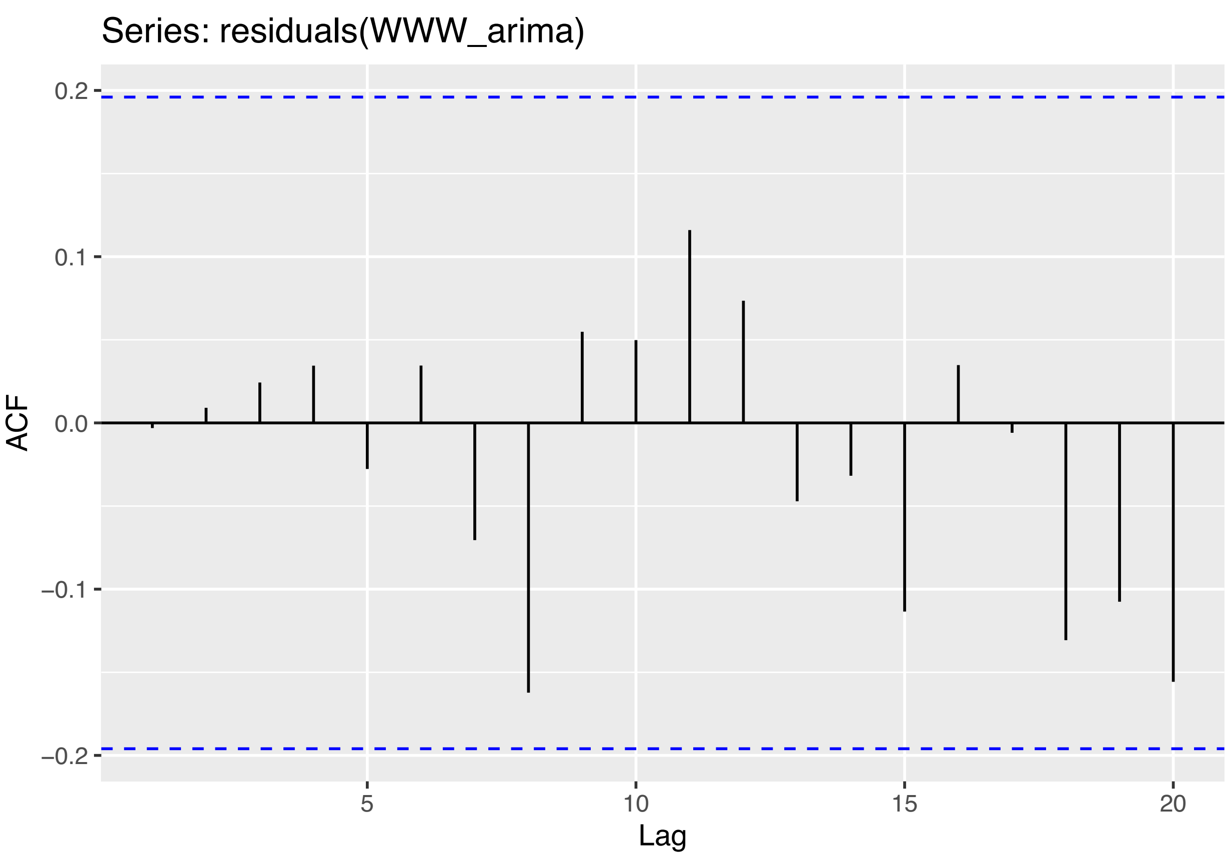
good

reject nonstationary

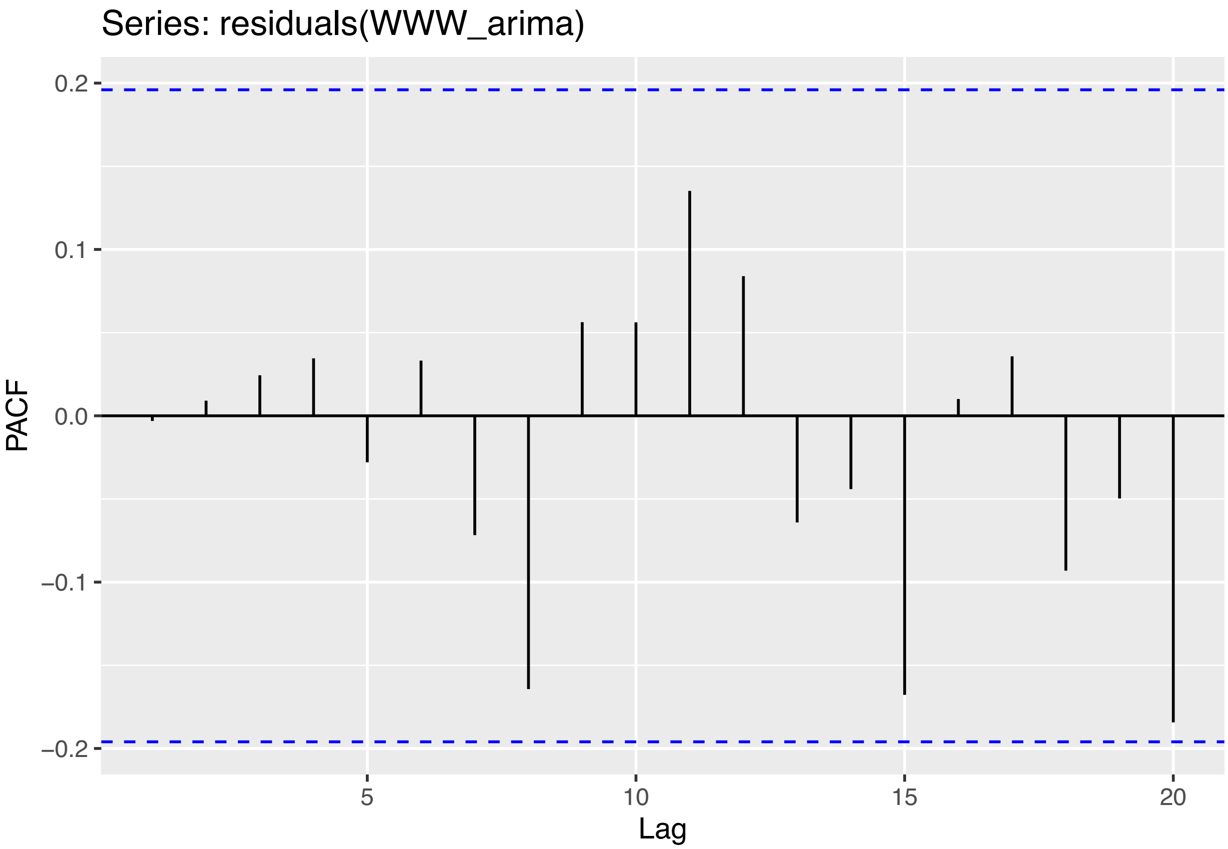
```
autoplot(residuals(WWW_arima))
```




```
ggAcf(residuals(WWW_arima))
```



```
ggPacf(residuals(WWW_arima))
```



From the *auto-arima* function, we see that AIC chooses an ARIMA(3,1,0) model which indicates that we should use a first-order difference ($d = 1$) with $p = 3$ and $q = 0$. The residuals appear to be indistinguishable from white noise and the differenced series does not appear to be nonstationary visually. Note that **auto.arima(.)** uses the KPSS test to decide on the differencing (i.e. it increases d until the KPSS test fails to reject the stationary hypothesis).

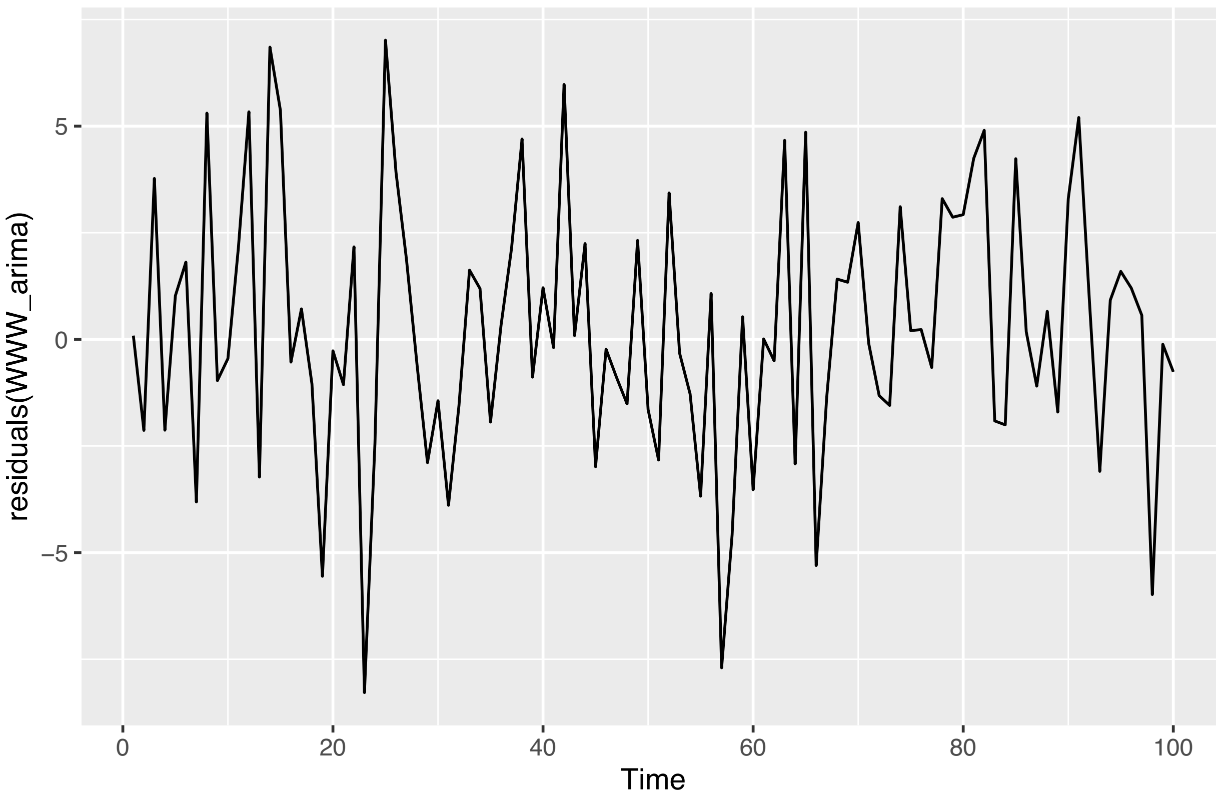
We could see if additional differencing helps by forcing $d = 2$:

```
WWW_arima_diff2 = auto.arima(WWWusage,d=2,seasonal=FALSE,stepwise=FALSE,approximation=FALSE)
summary(WWW_arima_diff2)
```

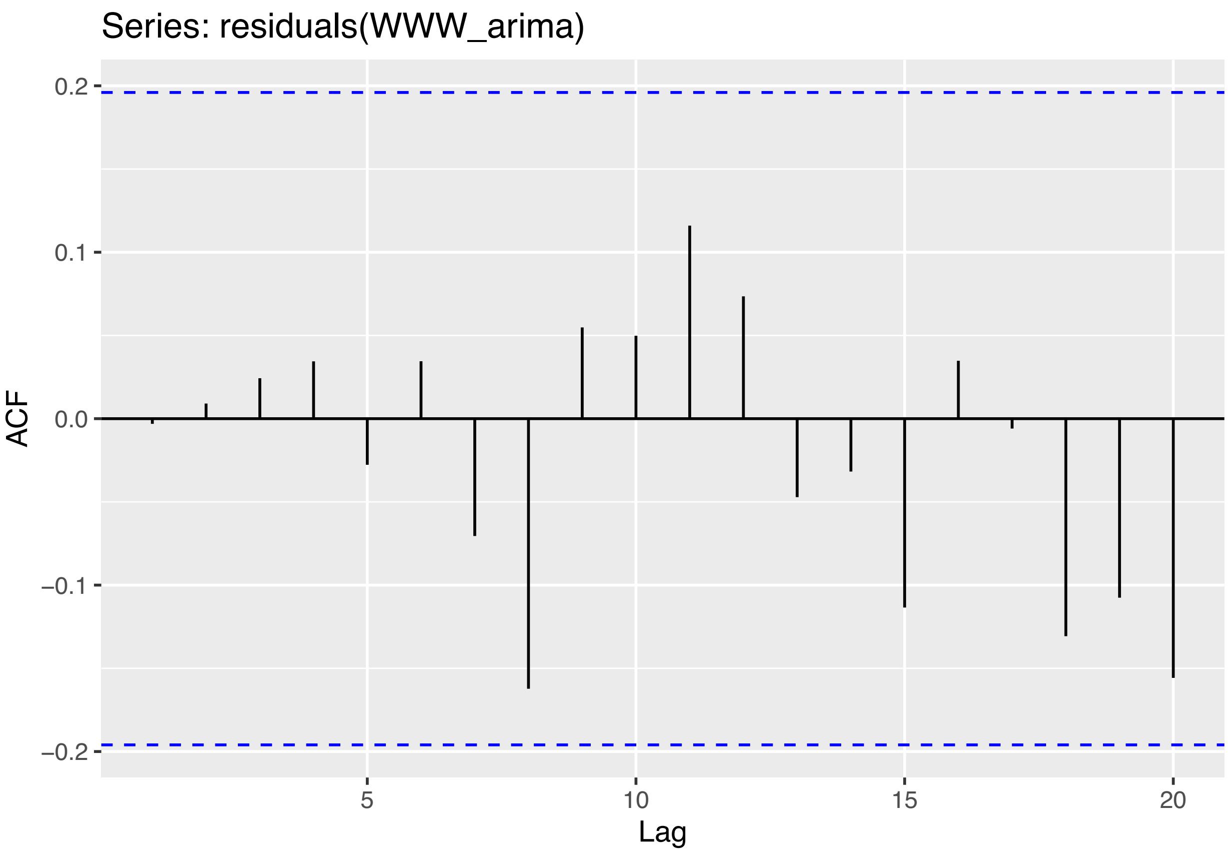
```
Series: WWWusage
ARIMA(2,2,0)

Coefficients:
         ar1      ar2
      0.2579  -0.4407
s.e.  0.0915   0.0906

sigma^2 estimated as 10.34:  log likelihood=-252.73
AIC=511.46   AICc=511.72   BIC=519.22

Training set error measures:
                     ME     RMSE      MAE      MPE     MAPE      MASE
Training set 0.02797758 3.150308 2.511921 0.206235 1.994727 0.5550897
                   ACF1
```

```
Training set -0.0235521
```

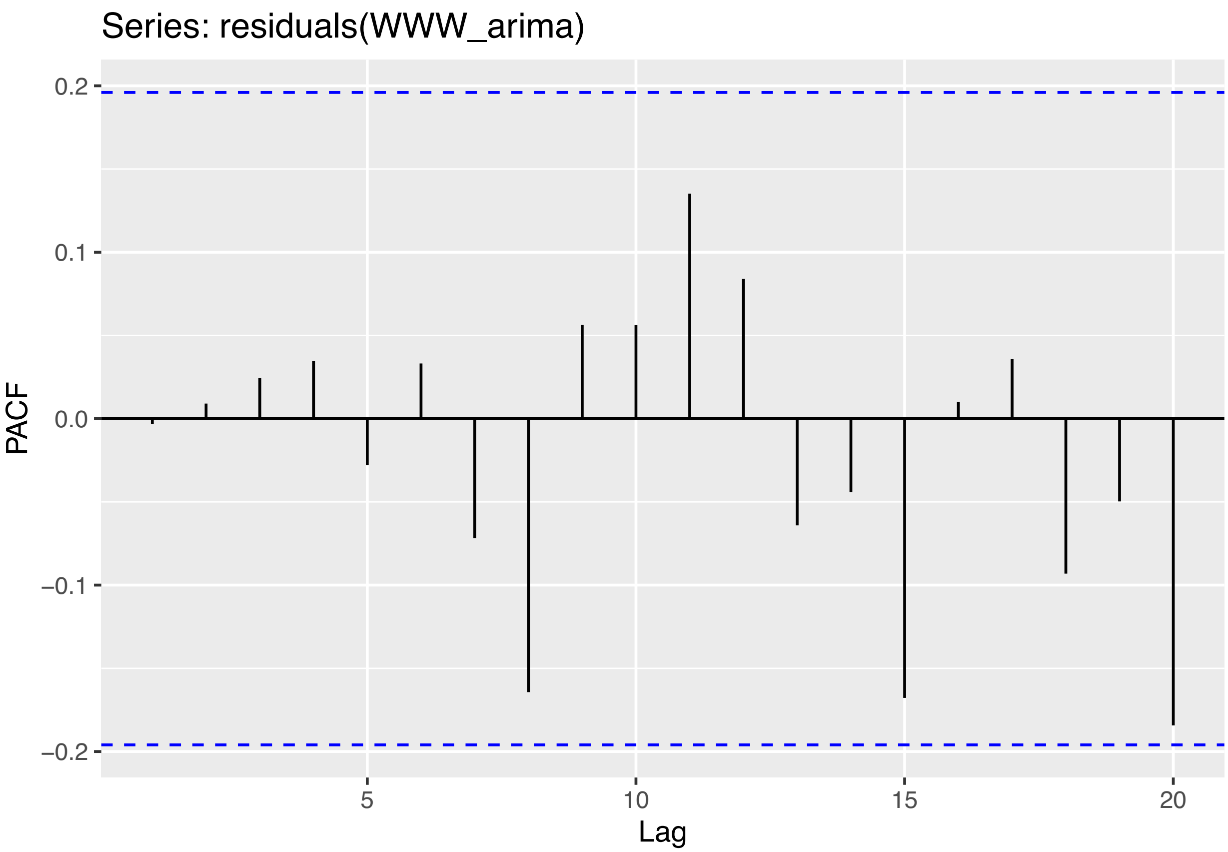
```r
autoplot(residuals(WWW_arima))
```




```r
ggAcf(residuals(WWW_arima))
```




```r
ggPacf(residuals(WWW_arima))
```

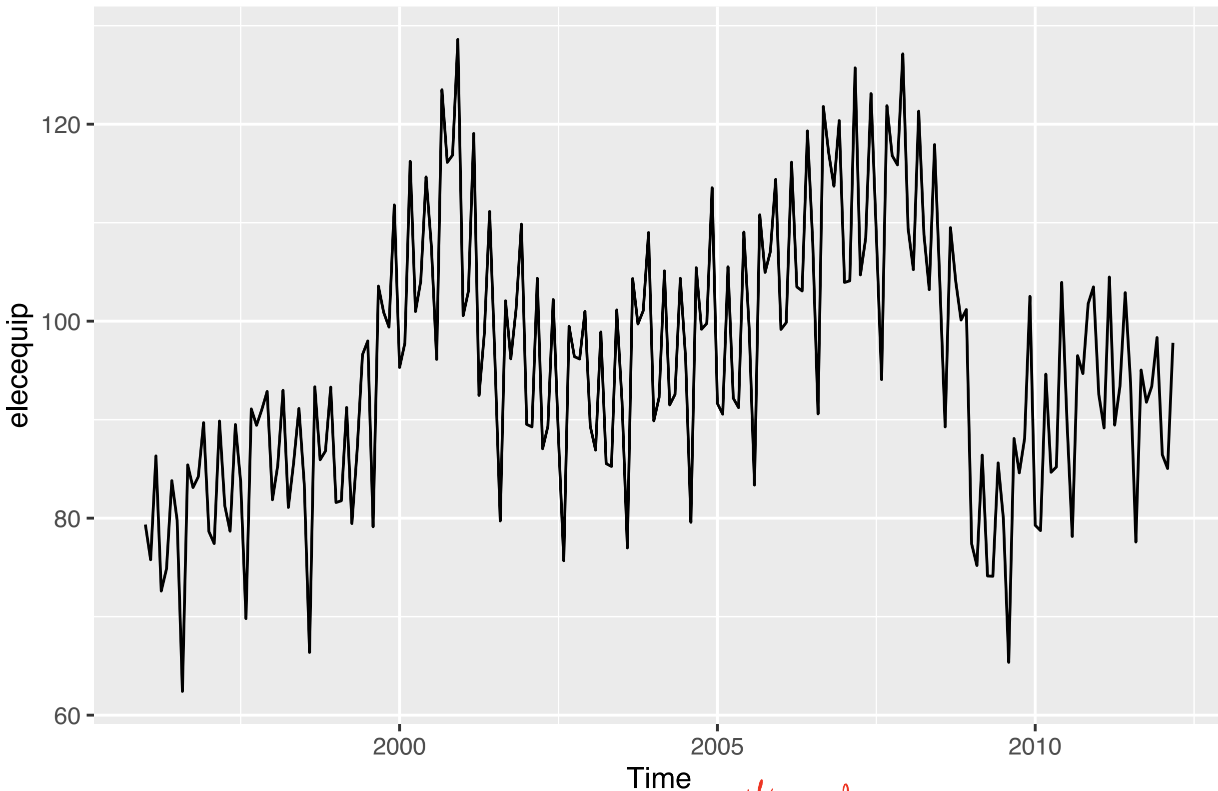


Note that the error is slightly worse (higher σ^2 estimate) and that there is no improvement in the residual plots. We would likely conclude that the ARIMA(3,1,0) is the preferred model for this data.

Electric equipment data

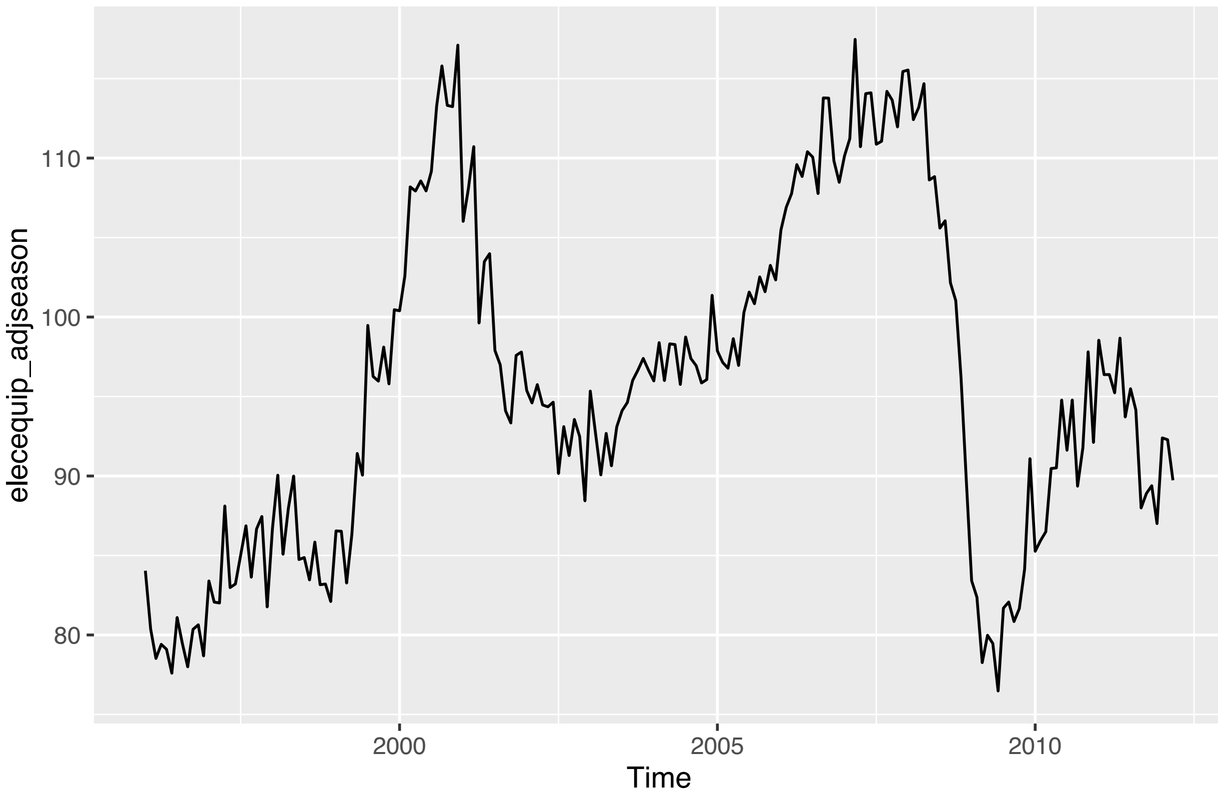
First we try to remove the seasonal component by hand.

```
autoplot(elecequip)
```

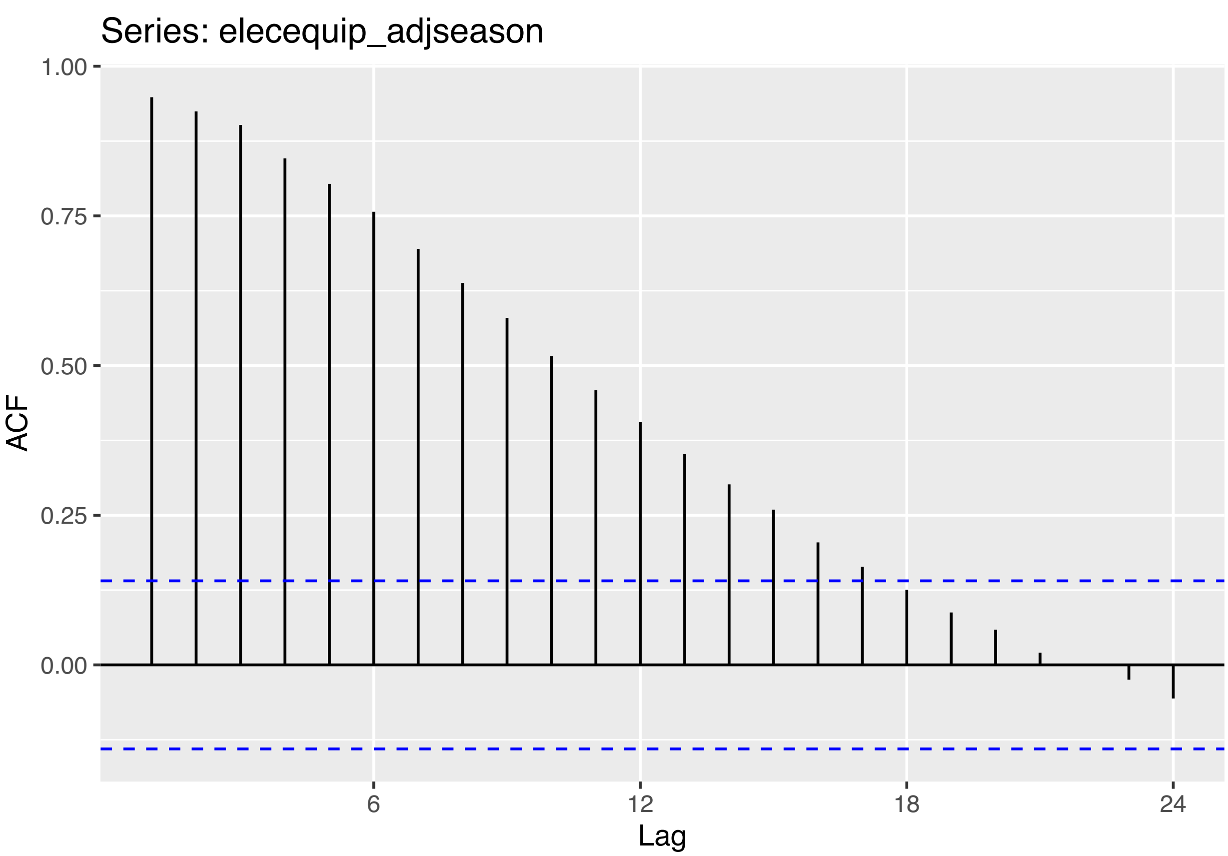



```
elecequip_decomp = stl(elecequip,s.window=12)

elecequip_adjseason = elecequip - seasonal(elecequip_decomp)

autoplot(elecequip_adjseason)
```

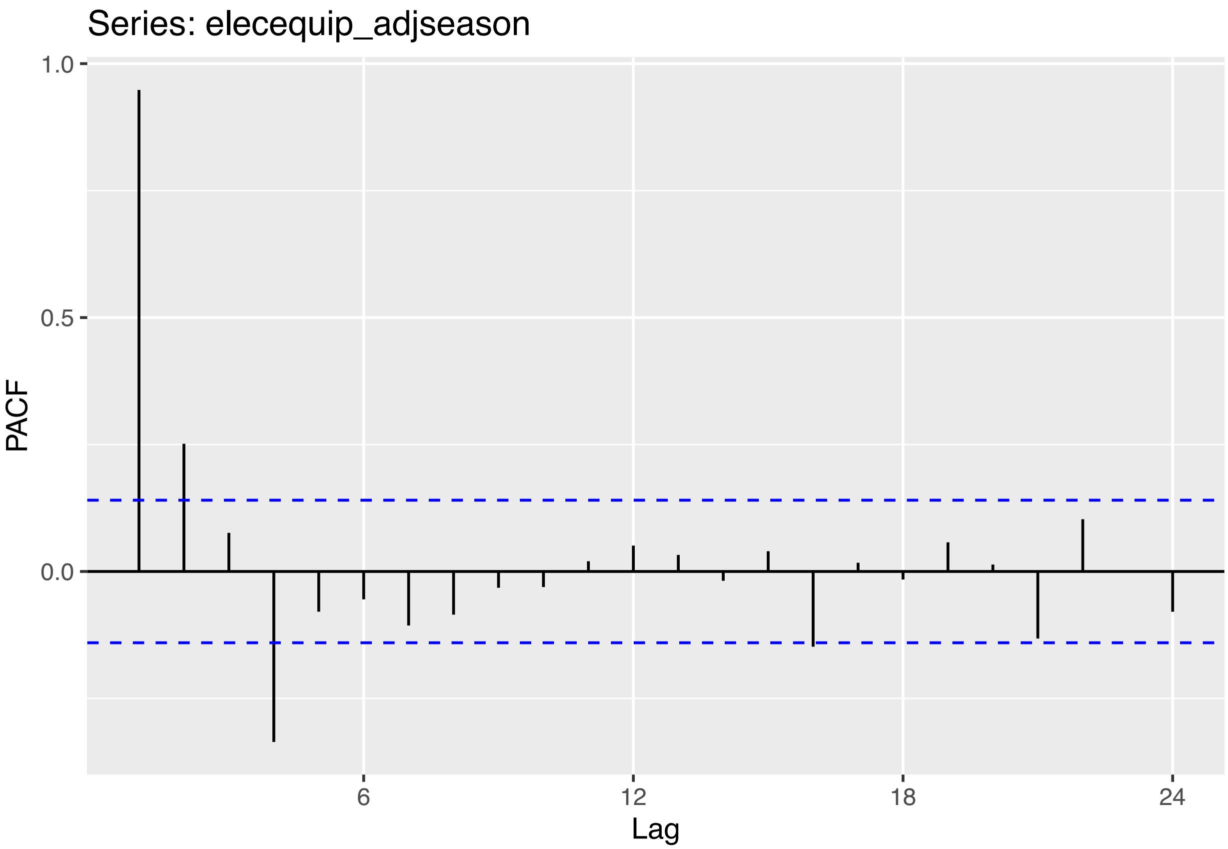
remove seasonal




```r
ggAcf(elecequip_adjseason)
```



```r
ggPacf(elecequip_adjseason)
```




```r
adf.test(elecequip_adjseason)
```

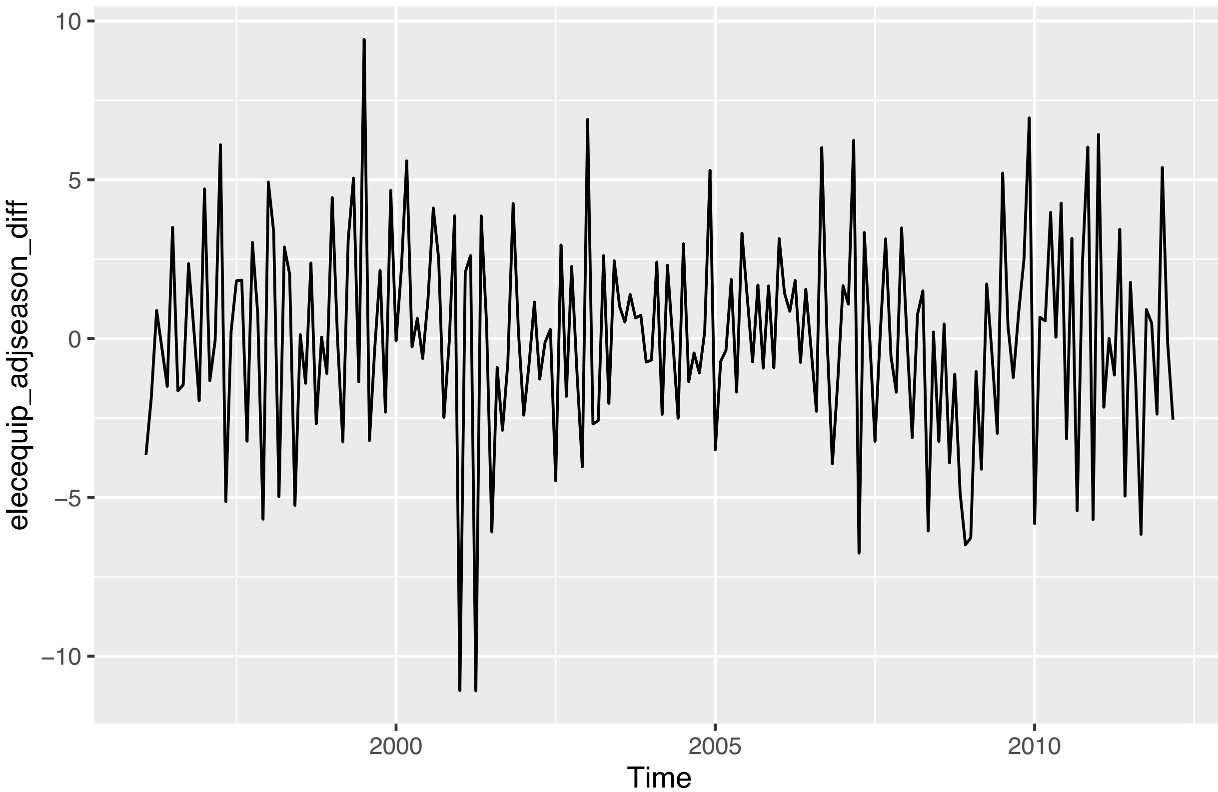
```
    Augmented Dickey-Fuller Test

data:  elecequip_adjseason
Dickey-Fuller = -2.4971, Lag order = 5, p-value = 0.368
alternative hypothesis: stationary
```

```r
kpss.test(elecequip_adjseason)
```

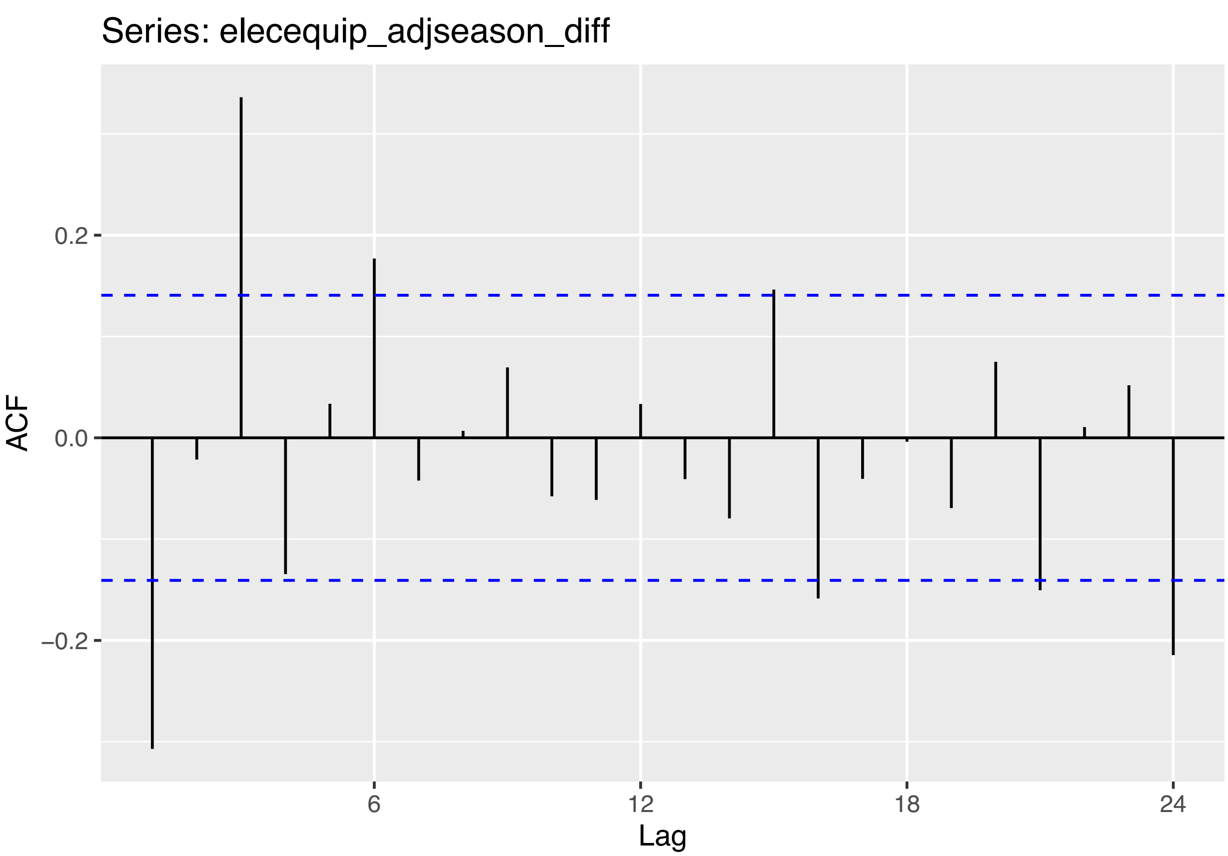
```
    KPSS Test for Level Stationarity

data:  elecequip_adjseason
KPSS Level = 0.7053, Truncation lag parameter = 4, p-value = 0.01306
```

The data still appear to be non-stationary, even after removing the seasonal component, according to both tests. We can then try differencing:

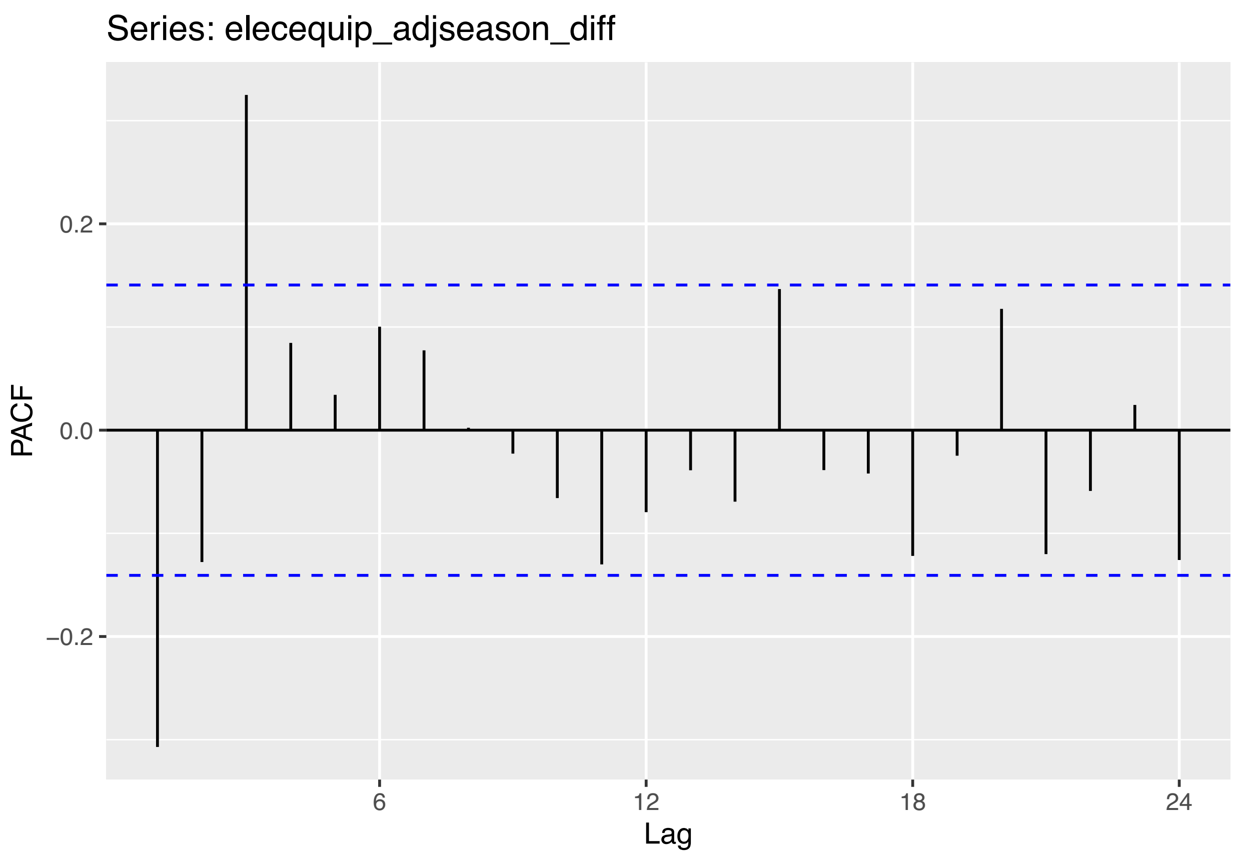
```r
elecequip_adjseason_diff = diff(elecequip_adjseason,1)

autoplot(elecequip_adjseason_diff)
```




```
ggAcf(elecequip_adjseason_diff)
```



```r
ggPacf(elecequip_adjseason_diff)
```




```r
adf.test(elecequip_adjseason_diff)
```

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

	Augmented Dickey-Fuller Test

data:  elecequip_adjseason_diff
Dickey-Fuller = -4.0924, Lag order = 5, p-value = 0.01
alternative hypothesis: stationary
```

reject non-stationary

✓

```r
adf.test(elecequip_adjseason_diff,k=3)
```

```
Warning in adf.test(elecequip_adjseason_diff, k = 3): p-value smaller than
printed p-value

	Augmented Dickey-Fuller Test

data:  elecequip_adjseason_diff
Dickey-Fuller = -5.4491, Lag order = 3, p-value = 0.01
alternative hypothesis: stationary
```

```r
kpss.test(elecequip_adjseason_diff)
```

```
Warning in kpss.test(elecequip_adjseason_diff): p-value greater than printed p-
```

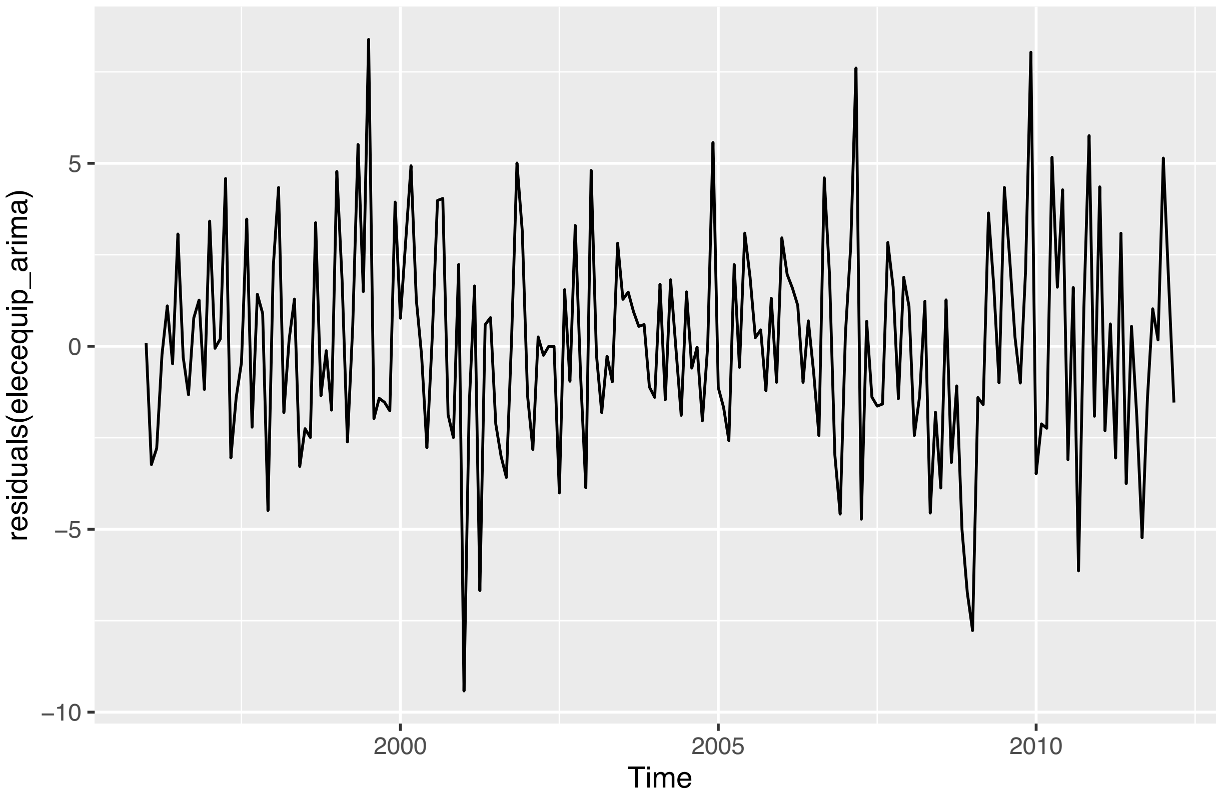
```
value


    KPSS Test for Level Stationarity

data:  elecequip_adjseason_diff
KPSS Level = 0.12712, Truncation lag parameter = 4, p-value = 0.1
```

We see here that the two tests again disagree at the default lags, but agree at smaller lags (which have more power).

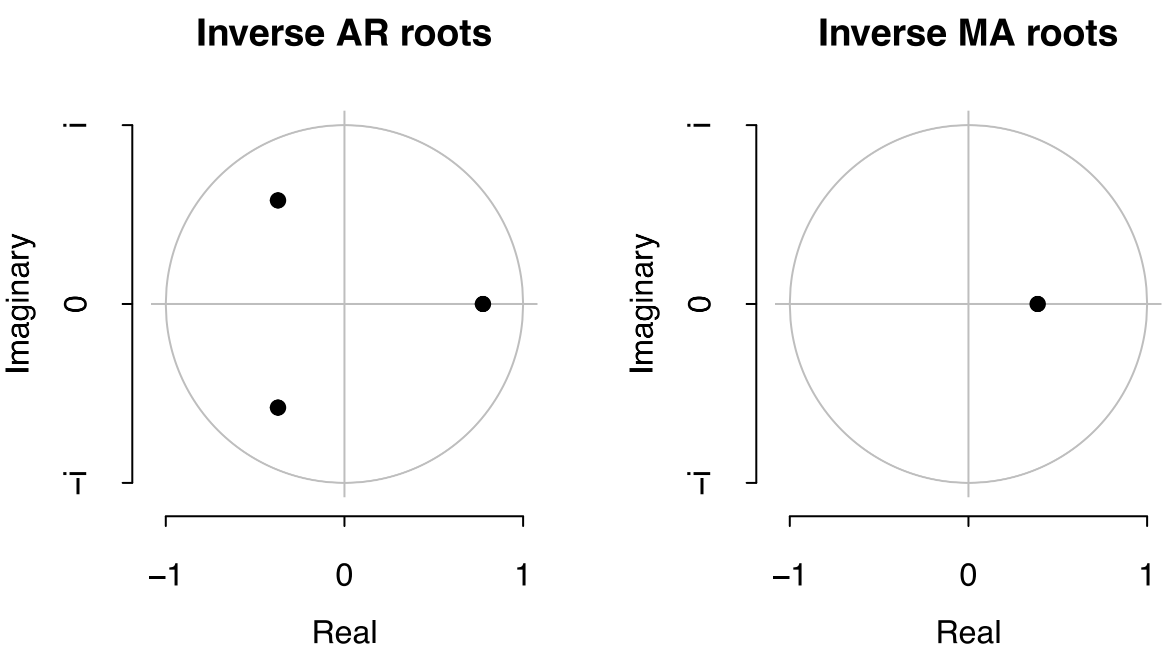
```r
elecequip_arima = auto.arima(elecequip_adjseason,stepwise=FALSE,approximation=FALSE,seasonal=FALSE,allo
summary(elecequip_arima)
```

```
Series: elecequip_adjseason
ARIMA(3,1,1)

Coefficients:
         ar1     ar2     ar3      ma1
      0.0313  0.1025  0.3674  -0.3875
s.e.  0.2041  0.0884  0.0670   0.2235

sigma^2 estimated as 8.439:  log likelihood=-480.41
AIC=970.82   AICc=971.14   BIC=987.16

Training set error measures:
                     ME     RMSE      MAE         MPE     MAPE      MASE
Training set 0.03589222 2.867495 2.240372 0.006428808 2.355094 0.2742413
                   ACF1
Training set 0.01018386
```

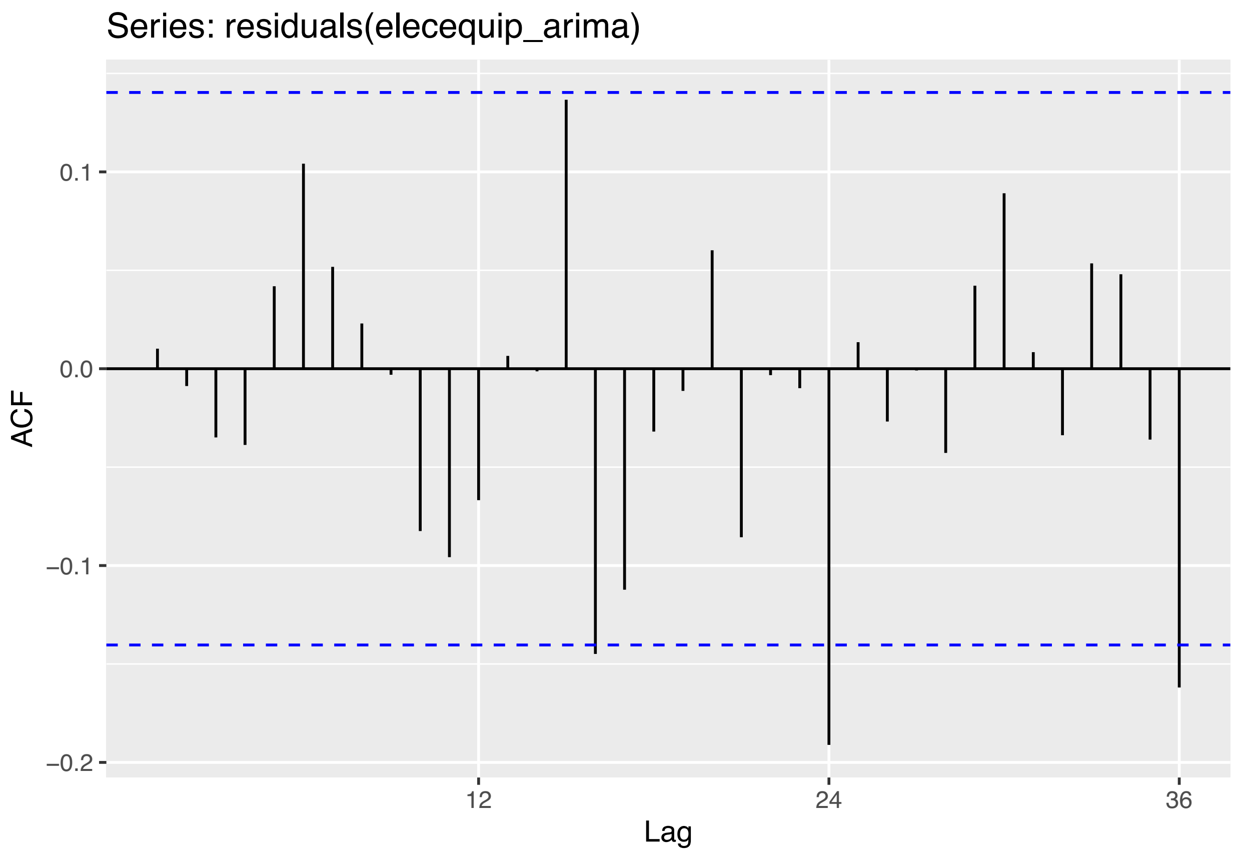
```r
autoplot(residuals(elecequip_arima))
```



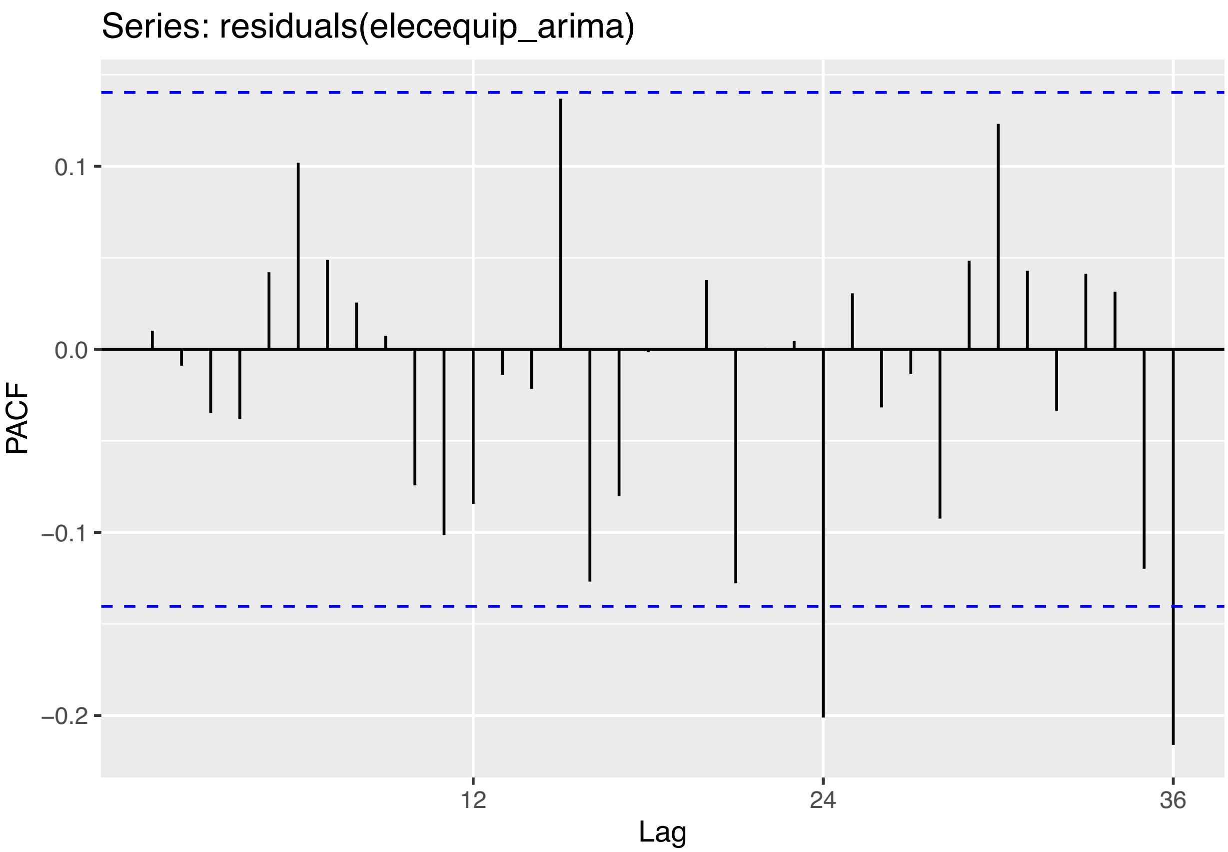

```r
plot(elecequip_arima)
```




```r
ggAcf(residuals(elecequip_arima),lag.max=36)
```




```
ggPacf(residuals(elecequip_arima),lag.max=36)
```




```
adf.test(residuals(elecequip_arima))
```

```
Warning in adf.test(residuals(elecequip_arima)): p-value smaller than printed p-
value

    Augmented Dickey-Fuller Test

data:  residuals(elecequip_arima)
Dickey-Fuller = -5.0567, Lag order = 5, p-value = 0.01
alternative hypothesis: stationary
```

```
kpss.test(residuals(elecequip_arima))
```

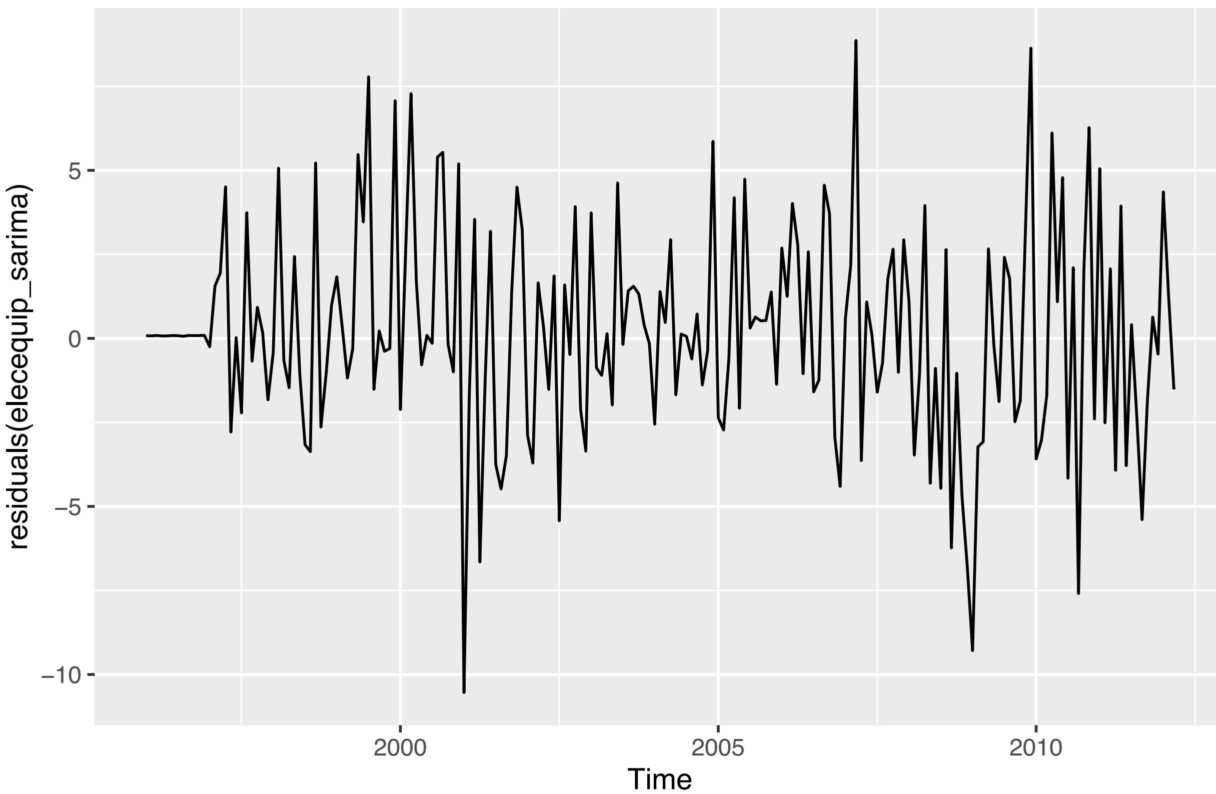
```
Warning in kpss.test(residuals(elecequip_arima)): p-value greater than printed
p-value

    KPSS Test for Level Stationarity

data:  residuals(elecequip_arima)
KPSS Level = 0.078283, Truncation lag parameter = 4, p-value = 0.1
```

We see here that the auto.arima functions selects an ARIMA(3,1,1), so that the first order differencing was sufficient. We do see a somewhat large correlation at 24 months and 36 months in the ACF and PCF. This suggests that we could also consider a SARIMA model on the original and that our original attempt to remove the seasonality was flawed in assuming that the seasonal contribution was the same across all periods.

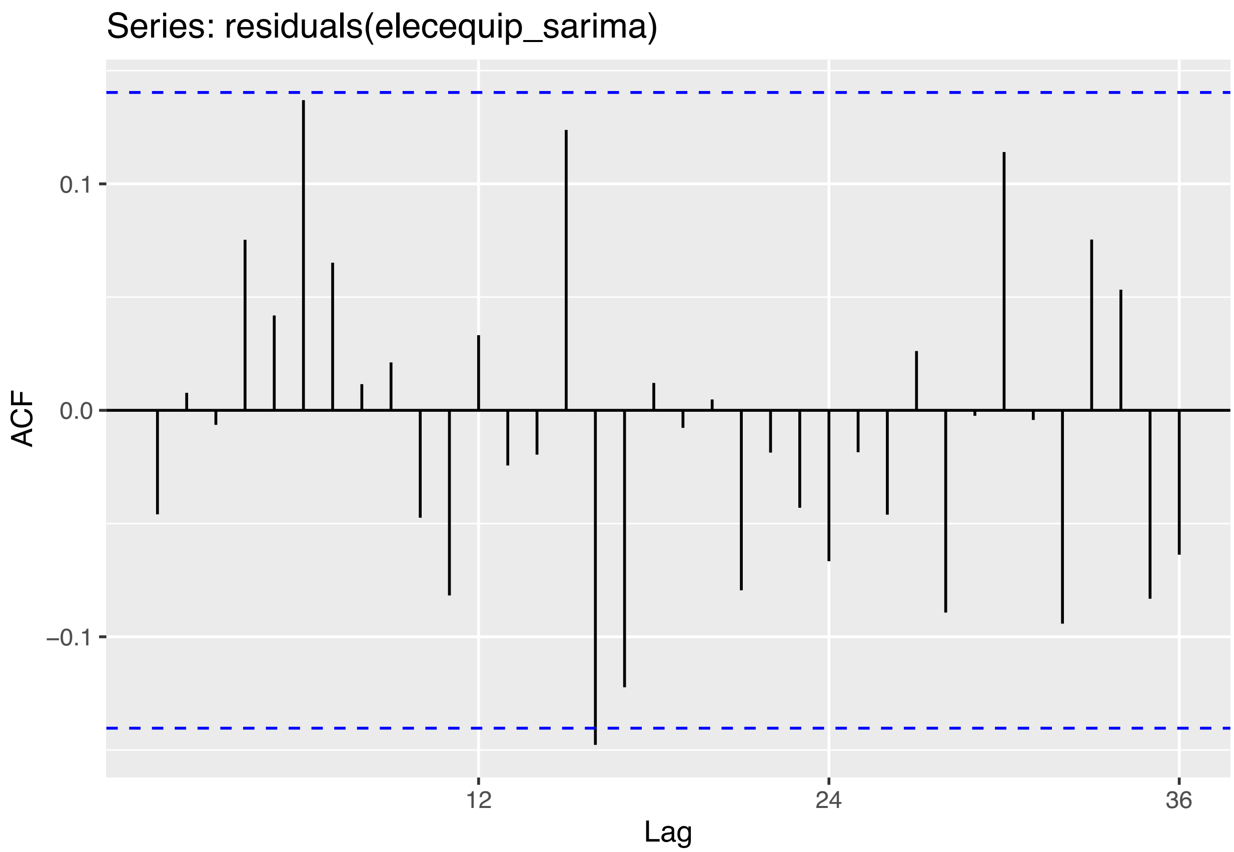
```
elecequip_sarima = auto.arima(elecequip,stepwise=FALSE,approximation=FALSE,seasonal=TRUE,allowdrift=FALS
summary(elecequip_sarima)
```

```
Series: elecequip
ARIMA(4,0,0)(0,1,1)[12]

Coefficients:
         ar1     ar2     ar3      ar4     sma1
      0.6556  0.2855  0.3516  -0.3351  -0.8307
s.e.  0.0694  0.0809  0.0818   0.0696   0.0751

sigma^2 estimated as 11.13:  log likelihood=-485.44
AIC=982.89   AICc=983.37   BIC=1002.15

Training set error measures:
                    ME     RMSE      MAE       MPE     MAPE      MASE
Training set 0.1742991 3.187073 2.411903 0.1026447 2.516161 0.2951376
                    ACF1
Training set -0.04592709
```

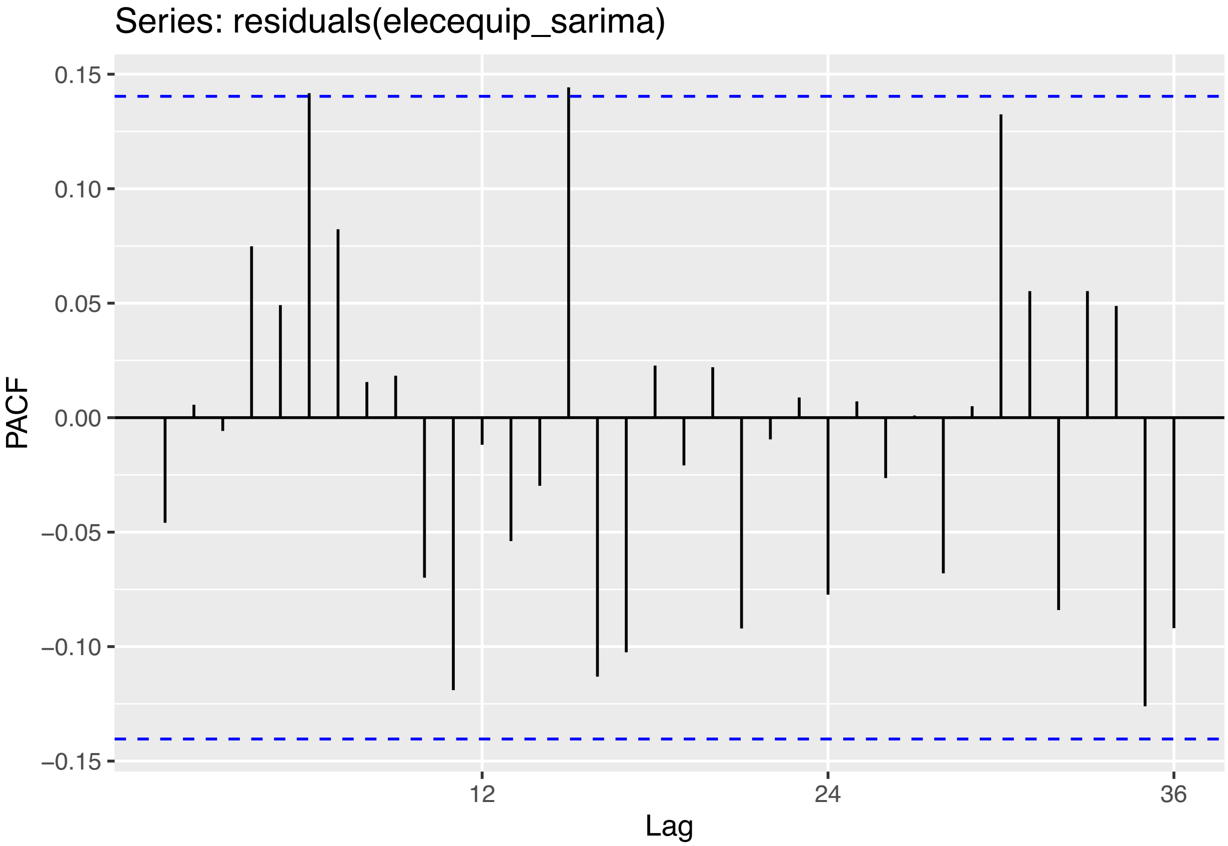
```
autoplot(residuals(elecequip_sarima))
```




```
ggAcf(residuals(elecequip_sarima),lag=36)
```




```
ggPacf(residuals(elecequip_sarima),lag=36)
```



Note that auto.arima selects a SARIMA(4,0,0)(0,1,1) model which indicates that we have a seasonal difference of order $D = 1$ with a seasonal MA component of the model (i.e. we should difference the values by *lag of 12* not by *lag 1*, there is a moving average component that changes the magnitude of the seasonal component that varies over time). Note that we cannot compare these models directly using AIC/AICc/BIC because the amounts of data used are different (because of the differencing by lag period rather than by lag). It seems from the residuals (and by the model selection process) that this would be potentially a better way to model the data.