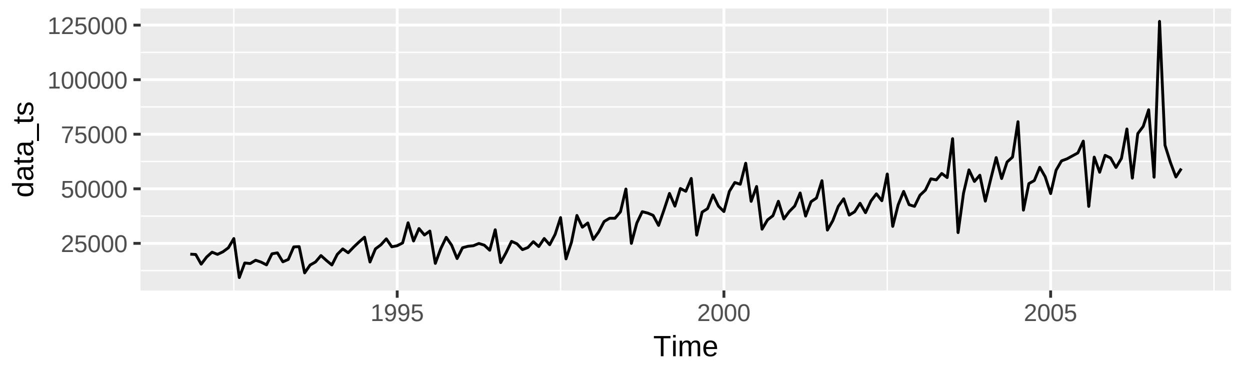
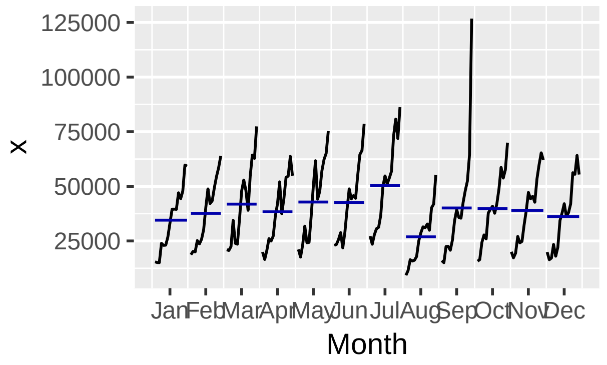
# Homework3

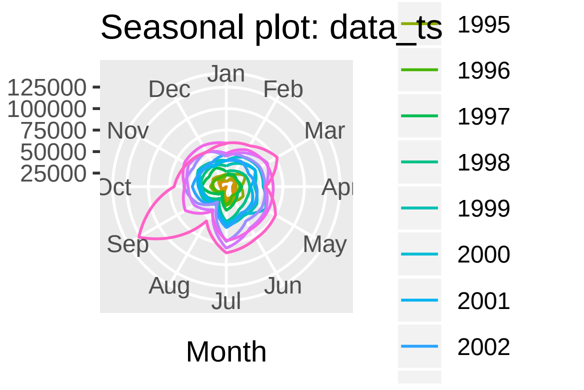
*Monika Wysoczanska, Manuel Barbas, Diogo Oliveira*

We start our analysis with plotting the data.



Observations:
1. There is a general increasing trend in the data.
2. Seasonality each year (around middle of a year)
3. Non-stationary behaviour

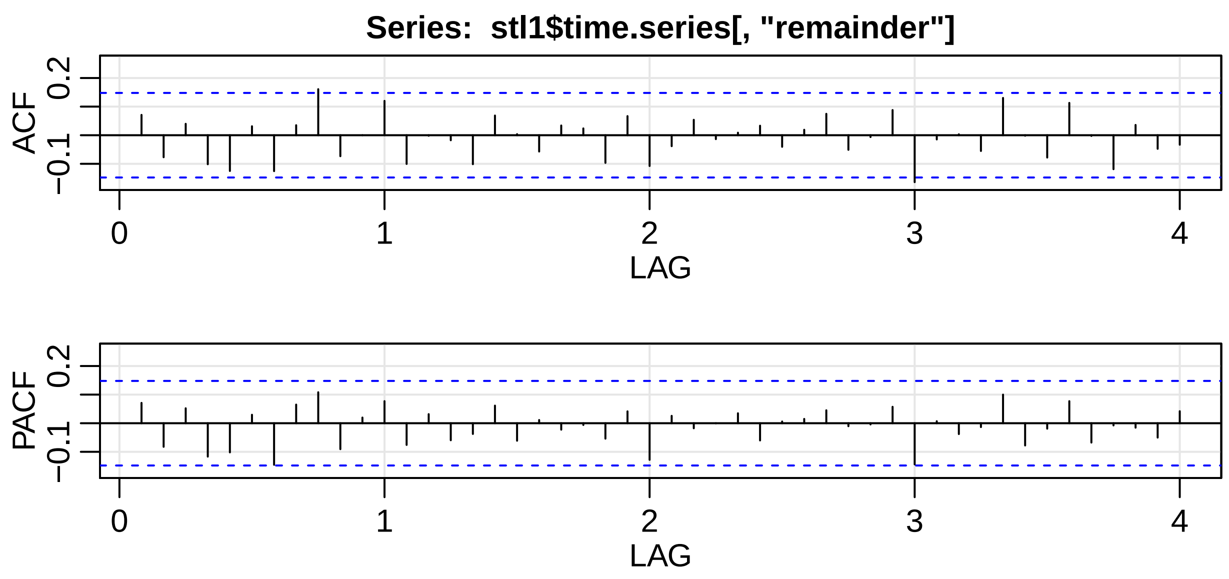




We observe a succesive increase in the first part of each year, reaching a peak in July and a significant drop in August. Another interesting thing is a very high value in September 2006.

Looking at the first plot, we concluded rather multiplicative behaviour of our data, so we apply log tranformation for stl decomposition function.

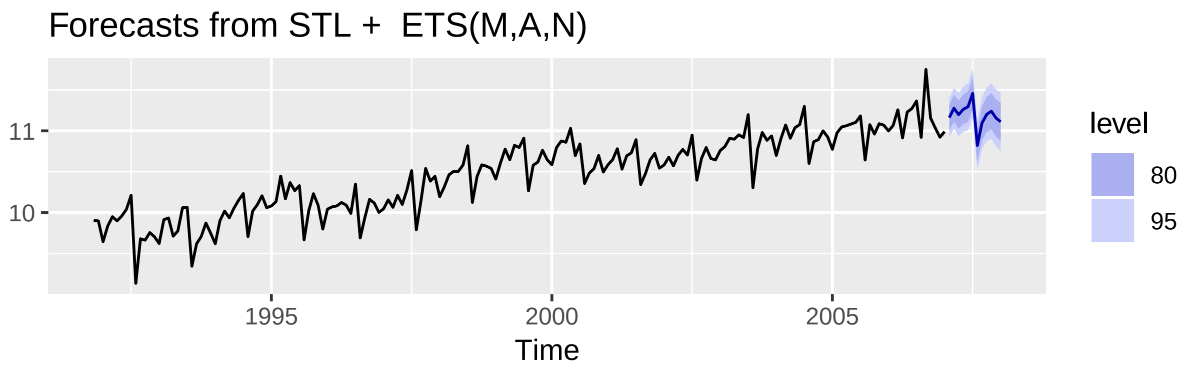
The Loess decomposition on logarithm transform of our data confirmed all of our previous observations, leaving rather random noise in the remainder. In order to make sure, we explore ACF and PACF plots for the remainder.

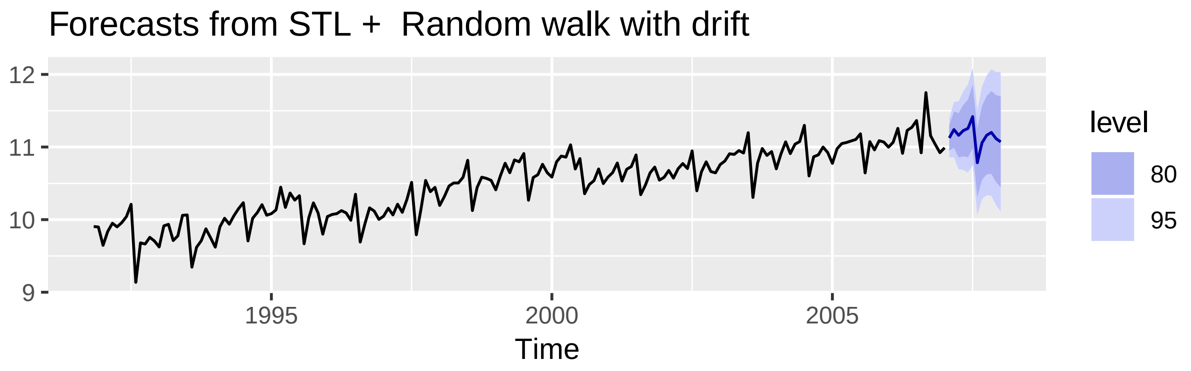


Looking at the plots we conclude, that in general correlations do not exceed the white noise bounds.

## Forecasting

We use 2 different methods for forecasting: “ets”, “rwdrift”. The plots can be seen below.





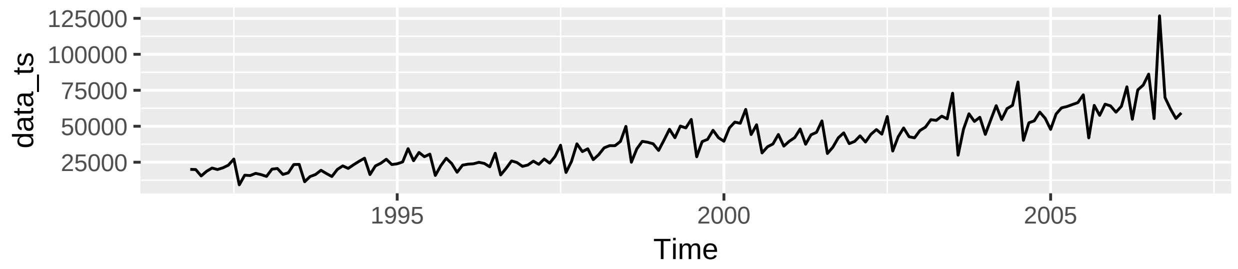
We evaluate our model’s accuracy using Cross Validation method. The results of two methods can be seen below:

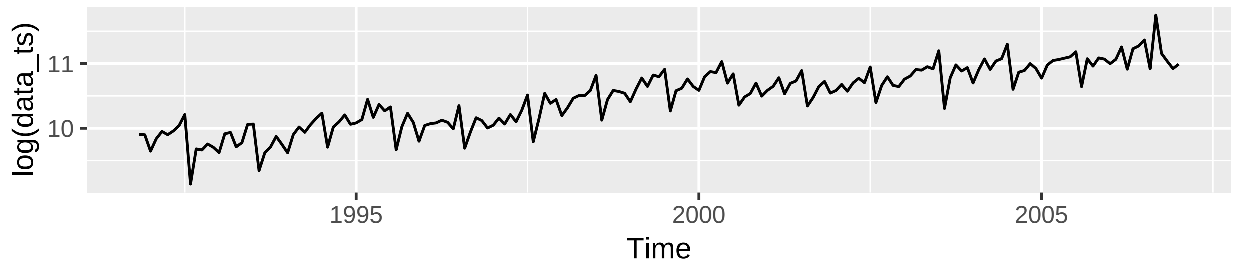
| | ets | rwdrift |
|---|---|---|
| RMSE | 46375.63 | 46375.6 |

Both methods seem to perform very similarily.

## Fitting ARIMA model

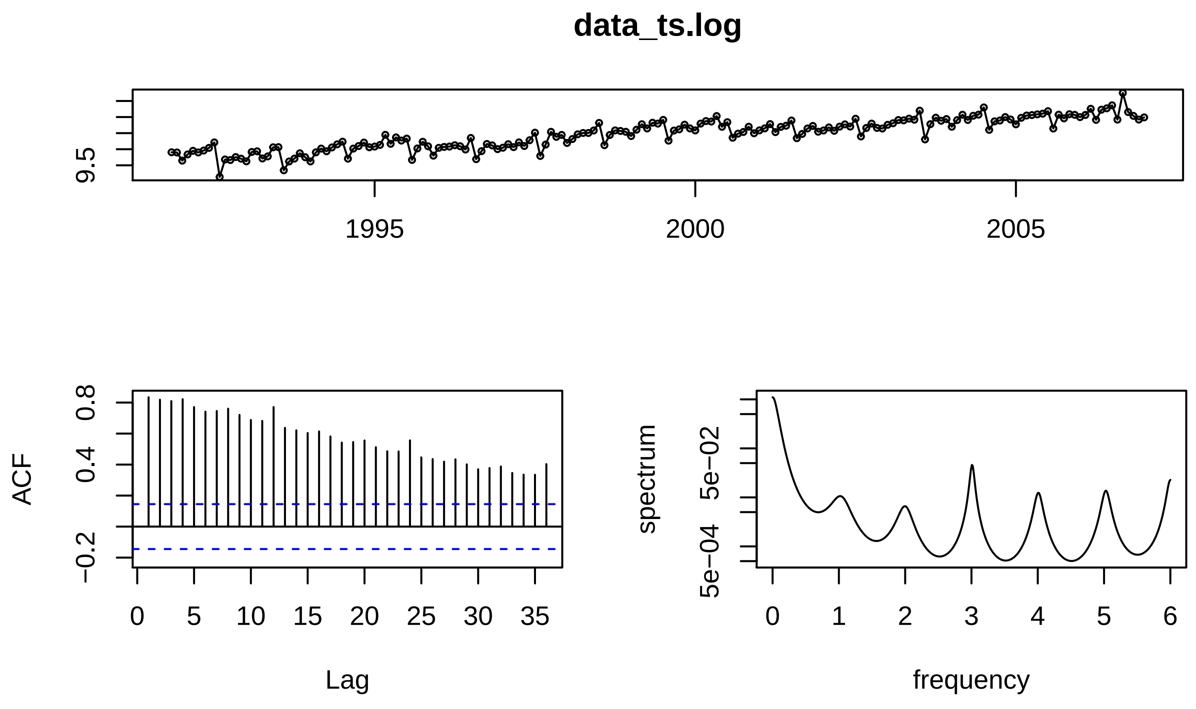
First of all we decide on the transformation we want to apply to our data.





As we can see above, the logarithm transformation stabilizes the variance in the data, so we decide to stick with our primal decision.

Now, we take a look at periodogram, is order to make sure we made a rigtht decision at the begining of our analysis about data seasonality.



We can see significant peaks each year, meaning our s parameter, as already stated in the beginning, should be s=12. This may be also visible on the seasonal plot we displayed earlier.

## Stationarity

As we concluded visually, our data is not stationary, here we conduct KPSS and ADF tests to confirm it.

```
## Warning in kpss.test(data_ts.log): p-value smaller than printed p-value
```

```
##
##  KPSS Test for Level Stationarity
##
## data:  data_ts.log
## KPSS Level = 4.2951, Truncation lag parameter = 3, p-value = 0.01
```

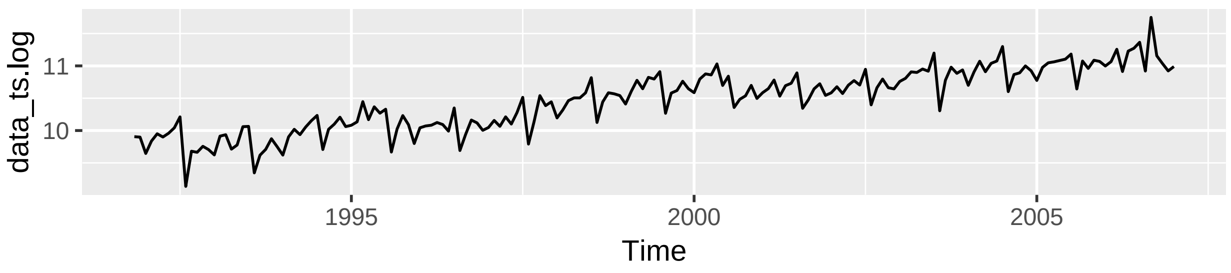
```
## Warning in adf.test(data_ts.log): p-value smaller than printed p-value
```

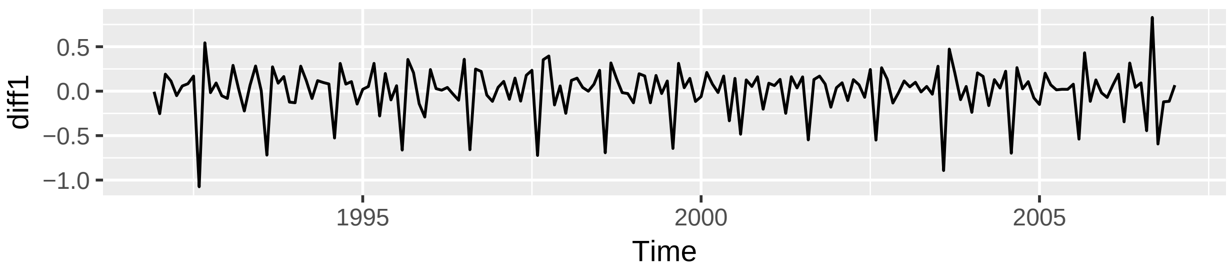
```
##
##  Augmented Dickey-Fuller Test
##
## data:  data_ts.log
## Dickey-Fuller = -4.2254, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

Both tests give rather contradictory results, as by KPSS test we reject stationarity and by ADF test we could accept the alternative hypothesis of stationarity. Nevertheless, the standard deviation is greater than zero, so we'll try to transform our data to achieve stationarity by differencing.

We first check by using ndiffs and nsdiffs functions, that are supossed to estimate the number of differences required to make a given time series stationary and the number of seasonal differences respectively.

We should be able to achieve stationarity by applying one differentiation and one seasonal differentiation. First, we try to detrend our data.





Visually we observe proper detrending of our data. We conduct again set of tests to check if the transform is sufficient to achieve stationarity.

```
## Warning in kpss.test(diff1): p-value greater than printed p-value
```

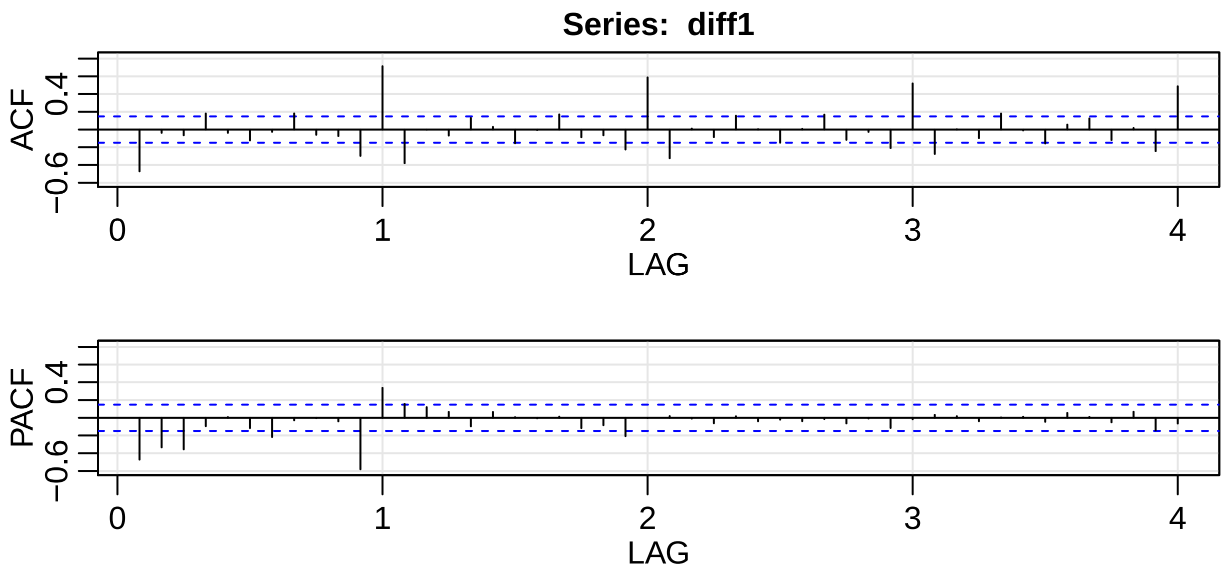
```
##
```

```
##  KPSS Test for Level Stationarity
##
## data:  diff1
## KPSS Level = 0.019232, Truncation lag parameter = 3, p-value = 0.1

## Warning in adf.test(diff1): p-value smaller than printed p-value

##
##  Augmented Dickey-Fuller Test
##
## data:  diff1
## Dickey-Fuller = -7.6419, Lag order = 5, p-value = 0.01
## alternative hypothesis: stationary
```

Both tests indicate that we achieved stationarity. Standard deviation also dropped. We'll take a look at ACF PACF plots.



There is no visible pattern on both plots above. Although, it already makes as state that first-order differentation is enough to achieve stationarity of our data, we check if applying seasonal differentation improves it, since we observed seasonal behaviour from the beginning.

| | sd.data_ts.log. | sd.diff1. | sd.diff2. |
|---|---|---|---|
| Standard deviation | 0.4726609 | 0.2667181 | 0.1576527 |

Following "The optimal order of differencing is often the order of differencing at which the standard deviation is lowest" advice, we will take it into account in ARIMA model construction.

We try different p,q,P,Q values configurations.

```
model.1=Arima(data_ts.log,order=c(1,1,1),seasonal=list(order=c(0,0,0), period=12))
model.2=Arima(data_ts.log,order=c(2,1,1),seasonal=list(order=c(0,1,1), period=12))
model.3=Arima(data_ts.log,order=c(2,1,1),seasonal=list(order=c(2,1,1), period=12))
model.4=Arima(data_ts.log,order=c(2,1,1),seasonal=list(order=c(0,1,3), period=12))
model.5=Arima(data_ts.log,order=c(2,1,0),seasonal=list(order=c(2,1,1), period=12))

aics = data.frame(model.1$aic, model.2$aic, model.3$aic, model.4$aic,model.5$aic)
kable(aics)%>% kable_styling(full_width = F)
```

| model.1.aic | model.2.aic | model.3.aic | model.4.aic | model.5.aic |
|---:|---:|---:|---:|---:|
| -50.15082 | -209.579 | -217.0291 | -214.7497 | -212.889 |

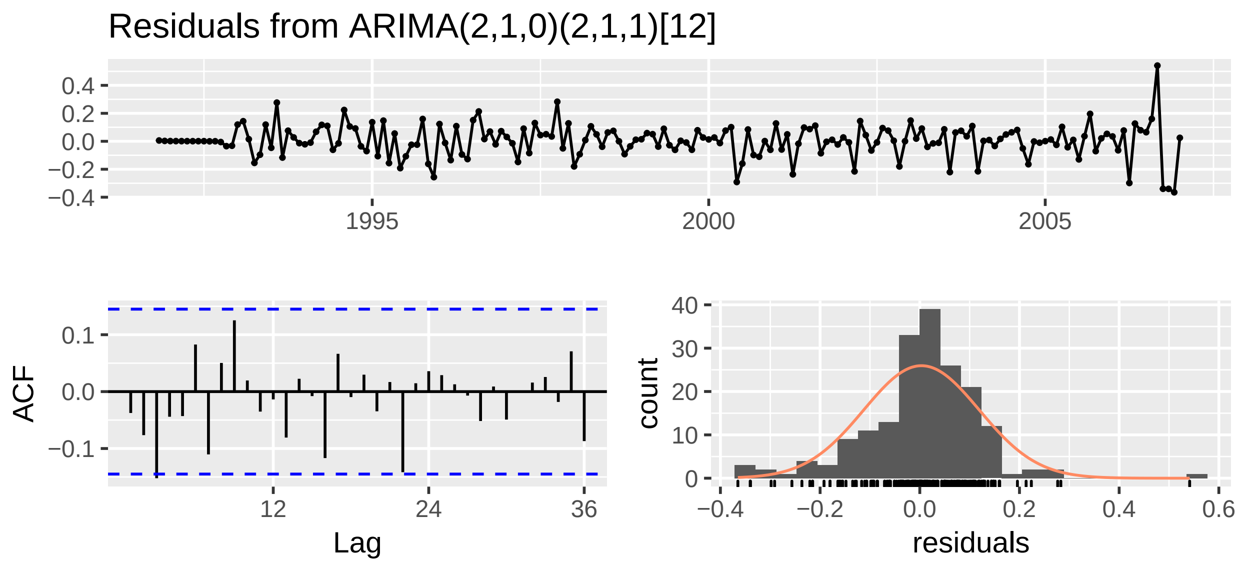
The worst model is the one without seasonal component, whereas the best model based on AIC is model number 3. We check correlations between coefficients.

| | ar1 | ar2 | ma1 | sar1 | sar2 | sma1 |
|---|---:|---:|---:|---:|---:|---:|
| ar1 | 1.0000000 | 0.5618688 | -0.8451657 | 0.1152077 | -0.0873044 | -0.0891600 |
| ar2 | 0.5618688 | 1.0000000 | -0.6754116 | 0.0097206 | 0.0077647 | 0.0441905 |
| ma1 | -0.8451657 | -0.6754116 | 1.0000000 | -0.0019933 | 0.1108613 | 0.0459616 |
| sar1 | 0.1152077 | 0.0097206 | -0.0019933 | 1.0000000 | 0.1106989 | -0.4729614 |
| sar2 | -0.0873044 | 0.0077647 | 0.1108613 | 0.1106989 | 1.0000000 | -0.2315977 |
| sma1 | -0.0891600 | 0.0441905 | 0.0459616 | -0.4729614 | -0.2315977 | 1.0000000 |

We observe high negative correlation between ar1 and ma1, therefore we check next model for its coefficients. Since model 4 has the same parameters for non-seasonal components, we check model nr 3, which does not include MA component.

| | ar1 | ar2 | sar1 | sar2 | sma1 |
|---|---:|---:|---:|---:|---:|
| ar1 | 1.0000000 | 0.4030987 | 0.2281928 | -0.0298206 | -0.1294042 |
| ar2 | 0.4030987 | 1.0000000 | 0.1602535 | 0.0865171 | -0.0068352 |
| sar1 | 0.2281928 | 0.1602535 | 1.0000000 | 0.1358572 | -0.4940980 |
| sar2 | -0.0298206 | 0.0865171 | 0.1358572 | 1.0000000 | -0.2522282 |
| sma1 | -0.1294042 | -0.0068352 | -0.4940980 | -0.2522282 | 1.0000000 |

It seems like coefficients look good enough. Now we check the residuals of the model.



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,1,0)(2,1,1)[12]
## Q* = 24.098, df = 19, p-value = 0.1924
##
## Model df: 5.   Total lags used: 24
```

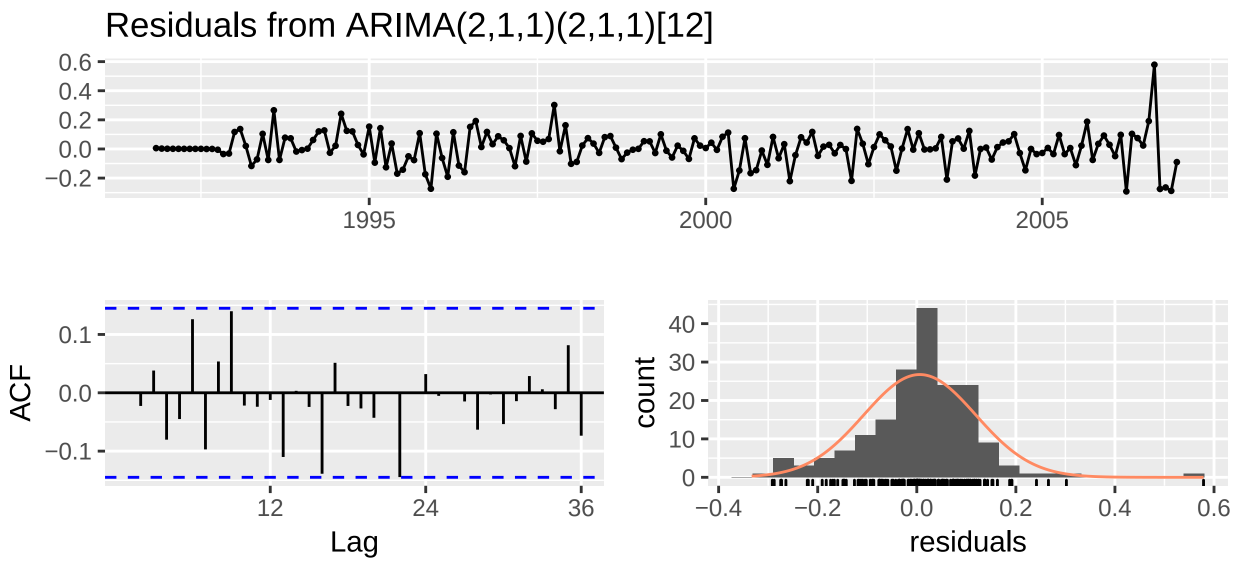
```
##
##  One Sample t-test
```

```
##
## data:  model.5$residuals
## t = 0.39785, df = 182, p-value = 0.6912
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.01349539  0.02031223
## sample estimates:
##  mean of x
## 0.00340842
```

```
##
##  Jarque Bera Test
##
## data:  model.5$residuals[-which.max(model.5$residuals)]
## X-squared = 28.313, df = 2, p-value = 7.111e-07
```

The autocorrelation plot of residuals looks reasonabily; we can also accept the hypothesis that true mean is equal to 0, and the hyopthesis of residuals independence (by Ljung-Box test). Nevertheless, residuals are not normally distributed, which can be seen already on the plot, but we also confirm it by Jarque Bera test. Outlier removal also didn't improve normality.

Just for sanity check and to not regret anything later, we checked the residuals of model 5, which we rejected becuse of coefficient correlations.



```
##
##  Ljung-Box test
##
## data:  Residuals from ARIMA(2,1,1)(2,1,1)[12]
## Q* = 23.639, df = 18, p-value = 0.1672
##
## Model df: 6.   Total lags used: 24
```

```
##
##  One Sample t-test
##
## data:  model.3$residuals
## t = 0.72221, df = 182, p-value = 0.4711
## alternative hypothesis: true mean is not equal to 0
## 95 percent confidence interval:
##  -0.01046047  0.02253952
## sample estimates:
##   mean of x
## 0.006039523
```

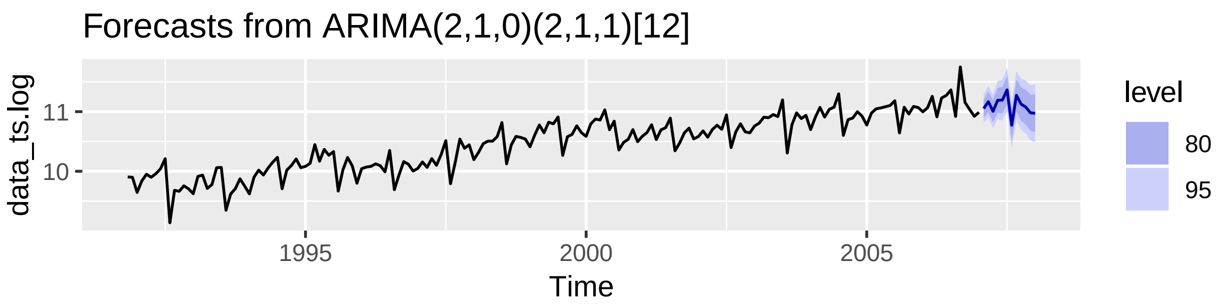
```
##
##  Jarque Bera Test
##
## data:  model.3$residuals[-which.max(model.3$residuals)]
## X-squared = 10.642, df = 2, p-value = 0.004888
```

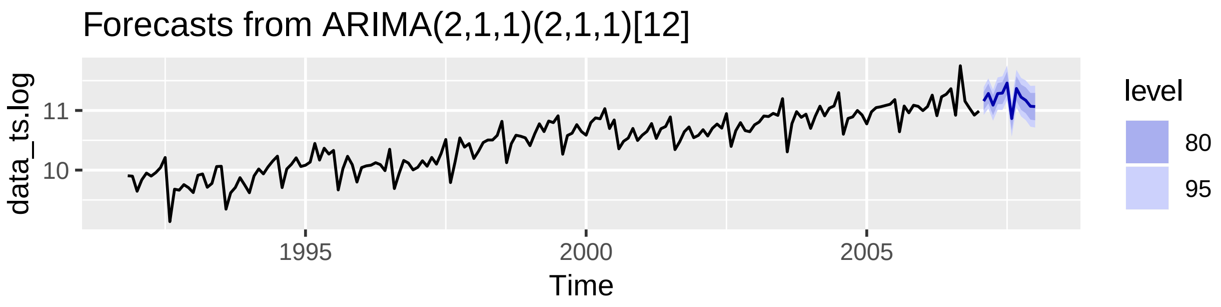
Model 3 performs similarily in terms of residuals. It may seem visually that residuals are "more normally" distributed, but it still fails Jarque Bera test.

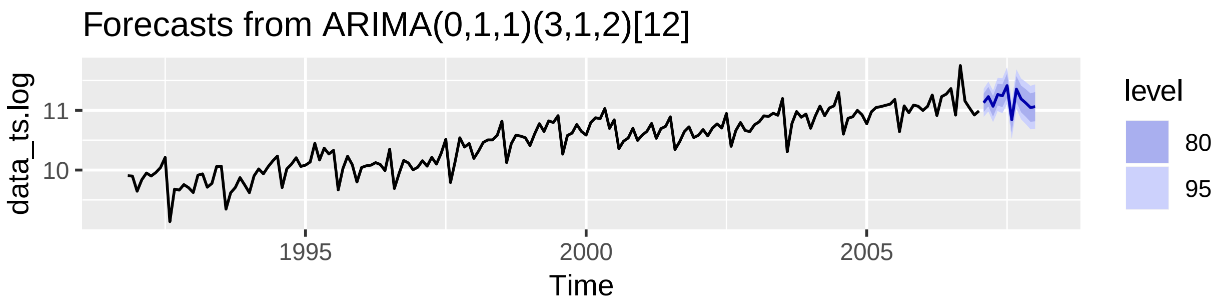
## Forecast and models' evaluation

As we cannot decide between those two models, we conduct the forecast for both of them and compare to auto ARIMA model.

```
## Series: data_ts.log
## ARIMA(0,1,1)(3,1,2)[12]
##
## Coefficients:
##          ma1    sar1     sar2     sar3     sma1    sma2
##       -0.623  0.4911  -0.3887  -0.1794  -1.0771  0.4307
## s.e.   0.076  0.3583   0.2040   0.1278   0.3287  0.4250
##
## sigma^2 estimated as 0.01422:  log likelihood=116.7
## AIC=-219.4   AICc=-218.71   BIC=-197.45
##
## Training set error measures:
##                      ME      RMSE        MAE        MPE      MAPE
## Training set 0.00504326 0.1128804 0.08037017 0.04785345 0.7663313
##                   MASE      ACF1
## Training set 0.5211356 0.0382414
```







And we compute RMSE for both of them keeping last 12 months as test data.

| | model3 | model5 | auto |
|---|---|---|---|
| RMSE | 0.2318172 | 0.2309788 | 0.2304854 |

Although the differences between models' RMSEs aren't big, unfortunately, we need to admit that Auto ARIMA model performs better than both of our proposed models. But since it requires more paramters, we therefore state that the best model is model.5 with the lowest RMSE on test data, and (what is apparently very important for us) without correlated coefficients.