Assignment 3 - Team 35

Applied Forecasting in Complex Systems 2021

Emiel Steegh (14002558) Nina Spreitzer (13725378)
Adam Mehdi Arafan (11595019) Riccardo Fiorista (14012987)

University of Amsterdam

November, 26, 2021

Note to the reader:
All of the code can be found in the appendix, starting on page 15

Exercise 1

1.1

This model results in a seasonal ARIMA model that has the following general notation $ARIMA(p,d,q)(P,D,Q)_m$, where

- m denotes the seasonal period,
- (p,d,q) presents the non-seasonal part and
- (P,D,Q) refers to the seasonal part of the model.

We can identify the following ARIMA model for η_T: $ARIMA(0,1,1)(2,1,0)_{12}$

The seasonal part in this model is based on monthly seasonality as our $m = 12$. Having d and D as 1 indicates that seasonal, as well as first differencing was necessary to make the original data stationary. The other components indicate how the ACF and PACF of the differenced data looks like. The PACF would show two significant spikes at lags 12 and 24, therefore the model includes a seasonal AR(2) ($P = 2$)but no non-seasonal AR component ($p = 0$). As the model has a non-seasonal MA(1) ($q = 1$) and no seasonal MA component ($Q = 0$), the ACF would have a significant spike in the ACF at lag 1 and the the early lags of the PACF would show a geometric decay.

1.2

The following system of equations serves as the basis of our forecast:

$$\begin{cases} y_t^* = \beta_1^* * x_{1,t}^* + \beta_2^* * x_{2,t}^* + \eta_t & (1) \\ (1 - \phi_1 B^{12} - \phi_2 B^{24}) * (1 - B) * (1 - B^{12}) * \eta_t = (1 + \theta_1 B) * \varepsilon_t & (2) \end{cases}$$

In order to get a more suitable equation for forecasting, we first need to apply differences on equation (1):

$$(1-B)*(1-B^{12})*y_t^* = \beta_1^**(1-B)*(1-B^{12})*x_{1,t}^*+\beta_2^**(1-B)*(1-B^{12})*x_{2,t}^*+(1-B)*(1-B^{12})*\eta_t$$

We will now replacing η_t and multiply by the AR polynomial $(1 - \phi_1 * B^{12} - \phi_2 * B^{24})$ to end up with one equation:

$$\begin{aligned}(1 - B) * (1 - B^{12}) * (1 - \phi_1 * B^{12} - \phi_2 * B^{24}) * y_t^* =\\ (1 - B) * (1 - B^{12}) * (1 - \phi_1 * B^{12} - \phi_2 * B^{24}) * \beta_1^* * x_{1,t}^* +\\ (1 - B) * (1 - B^{12}) * (1 - \phi_1 * B^{12} - \phi_2 * B^{24}) * \beta_2^* * x_{2,t}^* +\\ (1 + \theta_1 B) * \varepsilon_t\end{aligned}$$

The format of this equation is still in the backshift notation, where B means shifting the data back one period. Transforming the backshift notation results in:

$$\begin{aligned}&(y_t^* - y_{t-1}^* - y_{t-12}^* + y_{t-13}^*) - \phi_1 * (y_{t-12}^* - y_{t-13}^* - y_{t-24}^* + y_{t-25}^*) - \phi_2 * (y_{t-24}^* - y_{t-25}^* - y_{t-36}^* + y_{t-37}^*) =\\ &\beta_1 * (x_{1,t}^* - x_{1,t-1}^* - x_{1,t-12}^* + x_{1,t-13}^*) - \phi_1 * \beta_1 * (x_{1,t-12}^* - x_{1,t-13}^* - x_{t-24}^* + x_{1,t-25}^*) -\\ &\phi_2 * \beta_1 * (x_{1,t-24}^* - x_{1,t-25}^* - x_{1,t-36}^* + x_{1,t-37}^*) + \beta_2 * (x_{2,t}^* - x_{2,t-1}^* - x_{2,t-12}^* + x_{2,t-13}^*) -\\ &\phi_1 * \beta_2 * (x_{2,t-12}^* - x_{2,t-13}^* - x_{2,t-24}^* + x_{2,t-25}^*) - \phi_2 * \beta_2 * (x_{2,t-24}^* - x_{2,t-25}^* - x_{2,t-36}^* + x_{2,t-37}^*) +\\ &\varepsilon_t + \theta_1 * \varepsilon_{t-1}\end{aligned}$$

To solve y_t^* we will move everything else to the right side of the equation:

$$\begin{aligned}y_t^* = &\, y_{t-1}^* + y_{t-12}^* - y_{t-13}^* + \phi_1 * (y_{t-12}^* - y_{t-13}^* - y_{t-24}^* + y_{t-25}^*) + \phi_2 * (y_{t-24}^* - y_{t-25}^* - y_{t-36}^* + y_{t-37}^*) +\\ &\beta_1 * (x_{1,t}^* - x_{1,t-1}^* - x_{1,t-12}^* + x_{1,t-13}^*) - \phi_1 * \beta_1 * (x_{1,t-12}^* - x_{1,t-13}^* - x_{t-24}^* + x_{1,t-25}^*) -\\ &\phi_2 * \beta_1 * (x_{1,t-24}^* - x_{1,t-25}^* - x_{1,t-36}^* + x_{1,t-37}^*) + \beta_2 * (x_{2,t}^* - x_{2,t-1}^* - x_{2,t-12}^* + x_{2,t-13}^*) -\\ &\phi_1 * \beta_2 * (x_{2,t-12}^* - x_{2,t-13}^* - x_{2,t-24}^* + x_{2,t-25}) - \phi_2 * \beta_2 * (x_{2,t-24}^* - x_{2,t-25}^* - x_{2,t-36}^* + x_{2,t-37}^*) +\\ &\varepsilon_t + \theta_1 * \varepsilon_{t-1}\end{aligned}$$

1.3

The model retrieved in **(1.2)** can be used to forecast y_{t^*}, where $t = T + h$ for a certain time horizon h. As forecasting y_t^* is based on a regression model with ARIMA errors, we need to separately forecast the regression part as well as the ARIMA part of the model and combine the results.

For the regression part it is necessary to forecast the predictors. This can be done by applying a simple model such as the seasonal naive method. $x_{1,t}^*$ and $x_{2,t}^*$ can be used to calculate $x_{1,T+1}^*$ and $x_{2,T+1}^*$. Next, $\eta_T + 1$ needs to be calculated. Finding a good point estimate for ε_{T+1} should be based on the last residual $\hat{\varepsilon_t}$. As we assume white noise residuals, a good estimation is $\varepsilon_{T+1} = 0$.

Having all these components, the electricity demand forecast for, y_{T+1}^* can be made as all values for x^* are known up to the point where $t = T$.

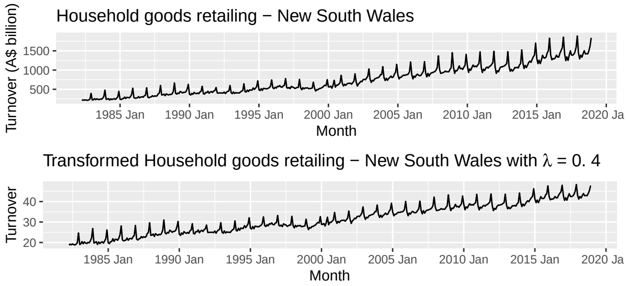
As shown for $h = 1$, the electricity demand for the next 12 months ($t = T + 1$, $t = T + 2$,..., $t = T + 12$) can be obtained in the same way.

Exercise 2

Through our random seed (55361019) we obtained the household good retailing dataset for New South Wales, covering the Turnover in billion Aus$ from April 1982 to December 2018. From the timeseries (TS) itself, but mainly by looking at the STL decomposition below, we notice a clear underlying positive trend in the data. Furthermore, we identify clear yearly seasonality, which increases from January until later and the Christmas season.

Note that the seasonality varies in magnitude over the levels and exhibits a multiplicative behaviour. Looking at the seasonally adjusted TS (remainder), we identify a progression against the yearly pattern with a much higher-than-usual expenditure in June 2000. Therefore we expect the harmonic regression in 2.1 to not be able to capture this and future outlier properly (as the model assumes seasonality to be fixed).

To stabilize the variance over the levels, we apply a Box-Cox transformation with $\lambda \approx 0.4$ (approximately equivalent to a quartic-root transformation). We see from the plot below that the applied transformation indeed stabilized the TS.

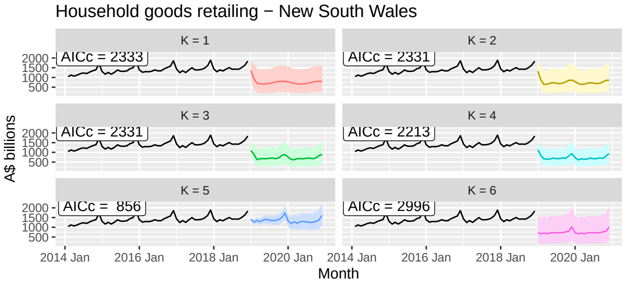


2.1

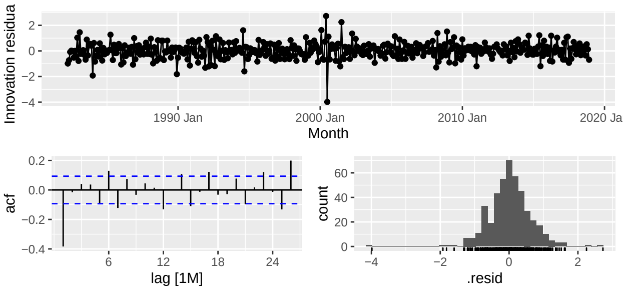
We model the dynamic regression with Fourier terms to capture seasonality, and use seasonal ARIMA errors to account for other dynamics. To find the appropriate model we compare models with 1 up to $\frac{\text{period}}{2} = 6$ Fourier terms where period = 12 months. The max Fourier-terms is limited at half of the seasonal period (1 year, i.e. 12 months in our case).

Furthermore, we let the system choose the optimal autoregressive (AR) and moving average (MA) components for the trend and seasonal components of the ARIMA part within set bounds. We set differencing to 0 as the Fourier part fits the entire TS and the ARIMA part the resulting residuals, meaning that we cannot differentiate against any data point.

As can be seen from the comparison below, the model with the maximum number of Fourier terms $K = 5$ performs best with AICc = 856 and visibly the most narrow prediction interval. Furthermore, the predicted values seem plausible considering the most recent data (we plot the training data from January 2014 onwards for better visual experience).



2.2



Above we see the residuals from our previously identified best harmonic dynamic regression model with $K = 5$ Fourier terms. We notice, as expected, that the model does not capture the outlier event in June 2000, and the residual for that time step and the autocorrelation are extremely high. Furthermore, we see that a considerable amount of the autocorrelations for the first 26 lags are not within the 95% CI around 0.

With a Box-Ljung test with dof = 13 (as we have 13 estimated parameters as shown in the model's report) and 24 lags, we confirm our observation with p-value $\approx 0 < \alpha$ ($\alpha = 0.05$) and the rejection of H_0, namely that the residuals are not 0. Moreover, the histogram of the residuals shows long tails. These issues may affect the coverage of the prediction intervals, but the point estimates should be admissible (following Chapter 10.7 of Forecasting: Principles and Practice (3rd Edition)).

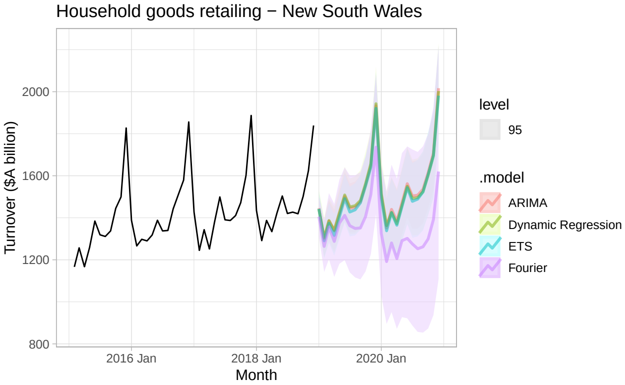
Box-Ljung for `K = 07` model

with 13 DoF and 24 lags

.model	lb_stat	lb_pvalue
K = 5	129	0

2.3

We perform the fitting on the Box-Cox transformed TS for all four inspected models, namely the 5 Fourier-term harmonic regression identified above, ETS, Auto-ARIMA, and dynamic regression.

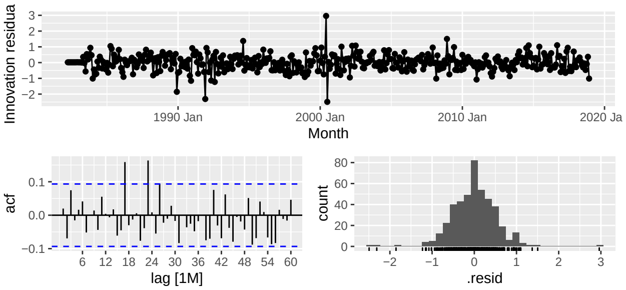


Comparing the four chosen models, we identify that the dynamic regression model is performing best with an AICc of 667, the highest log-likelihood. However, the residuals of the dynamic regression model are not distributed like white noise (appendix 2.2). Furthermore, the residual's variance (`sigma2`) is higher than the closely performing Auto-ARIMA model. Because of the lower variance in the residuals as well as the closely performing AICc and high log-likelihood, we choose the ARIMA model as the preferred one which still performs significantly better than the harmonic regression.

Household good retailing - NSW: Model Comparison Metrics

.model	sigma2	log_lik	AIC	AICc
Dynamic Regression	0.331	-324	666	667
ARIMA	0.270	-329	673	673
Fourier	0.384	-413	855	856
ETS	0.295	-1065	2165	2166

We see from the residuals below and the Box-Ljung test that the residuals follow a white-noise distribution. The histogram is normally distributed, centred around 0, and the Box-Ljung test returns p-value $= 0.155 > \alpha$ with $\alpha = 0.05$, meaning we cannot reject H_0, namely that the residuals indeed are 0. Furthermore, the Auto-ARIMA chose AR 1, MA 1 for the trend and AR 1, D 1, MA 2 for the seasonal component with the seasonal period $m = 12$ and included drift.



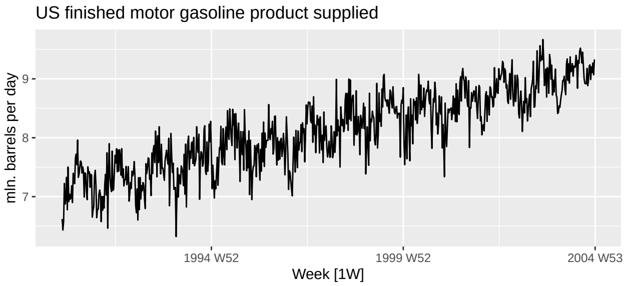
Box-Ljung for `K = 07` model

with 13 DoF and 24 lags

.model	lb_stat	lb_pvalue
ARIMA	9.34	0.155

Exercise 3

3.1)



As always we start by plotting the data.

An immediate problem is the amount of weeks in a year. First, on average there are 52.18 weeks in a year, individually they can be 52 or 53 weeks long. Having a non integer Second, 52 is a long period for seasonality, most models are not built to deal with that.

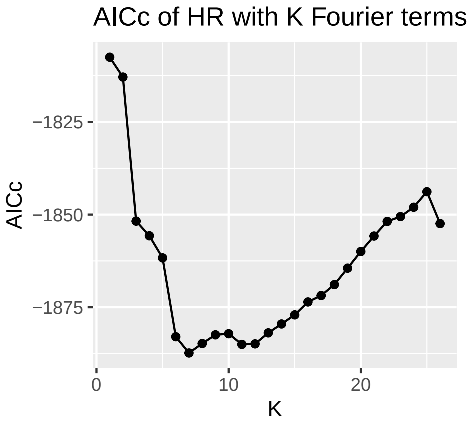
In the data we observe:
- a steady upward trend
- seasonality with a period of a year (52.18 weeks)
- light cyclicity obfuscated by noise/variance
- a lot of variance, but it looks stable until 2002-2003 after which point the variance seems to get smaller.

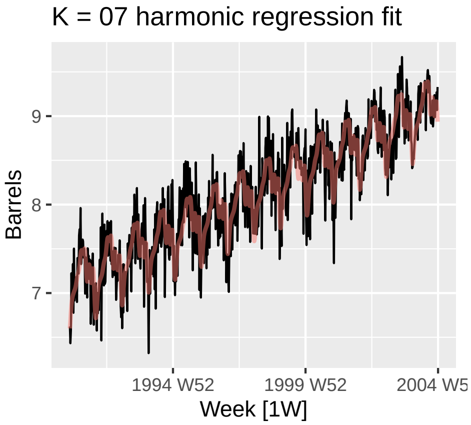
In the seasonal plot and ACF (Appendix 3.1) the seasonality becomes obvious, and so does some weird behaviour of the season length (e.g. some seasons are longer than others, the season start peaks in the ACF sometimes miss the season length indicators).

Any attempted transformation did not provide a significant enough benefit. The Guerrero test provided us with $\lambda = 1.5$ but performing a BoxCox transformation with this λ made almost no change (Appendix 3.1).

Because we have such long periods, Harmonic regression with fourier terms is likely to work well compared to other standard models. The only downside to this model is that we operate on the assumption that the seasonal pattern does not change, but the assumption seems true for our data.

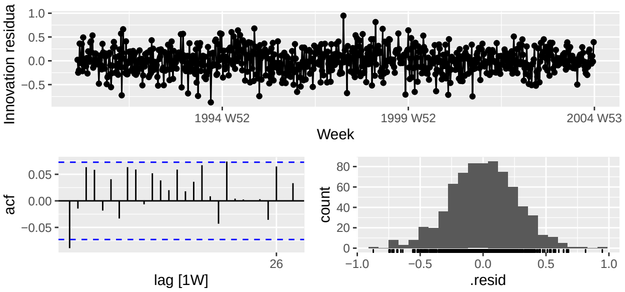
$K_{max} = \frac{m}{2} = 26$ so we fit 26 models with $K \in [1, 26]$ (Appendix 3.1).





A fit comparisson of $K(1, 6, 7, 8, 9, 26)$ is visible in Appendix 3.1. There is a lot of variance in the seasonal periods of the data. $K = 1$ captures way too little information, as we increase K the fourier terms are able to capture more wiggliness. The AICc vs K plot shows that $K = 7$ minimizes the AICc $= -1887$ and CV $= 0.0742$ (The CV plot is not made because it is asymptotically equivalent to the AICc).

Above $K = 7$ the model gets too wiggly for the variance in the model and becomes less accurate again. So, we pick `K = 07` as best model.



Checking the residuals for the harmonic regression with 7 fourier terms leads us to believe they behave almost like white noise. Except for two lags (which are barely outside the zone), all of them are within the confidence interval, yet they look biased to the positive side. The distribution looks like a normal one although perhaps a little left skewed. The timeline variance looks like it has a zero mean but looks a little unstable, towards the end the variance seems to decrease.

Box-Ljung for `K = 07` model

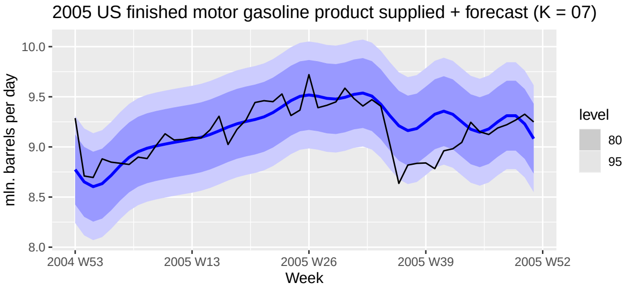
with 16 DoF and 53 lags

.model	lb_stat	lb_pvalue
K = 07	58.5	0.0137

We picked `lag=53` for the test. Because we are dealing with a very long seasonal period, instead of the more common $2m$, we went with the forecasting horizon of one year.
The Ljung-Box gives us $p = 0.0137 < 0.05$ so our residuals still have dependence on each other. However, with our visual residual inspection we still conclude that the model is pretty decent, but there is room for improvement.

3.2)

We forecast the TS for 2005 with the model obtained in the previous exercise.



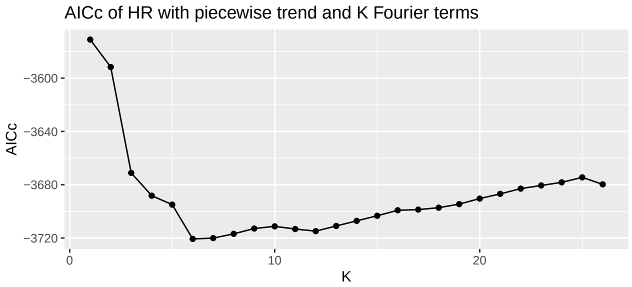
HR with 7 Foruier terms accuracy
as fitted to the pre 2005 US gasoline dataset, prediction on 2005

.model	RMSE	MAE	MPE	MAPE
K = 07	0.203	0.145	-0.501	1.6

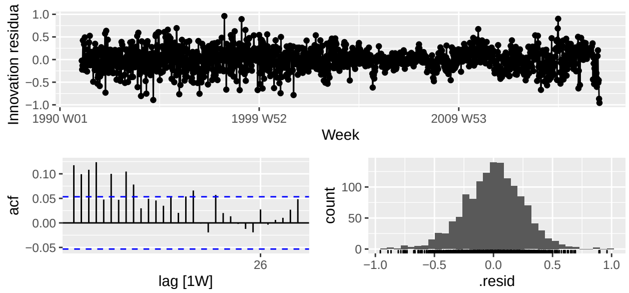
Until halfway through the year the model seems to follow the actual data well, but around week 32 the real data drops much more than the prediction. We also see that our prediction is much smoother than the actual data.

From the plot of the `us_gasoline` dataset, the decomposition of the dataset (Appendix 3.3) and some trial and error, we find that the elbows of the time series lie approximately at `t=2007`(coinciding with the 2008 financial crisis) and `t=2013`. There is an upward trend before (late) 2007, downward from 2007-2013 and then upward again.

We do the same trick as before, making 26 TSLM models for all $K \in [1, \frac{m}{2}]$



From the AICc vs Fourier terms plot we find that the `K = 06` model performs best in terms of AICc. Any more and the AICc gets worse. $K = 6$ is also the point at which the AICc curve of **(3.1)** lost most of it's downward steepness.



The residuals of this model don't look like white noise, less so than the previous model, not because it is necessarily worse, but probably because it is fit to data that exhibits a more complex pattern. Many of the ACF's lags are outside of the confidence interval and they tend to the positive side. The residuals timeline has a bowtie pattern variance and the distribution tells us that we may not have a zero mean. Especially to the right of the bowtie the variance mean does not look stable at zero but goes up and down.

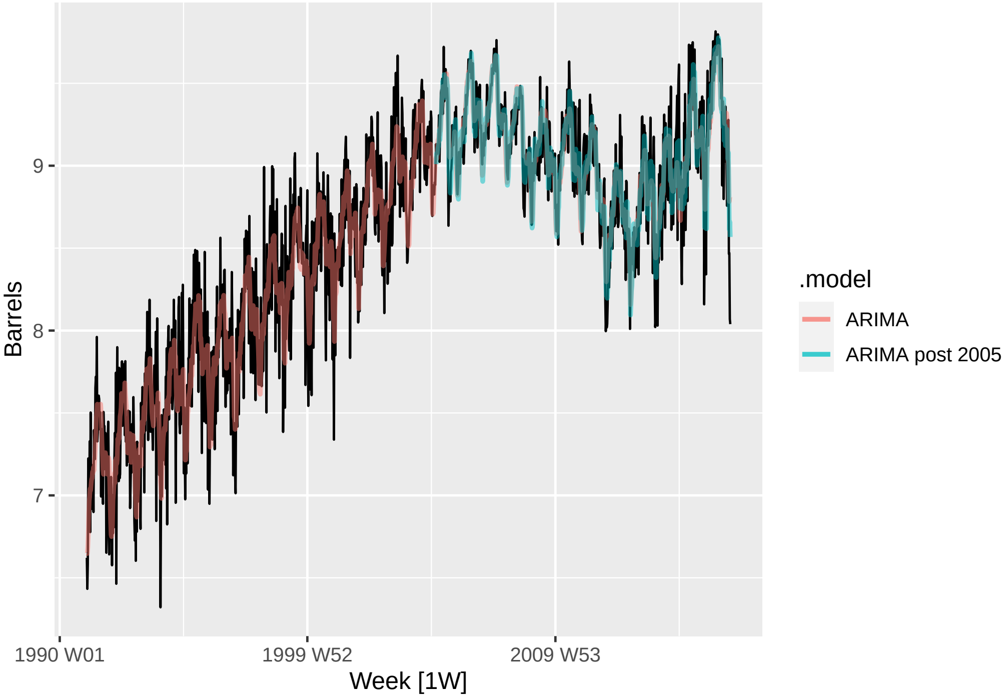
3.3)

We make a new model using ARIMA instead of TSLM, still using knots at the same places and with 6 Fourier terms. Because we use ARIMA now we want to restrict it's seasonal component with `PDQ(0:1,0,0:1)`, because seasonality should almost entirely be taken care of by the fourier terms.

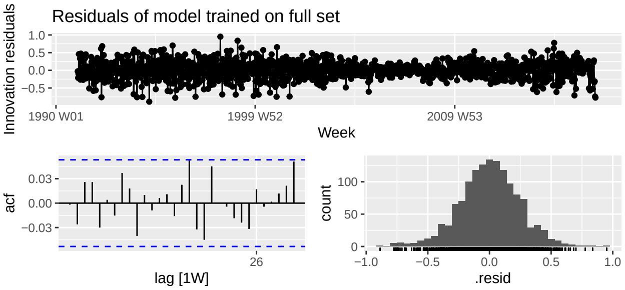
Note that we also fit the same model to the TS after 2005, more about that decision below.

ARIMA HR w/ 6 Fourier terms and knots

.model	sigma2	log_lik	AIC	AICc	BIC
ARIMA	0.0599	-5.19	54.4	55.1	169



Using `ARIMA` instead of `TSLM` added a lot of missing complexity to the model. Auto ARIMA with a piecewise trend with knots at 2007 and 2013 `+ fourier(K=6) + PDQ(0:1,0,0:1)` found the model `ARIMA(2,0,2)(1,0,0)` as best model.



The following section is about the model trained on the full dataset: The resiudal ACF look randomly distributed, there are no lags perpetrating the confidence interval. The residual distribution looks uniform and centered around 0. However, there is still a very visible bowtie in the residual timeline. This is likely caused by the time series' inherent heteroscedasticity: there is a decrease in variation in the 2004-2012 period. An STL decomposition (Appendix 3.3) shows that variance bowtie quite clearly. Unsurprisingly, a transformation gives no tangible benefit because the changes in variance level are not linear or quadratic. Which is why we tried the post 2005 data model.

Box-Ljung two ARIMA + HR models

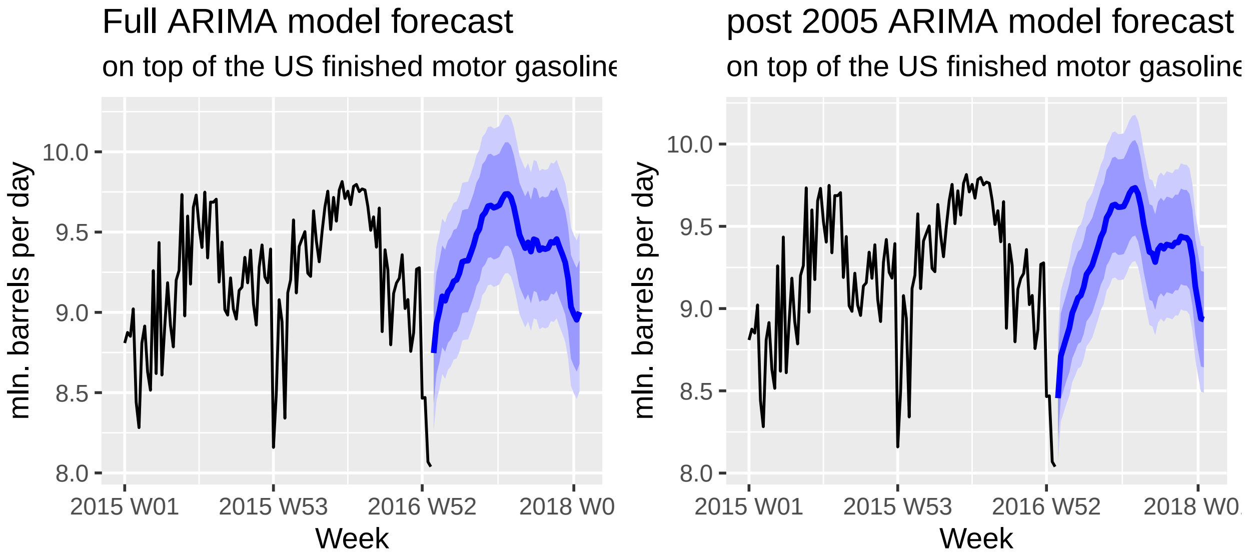
with 21 DoF and 53 lags

.model	lb_stat	lb_pvalue
ARIMA	51.2	0.0171
ARIMA post 2005	38.6	0.1974

We made a second model that only uses data after 2005 (after the pinch of the bowtie), in the hopes that trend in variance will be of a less complex nature: presenting only growth. The ARIMA part obtained for the `post 2005` is an `ARIMA(1,0,4)(0,0,1)`. In the plot above, we can see that the models behave very similarly.

Using the smaller cut of data improves how the residuals of the model looks only marginally (Appendix 3.3), but the residuals are still not homoscedastic. On the other hand, the Box-Ljung score improves a lot. It improved so much that we can say that $p = 0.1974 > \alpha$: the second model fails to reject H0. We conclude that the results are independently distributed, and behave like white noise. However, cheering here may be a bit naïve, as the `post 2005` model is fit to more recent data so will probably perform better on recent data (i.e. the 53 lags).

Luckily for us, the auto-correlation of both models is very small. So even though there is heteroscedasticity present in the residuals, it will likely have little effect and we conclude that both models are acceptable. With that being said, we forecast a year with both.



Full ARIMA model forecast

on top of the US finished motor gasoline

post 2005 ARIMA model forecast

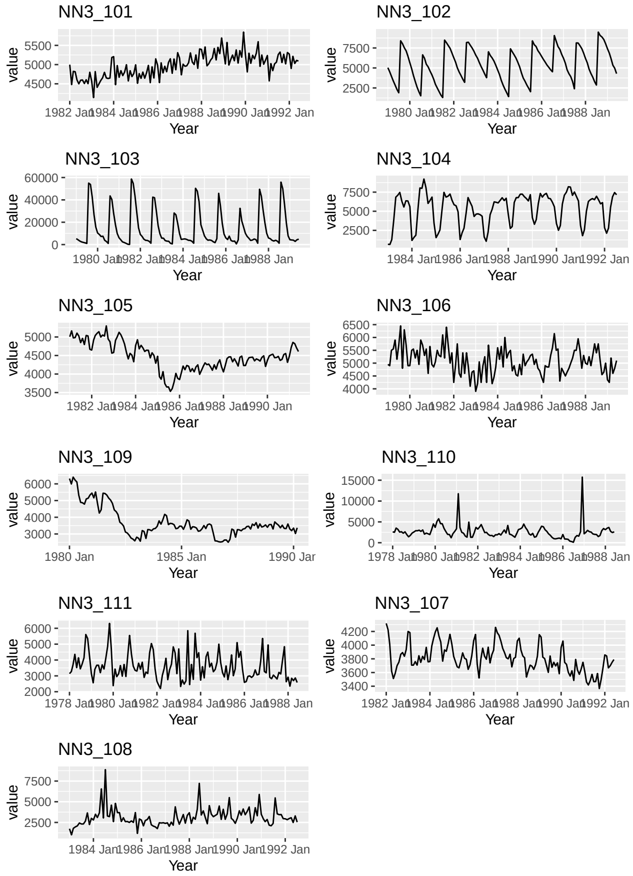
on top of the US finished motor gasoline

The `post 2005` model looks to start closer to the actual data and has a narrower confidence interval. This combined with the Ljung-Box test makes the `post 2005` ARIMA model our recommendation use in the real world.

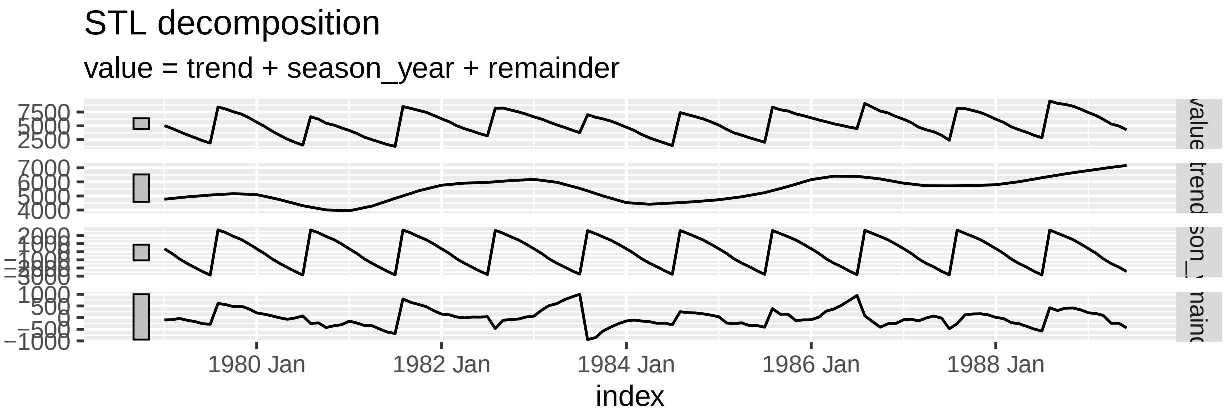
Exercise 4

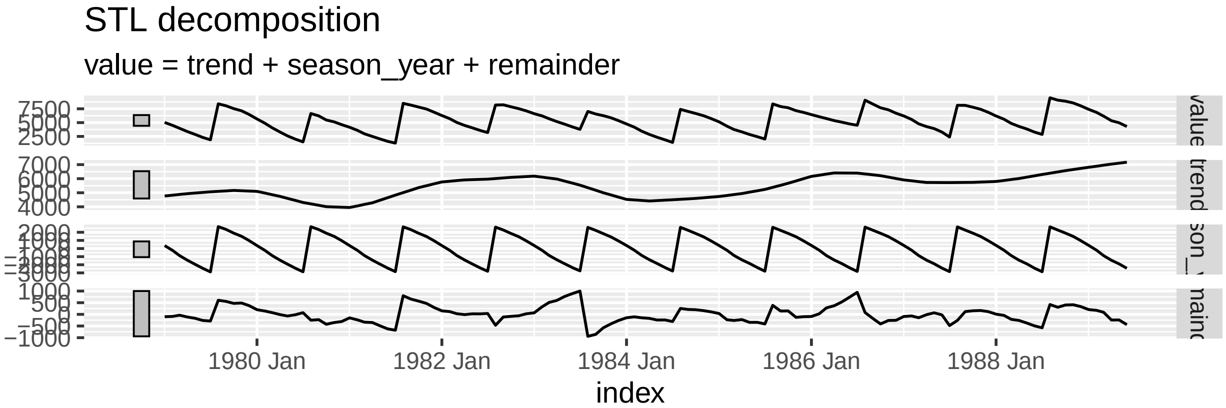
This exercise is quite code/function heavy, all of that is available in the appendix.

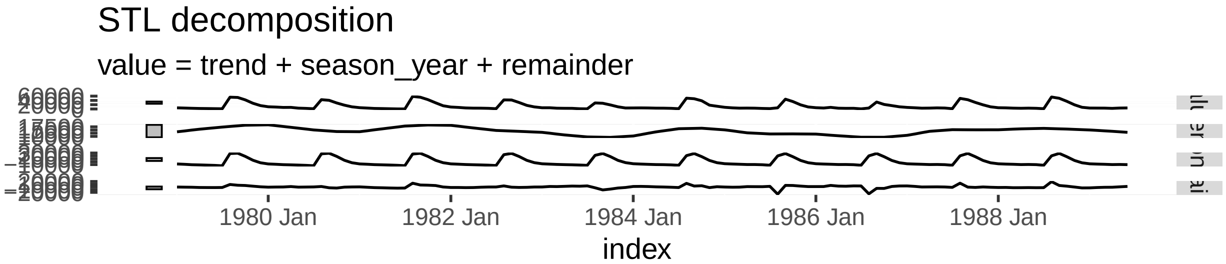
4.1)

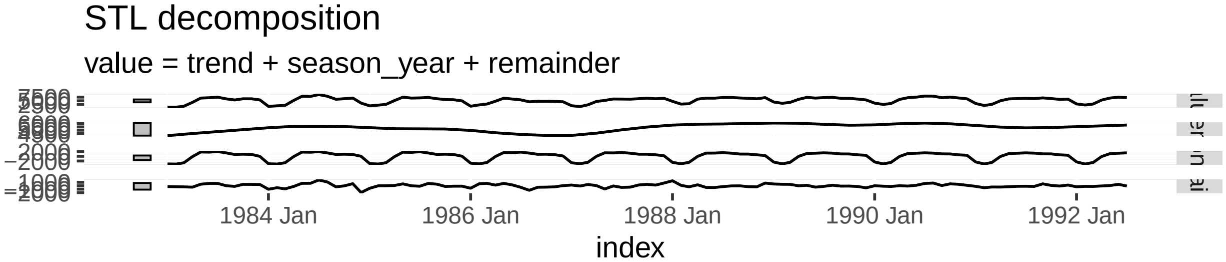


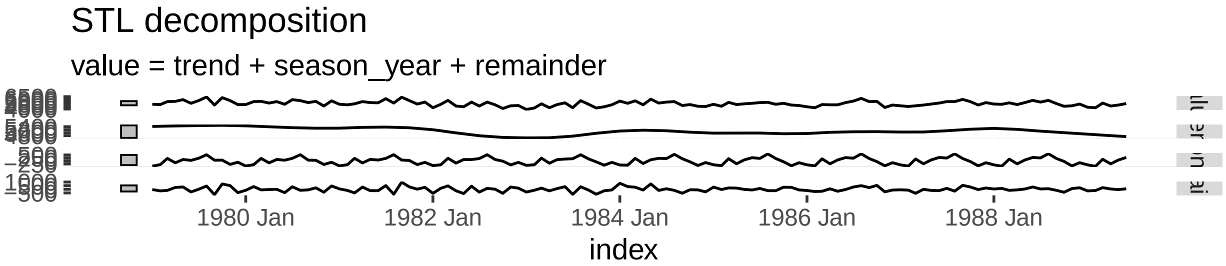
Judging from the visual analysis (appendix 4.1), time series 101 and 106 could potentially contain a seasonal pattern made invisible due to noise. Although 102, 103 and 104 already have a seasonal pattern visible in the plots (appendix 4.1), the influence trend has on seasonality will be further investigated. An STL decomposition is therefore performed below for further analysis of 101, 102, 103, 104 and 106.


STL decomposition
value = trend + season_year + remainder
7500
5000
2500
7000
6000
5000
4000
2000
1000
0
−1000
−2000
−3000
1000
500
0
−500
−1000
value
trend
son_
naind
1980 Jan
1982 Jan
1984 Jan
1986 Jan
1988 Jan
index


STL decomposition
value = trend + season_year + remainder
7500
5000
2500
7000
6000
5000
4000
2000
1000
0
−1000
−2000
−3000
1000
500
0
−500
−1000
value
trend
son_
naind
1980 Jan
1982 Jan
1984 Jan
1986 Jan
1988 Jan
index


STL decomposition
value = trend + season_year + remainder
60000
20000
17500
10000
−10000
−20000
1980 Jan
1982 Jan
1984 Jan
1986 Jan
1988 Jan
index


STL decomposition
value = trend + season_year + remainder
7500
2500
6000
4000
2000
−2000
1000
−2000
1984 Jan
1986 Jan
1988 Jan
1990 Jan
1992 Jan
index


STL decomposition
value = trend + season_year + remainder
6500
4000
5400
4800
500
−250
1000
−500
1980 Jan
1982 Jan
1984 Jan
1986 Jan
1988 Jan
index

The STL decomposition above reveals the seasonal pattern as well as substantial amount of variability in the trend. These changes in the time series could potentially be caused by the unpredictability of traffic as these timeseries represent transportation data. In fact, although clear seasonal patterns are visible at the end of the year (the lowest point appears to be January for most timeseries above), the relative volume reflected in the trend has significant amounts of variability, due to the constant change in optimal transportation methods over the years.

4.2

```
## TimeSeries no. 1 has a lambda value of 0.71
## TimeSeries no. 2 has a lambda value of 1.6
## TimeSeries no. 3 has a lambda value of -0.6
## TimeSeries no. 4 has a lambda value of 1.45
## TimeSeries no. 5 has a lambda value of 1.21
## TimeSeries no. 6 has a lambda value of 0.92
## TimeSeries no. 7 has a lambda value of -0.9
## TimeSeries no. 8 has a lambda value of -0.77
## TimeSeries no. 9 has a lambda value of 0.39
## TimeSeries no. 10 has a lambda value of 1.3
## TimeSeries no. 11 has a lambda value of -0.9
```

With the method defined in appendix 4.2, we add the missing accuracy metrics.

Accuracy measures

sorted by MSE

.model	MAE	MAPE	RMSE	SMAPE	MSE
seasonal101	110	2.07	137	2.07	18811
damped101	112	2.10	139	2.10	19185
ETSsimple101	112	2.12	141	2.12	19854
arima101	121	2.27	150	2.27	22624
naive101	179	3.34	224	3.34	50275
mean101	292	5.45	336	5.45	112883

Out of all eleven forecasts the forecasts below were considered the best according to the below error metrics:

```
res_acc %>%
    select(MAE) %>%
    unlist() %>%
    which.min()  # Refers to ARIMA 101
```

```
## MAE6
##    6
```

```
res_acc %>%
    select(MAPE) %>%
    unlist() %>%
    which.min()  # Refers to ARIMA 101
```

```
## MAPE6
##     6
```

```
res_acc %>%
    select(RMSE) %>%
    unlist() %>%
    which.min()  # Refers to ETSsimple105
```

```
## RMSE25
##     25
```

```
res_smape_mse %>%
    select(SMAPE) %>%
    unlist() %>%
    which.min()  # Refers to seasonal 101
```

```
## SMAPE4
##      4
```

```
res_smape_mse %>%
    select(MSE) %>%
    unlist() %>%
    which.min()  # Refers to to ETSsimple105
```

```
## MSE4
##    4
```

The MAE and MAPE error scores both reach a minimum for the ARIMA 101 time series. As the MAPE penalizes negative errors more than positive errors as well as extreme values (an issue which is resolved by SMAPE), the first forecast minimizes the error according to these metrics. More robust metrics such as the RMSE and MSE converged on the simple exponential smoothing model for time series 105 as minimizing these metrics usually lead to the mean with the only difference between MSE and RMSE being explainability. Finally, according to SMAPE, the best model is ETSsimple 105. This decision is unique among all other metrics due to the symmetric nature of the metric. However, although SMAPE tends to the weaknesses of MAPE and MAE, the use of the metric remains heavily discouraged due to its unstable nature.

4.3)

Based on visual analysis(plots in appendix) and the metrics extracted in the results table fully placed in the appendix, the best forecast for each time series has been chosen. For 101 and 102, all metrics except MSE, converged towards the ARIMA models as the most accurate. The outlying score provided by the MSE can be explained by the score's tendency to converge towards the mean, which penalizes the forecast in the case where the trend is shifting during the last segment of the time series. As 101 and 102 both have a change in trend towards the end, the MSE will heavily penalize these forecasts. ARIMA is the best model for both 103 and 104 with the RMSE being the most penalizing metric, in a similar fashion to the previous two series. As the RMSE also converges to the mean, a change in trend towards the end of the series will also heavily affect the score, 104 especially, suffers from a rapidly decreasing trend between 1993 and 1994, which was successfully captured by the ARIMA model and subsequently penalized by the RMSE.

In the case of 105, 106 and 107, all metrics agree on the best model, with as a result, low variability between each score. For 105, the simple Exponential smoothing captures most information while in 106 and 107, the arima model is best. All scores converge to the same model in 105 and 106 due to the relatively low variability in the series compared to other time series, while in 107, the RMSE shows slight deviance from the rest due to the up-shifting trend pattern towards the end. In 108, the mean model seems to be the best for this time series as the increasing and decreasing trends seem to average out.

Finally, for timeseries 109, 110 and 111, seasonal, simple exponential smoothing and Damped models achieve the best results. Due to the absence of spiking trends toward the end 109 and 111 all achieve low variabiliy between metric scores intra-model. However 110 performs poorly with RMSE and MSE due to the change in trend towards the end of the series.

To conclude, each of these metrics are better suited for certain time series, some metrics such as SMAPE are often unreliable but sometimes useful, while other metrics such as RMSE and MSE are heavily trusted, however seem to punish sudden trend changes towards the end of a series, although it is perfectly historically justifiable. The use of metrics depends on the series' seasonal and trend patterns as well as other factors outlined above and in **(4.2)**

Appendix

Note: Due to an error in Knitr that we have not been able to fix, ggplot titles and labels are not being wrapped properly. However, you are able to read the contents in the graphs just fine.

Setup code

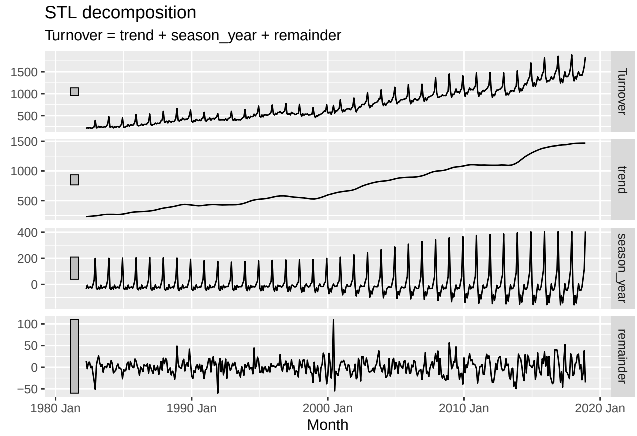
```
options(digits = 3)
library(fpp3)
library(latex2exp)
library(forecast)
library(formatR)
library(gridExtra)
library(gt)
library(glue)

knitr::opts_chunk$set(echo = FALSE, message = FALSE, warning = FALSE,
    cache = TRUE, dev.args = list(pointsize = 11), out.width = "50%",
    fig.align = "center")
knitr::opts_chunk$set(tidy.opts = list(width.cutoff = 60), tidy = TRUE)
```

2.1 code

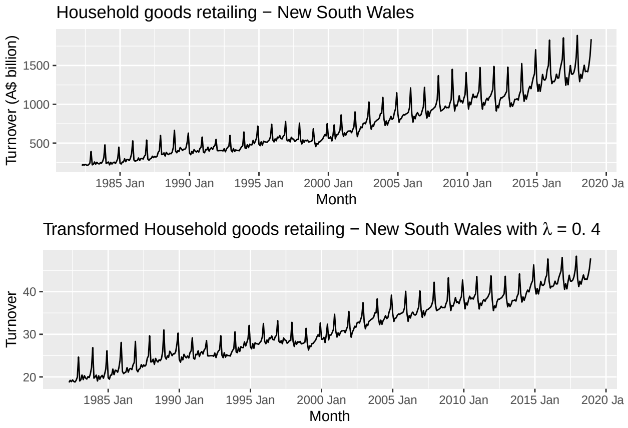
```
set.seed(55361019)
data_2 <- aus_retail %>%
    filter(`Series ID` == sample(aus_retail$`Series ID`, 1))
```

```
lambda <- data_2 %>%
    features(Turnover, features = guerrero) %>%
    pull(lambda_guerrero)
```

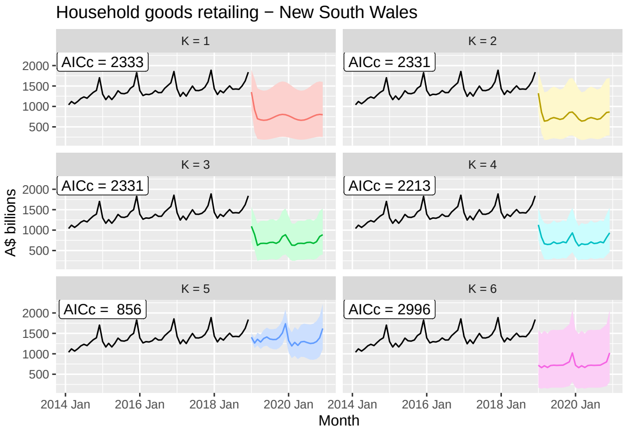
```
dcmp <- data_2 %>%
    model(STL(Turnover))
components(dcmp) %>%
    autoplot()
```



```
p0 <- data_2 %>%
    autoplot(Turnover) + labs(title = sprintf("%s - %s", data_2$Industry,
    data_2$State), y = "Turnover (A$ billion)", x = "Month") +
    scale_x_yearmonth(breaks = "5 years")

p1 <- data_2 %>%
    autoplot(box_cox(Turnover, lambda)) + labs(title = latex2exp::TeX(paste0(sprintf("Transfor
    data_2$Industry, data_2$State), round(lambda, 2))), y = "Turnover",
    x = "Month") + scale_x_yearmonth(breaks = "5 years")

p0 <- ggplotGrob(p0)
p1 <- ggplotGrob(p1)
grid.arrange(p0, p1, ncol = 1)
```

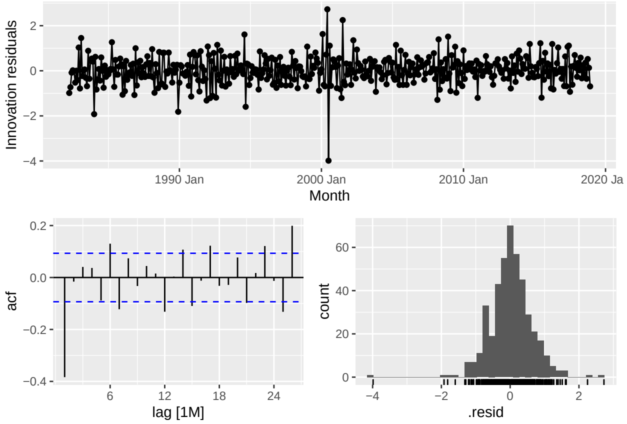


```
fit_2_1 %>%
    forecast(h = "2 years") %>%
    autoplot(data_2 %>%
        filter(Month > yearmonth("Jan 2014")), level = 95) +
    facet_wrap(vars(.model), ncol = 2, ) + guides(colour = "none",
    fill = "none", level = "none") + geom_label(aes(x = yearmonth("2015 Jan"),
    y = 2100, label = paste0("AICc = ", format(AICc))), data = glance(fit_2_1)) +
    labs(title = sprintf("%s - %s", data_2$Industry, data_2$State),
        y = "A$ billions")
```



2.2 code

```
# NOT APPENDIX because it's required for this part of the
# Exercise
fit_2_1 %>%
    select(`K = 5`) %>%
    gg_tsresiduals("innovation")
```



```
report(fit_2_1 %>%
    select(`K = 5`))
```

```
## Series: Turnover
## Model: LM w/ ARIMA(1,0,0)(1,0,0)[12] errors
## Transformation: box_cox(Turnover, lambda)
##
## Coefficients:
##         ar1    sar1  fourier(K = 5)C1_12  fourier(K = 5)S1_12
##       0.979  0.6449                0.665               -1.025
## s.e.  0.010  0.0392                0.215                0.216
##       fourier(K = 5)C2_12  fourier(K = 5)S2_12  fourier(K = 5)C3_12
##                     0.276               -1.185               -0.195
## s.e.                0.112                0.112                0.079
##       fourier(K = 5)S3_12  fourier(K = 5)C4_12  fourier(K = 5)S4_12
##                   -0.7583              -0.4721              -0.6856
## s.e.               0.0789               0.0645               0.0644
##       fourier(K = 5)C5_12  fourier(K = 5)S5_12  intercept
##                   -0.4432              -0.5416      31.31
## s.e.               0.0578               0.0577       3.45
##
## sigma^2 estimated as 0.3835:  log likelihood=-413
## AIC=855   AICc=856   BIC=912
```

```
augment(fit_2_1 %>%
    select(`K = 5`)) %>%
    features(.innov, ljung_box, lag = 24, dof = 13) %>%
    gt() %>%
    tab_header(title = md("**Box-Ljung for `K = 07` model**"),
        subtitle = md("with 13 DoF and 24 lags")) %>%
    opt_align_table_header(align = "center")
```

Box-Ljung for `K = 07` model

with 13 DoF and 24 lags

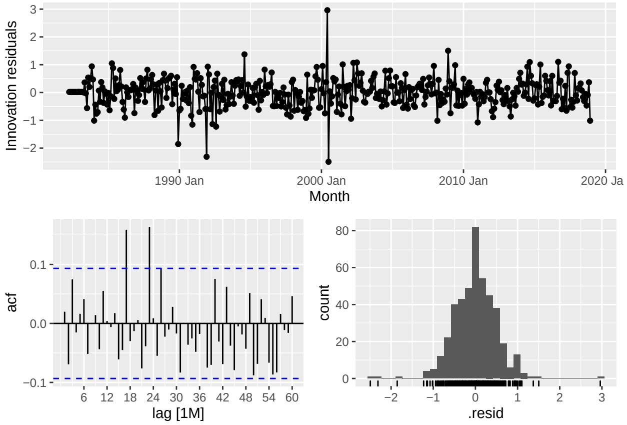
.model	lb_stat	lb_pvalue
K = 5	129	0

2.3 code

```
report(fit_2_3 %>%
    select(ARIMA))
```

```
## Series: Turnover
## Model: ARIMA(1,0,1)(1,1,2)[12] w/ drift
## Transformation: box_cox(Turnover, lambda)
##
## Coefficients:
##          ar1      ma1    sar1    sma1    sma2  constant
##       0.9338  -0.4327  -0.275  -0.302  -0.327    0.0552
## s.e.  0.0211   0.0531   0.472   0.457   0.300    0.0055
##
## sigma^2 estimated as 0.2704:  log likelihood=-329
## AIC=673   AICc=673   BIC=701
```

```
fit_2_3 %>%
    select(ARIMA) %>%
    gg_tsresiduals(lag = 12 * 5)
```



```
augment(fit_2_3 %>%
    select(ARIMA)) %>%
    features(.innov, ljung_box, lag = 12, dof = 6)
```

```
## # A tibble: 1 x 3
##   .model lb_stat lb_pvalue
##   <chr>    <dbl>     <dbl>
## 1 ARIMA     9.34     0.155
```

```
report(fit_2_3) %>%
    select(".model", "sigma2", "log_lik", "AIC", "AICc") %>%
    arrange(AICc) %>%
    gt() %>%
    tab_header(title = md("**Household good retailing - NSW: Model Comparison Metrics**")) %>%
    opt_align_table_header(align = "center")
```

Household good retailing - NSW: Model Comparison Metrics

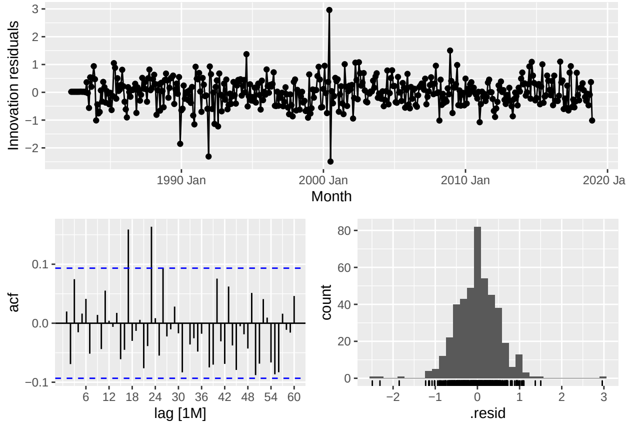
.model	sigma2	log_lik	AIC	AICc
Dynamic Regression	0.331	-324	666	667
ARIMA	0.270	-329	673	673
Fourier	0.384	-413	855	856
ETS	0.295	-1065	2165	2166

```
report(fit_2_3 %>%
    select(ARIMA))
```

```
## Series: Turnover
## Model: ARIMA(1,0,1)(1,1,2)[12] w/ drift
## Transformation: box_cox(Turnover, lambda)
##
```

```
## Coefficients:
##          ar1      ma1     sar1     sma1     sma2  constant
##       0.9338  -0.4327  -0.275  -0.302  -0.327    0.0552
## s.e.  0.0211   0.0531   0.472   0.457   0.300    0.0055
##
## sigma^2 estimated as 0.2704:  log likelihood=-329
## AIC=673   AICc=673   BIC=701
```

```
fit_2_3 %>%
    select(ARIMA) %>%
    gg_tsresiduals(lag = 12 * 5)
```



```
augment(fit_2_3 %>%
    select(ARIMA)) %>%
    features(.innov, ljung_box, lag = 12, dof = 6) %>%
    gt() %>%
    tab_header(title = md("**Box-Ljung for `K = 07` model**"),
        subtitle = md("with 13 DoF and 24 lags")) %>%
    opt_align_table_header(align = "center")
```

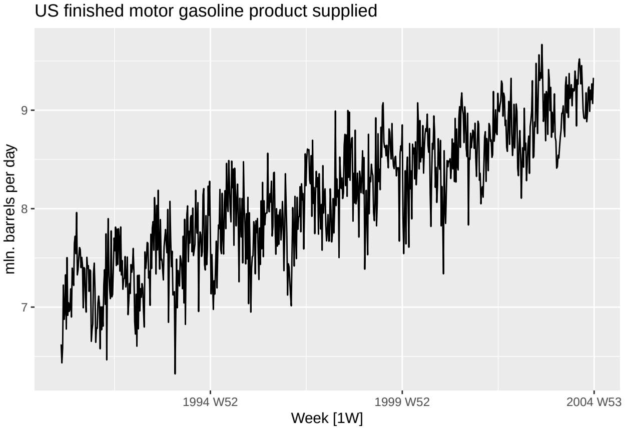
Box-Ljung for `K = 07` model

with 13 DoF and 24 lags

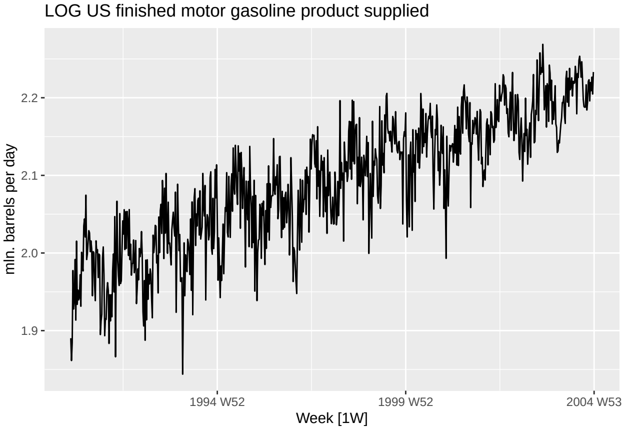
.model	lb_stat	lb_pvalue
ARIMA	9.34	0.155

3.1 code

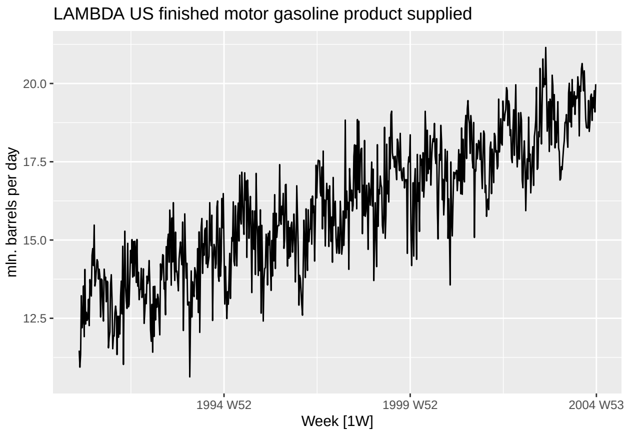
```
usgas <- us_gasoline %>%
    filter(Week < yearweek("2005"))
usgas %>%
    autoplot(Barrels) + labs(title = "US finished motor gasoline product supplied",
    y = "mln. barrels per day")
```



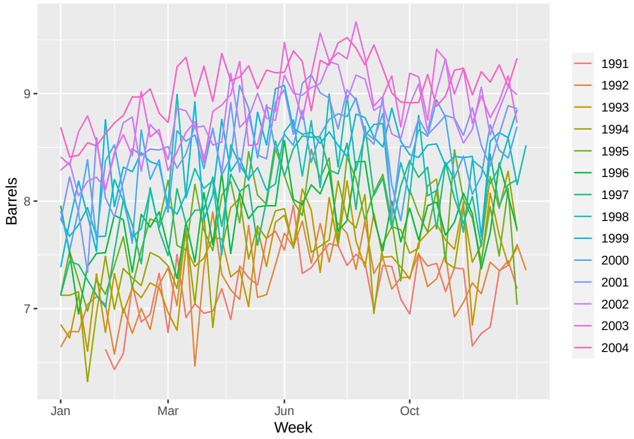
```
usgas %>%
    autoplot(log(Barrels)) + labs(title = "LOG US finished motor gasoline product supplied",
    y = "mln. barrels per day")
```



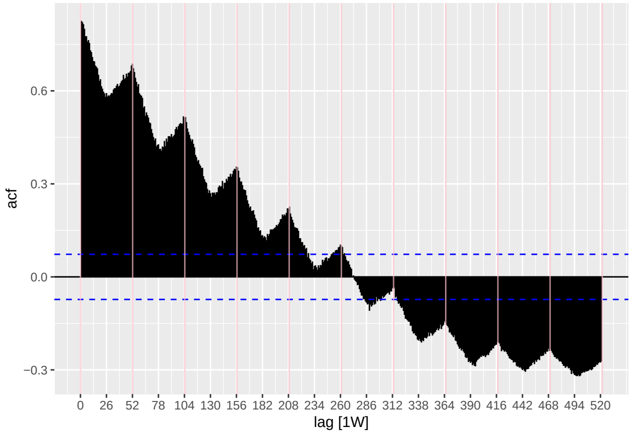
```
lambda <- usgas %>%
    features(Barrels, features = guerrero) %>%
    pull(lambda_guerrero)

usgas %>%
    autoplot(BoxCox(Barrels, lambda)) + labs(title = "LAMBDA US finished motor gasoline produc
    y = "mln. barrels per day")
```



```
usgas %>%
    gg_season(Barrels)
```

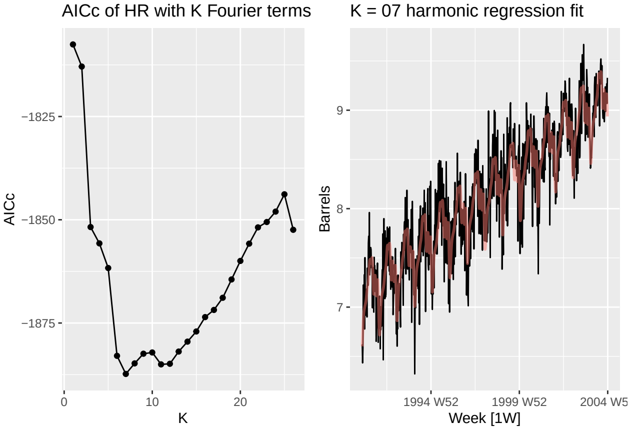


```
usgas %>%
    select(Barrels) %>%
    ACF(lag_max = 52.18 * 10) %>%
    autoplot() + geom_vline(xintercept = seq(0, 52.18 * 10, 52.18),
    colour = "pink", alpha = 0.7)
```

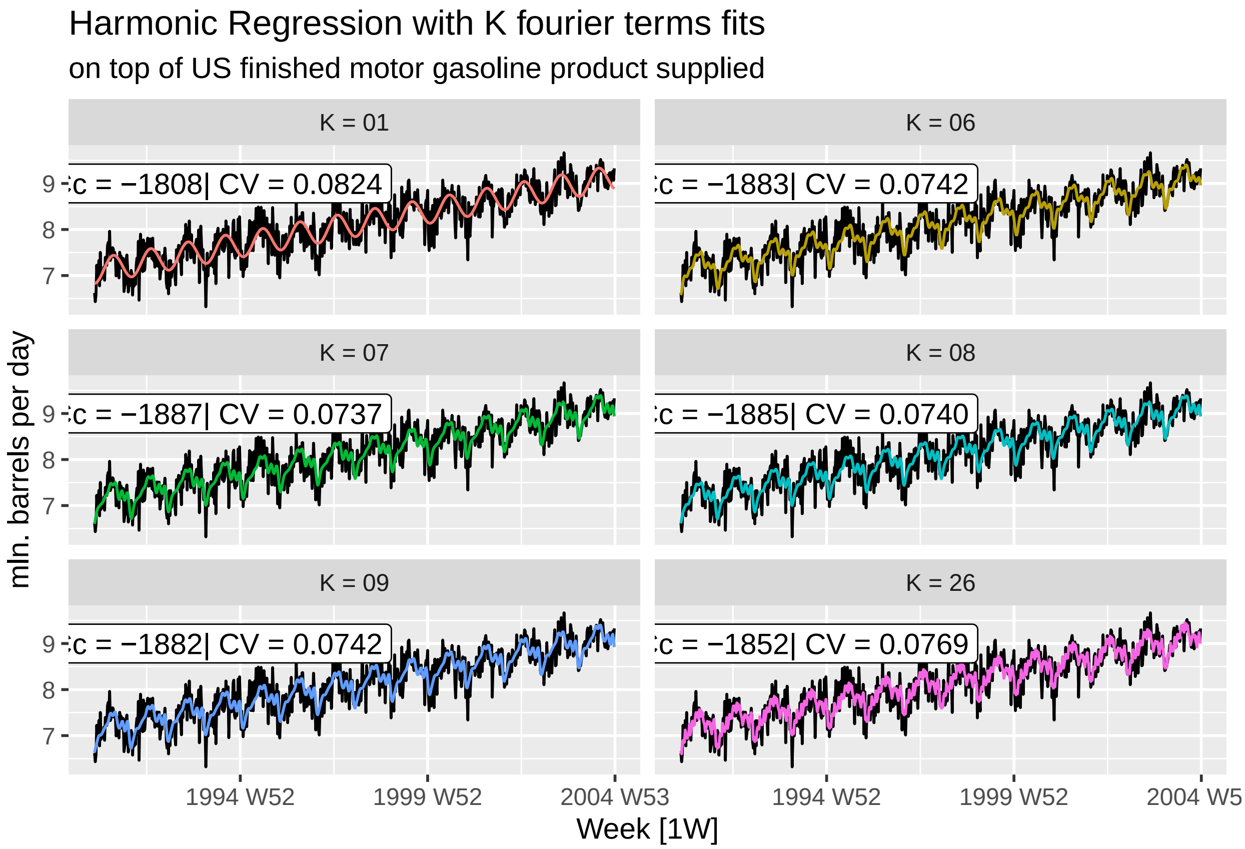


```
# Careful this takes forever
fit <- model(usgas, `K = 01` = TSLM(Barrels ~ trend() + fourier(K = 1)),
    `K = 02` = TSLM(Barrels ~ trend() + fourier(K = 2)), `K = 03` = TSLM(Barrels ~
        trend() + fourier(K = 3)), `K = 04` = TSLM(Barrels ~
        trend() + fourier(K = 4)), `K = 05` = TSLM(Barrels ~
        trend() + fourier(K = 5)), `K = 06` = TSLM(Barrels ~
        trend() + fourier(K = 6)), `K = 07` = TSLM(Barrels ~
        trend() + fourier(K = 7)), `K = 08` = TSLM(Barrels ~
        trend() + fourier(K = 8)), `K = 09` = TSLM(Barrels ~
        trend() + fourier(K = 9)), `K = 10` = TSLM(Barrels ~
        trend() + fourier(K = 10)), `K = 11` = TSLM(Barrels ~
        trend() + fourier(K = 11)), `K = 12` = TSLM(Barrels ~
        trend() + fourier(K = 12)), `K = 13` = TSLM(Barrels ~
        trend() + fourier(K = 13)), `K = 14` = TSLM(Barrels ~
        trend() + fourier(K = 14)), `K = 15` = TSLM(Barrels ~
        trend() + fourier(K = 15)), `K = 16` = TSLM(Barrels ~
        trend() + fourier(K = 16)), `K = 17` = TSLM(Barrels ~
        trend() + fourier(K = 17)), `K = 18` = TSLM(Barrels ~
        trend() + fourier(K = 18)), `K = 19` = TSLM(Barrels ~
        trend() + fourier(K = 19)), `K = 20` = TSLM(Barrels ~
        trend() + fourier(K = 20)), `K = 21` = TSLM(Barrels ~
        trend() + fourier(K = 21)), `K = 22` = TSLM(Barrels ~
        trend() + fourier(K = 22)), `K = 23` = TSLM(Barrels ~
        trend() + fourier(K = 23)), `K = 24` = TSLM(Barrels ~
        trend() + fourier(K = 24)), `K = 25` = TSLM(Barrels ~
        trend() + fourier(K = 25)), `K = 26` = TSLM(Barrels ~
        trend() + fourier(K = 26)))
```

```
# fit %>% glance() %>% # if you want to plot the accuracies table, this one works
#   select(-c(ar_roots,ma_roots)) %>%
#    gt() %>%
#   tab_header(
#     title = md("**HR w/ Fouriers comparison**")
#   ) %>% opt_align_table_header(align = "center")

g <- fit %>% glance() %>% select(AICc, CV)
g <- data.frame(AICc = g$AICc, CV = g$CV, K = seq(1,26,1))

p1 <- ggplot(g, aes(x=K, y=AICc, group =1)) + geom_path()+
 geom_point() + labs(title = "AICc of HR with K Fourier terms")

p2 <- usgas %>% autoplot(Barrels) + #plot
 geom_line(data = fitted(fit %>% select("K = 07")),
      aes(y = .fitted, colour = .model), size = 1, alpha = 0.5) +
 labs(title = "K = 07 harmonic regression fit") +
 theme(legend.position = "none")

grid.arrange(p1,p2, ncol=2)
```



```r
poi = c('K = 01', 'K = 06', 'K = 07', 'K = 08', 'K = 09', 'K = 26')

usgas %>% autoplot(Barrels) + #plot
  facet_wrap(vars(.model), ncol = 2) +
  geom_line(data = fitted(fit %>% select(poi)),
            aes(y = .fitted, colour = .model)) +
  guides(colour = "none", fill = "none", level = "none") +
  labs( title = "Harmonic Regression with K fourier terms fits",
        subtitle = "on top of US finished motor gasoline product supplied",
        y = "mln. barrels per day") +
  geom_label(
    aes(x = yearweek("1993 W52"), y = 9,
        label = paste0("AICc = ", format(AICc), "| CV = ", format(CV))),
    data = glance(fit %>% select(poi))
  )
```



```
# report(fit %>% select('K = 07'))
gg_tsresiduals(fit %>%
    select("K = 07"))
```

```
# fit %>% select('K = 07') %>% report()

augment(fit %>%
    select("K = 07")) %>%
    features(.innov, ljung_box, dof = 16, lag = 53) %>%
    gt() %>%
    tab_header(title = md("**Box-Ljung for `K = 07` model**"),
        subtitle = md("with 16 DoF and 53 lags")) %>%
    opt_align_table_header(align = "center")
```

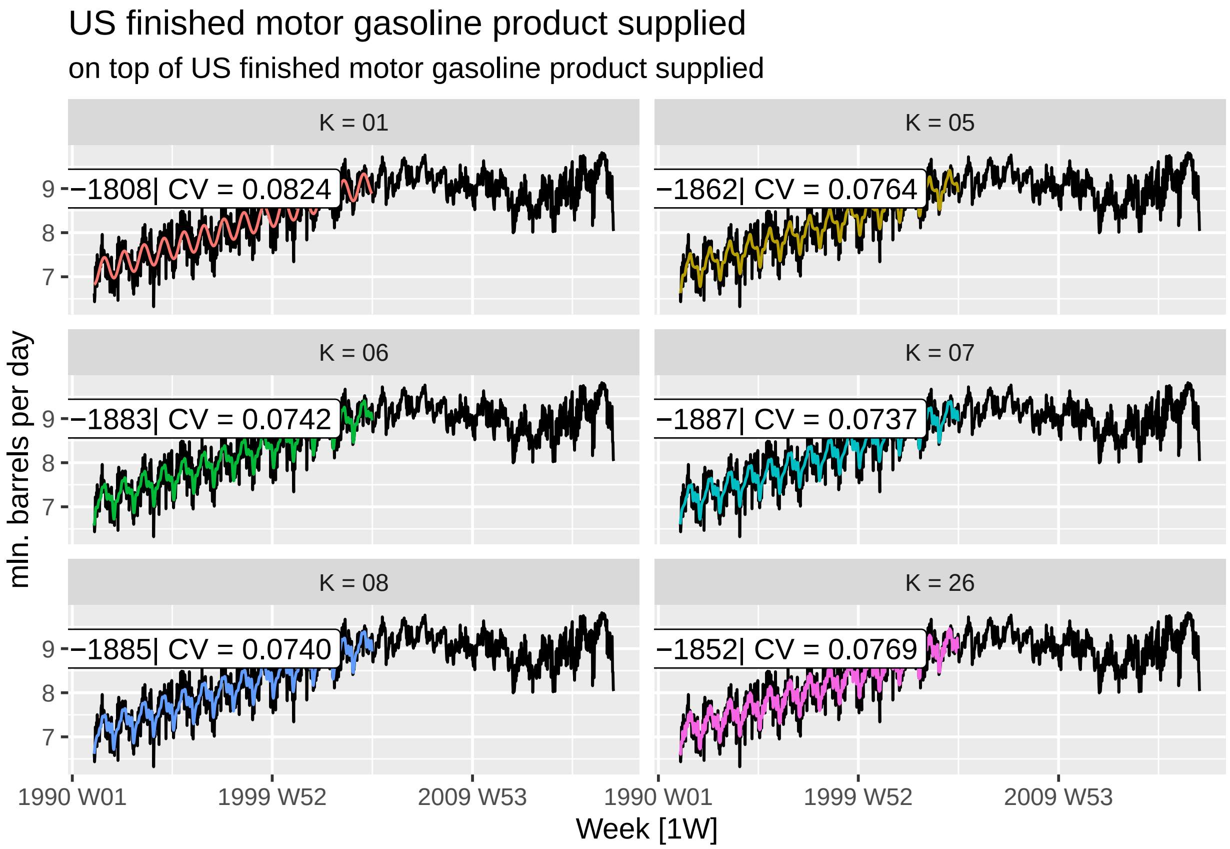
3.2 code

```
# fit11 <- model(usgas, `K = 08` = ARIMA(Barrels ~
# fourier(K=08) + PDQ(0:1,0,0:1)), )

fc <- fit %>%
    select("K = 07") %>%
    forecast(h = "1 year")
fc %>%
    autoplot(size = 1) + autolayer((us_gasoline %>%
    filter(Week >= yearweek("2005"), Week < yearweek("2006")))) +
    labs(title = "2005 US finished motor gasoline product supplied + forecast (K = 07)",
        y = "mln. barrels per day")

accuracy(fc, us_gasoline) %>%
    select(c(".model", "RMSE", "MAE", "MPE", "MAPE")) %>%
    gt() %>%
    tab_header(title = md("**HR with 7 Foruier terms accuracy**"),
        subtitle = md("as fitted to the pre 2005 US gasoline dataset, prediction on 2005")) %>%
    opt_align_table_header(align = "center")
```

```
elbows = c(yearweek("2007"), yearweek("2013"))
fit <- model(us_gasoline, `K = 01` = TSLM(Barrels ~ trend(knots = elbows) +
    fourier(K = 1)), `K = 02` = TSLM(Barrels ~ trend(knots = elbows) +
    fourier(K = 2)), `K = 03` = TSLM(Barrels ~ trend(knots = elbows) +
    fourier(K = 3)), `K = 04` = TSLM(Barrels ~ trend(knots = elbows) +
    fourier(K = 4)), `K = 05` = TSLM(Barrels ~ trend(knots = elbows) +
    fourier(K = 5)), `K = 06` = TSLM(Barrels ~ trend(knots = elbows) +
    fourier(K = 6)), `K = 07` = TSLM(Barrels ~ trend(knots = elbows) +
    fourier(K = 7)), `K = 08` = TSLM(Barrels ~ trend(knots = elbows) +
    fourier(K = 8)), `K = 09` = TSLM(Barrels ~ trend(knots = elbows) +
    fourier(K = 9)), `K = 10` = TSLM(Barrels ~ trend(knots = elbows) +
    fourier(K = 10)), `K = 11` = TSLM(Barrels ~ trend(knots = elbows) +
    fourier(K = 11)), `K = 12` = TSLM(Barrels ~ trend(knots = elbows) +
    fourier(K = 12)), `K = 13` = TSLM(Barrels ~ trend(knots = elbows) +
    fourier(K = 13)), `K = 14` = TSLM(Barrels ~ trend(knots = elbows) +
    fourier(K = 14)), `K = 15` = TSLM(Barrels ~ trend(knots = elbows) +
    fourier(K = 15)), `K = 16` = TSLM(Barrels ~ trend(knots = elbows) +
    fourier(K = 16)), `K = 17` = TSLM(Barrels ~ trend(knots = elbows) +
    fourier(K = 17)), `K = 18` = TSLM(Barrels ~ trend(knots = elbows) +
    fourier(K = 18)), `K = 19` = TSLM(Barrels ~ trend(knots = elbows) +
    fourier(K = 19)), `K = 20` = TSLM(Barrels ~ trend(knots = elbows) +
    fourier(K = 20)), `K = 21` = TSLM(Barrels ~ trend(knots = elbows) +
    fourier(K = 21)), `K = 22` = TSLM(Barrels ~ trend(knots = elbows) +
    fourier(K = 22)), `K = 23` = TSLM(Barrels ~ trend(knots = elbows) +
    fourier(K = 23)), `K = 24` = TSLM(Barrels ~ trend(knots = elbows) +
    fourier(K = 24)), `K = 25` = TSLM(Barrels ~ trend(knots = elbows) +
    fourier(K = 25)), `K = 26` = TSLM(Barrels ~ trend(knots = elbows) +
    fourier(K = 26)), )
```

```
g <- fit %>%
    glance() %>%
    select(AICc, CV)
g = data.frame(AICc = g$AICc, CV = g$CV, K = seq(1, 26, 1))
ggplot(g, aes(x = K, y = AICc, group = 1)) + geom_path() + geom_point() +
    labs(title = "AICc of HR with piecewise trend and K Fourier terms")
```

```
poi = c('K = 01', 'K = 05', 'K = 06', 'K = 07', 'K = 08', 'K = 26')

us_gasoline %>% autoplot(Barrels) + #plot
  facet_wrap(vars(.model), ncol = 2) +
  geom_line(data = fitted(fit %>% select(poi)),
            aes(y = .fitted, colour = .model)) +
  guides(colour = "none", fill = "none", level = "none") +
  labs( title = "US finished motor gasoline product supplied",
        subtitle = "on top of US finished motor gasoline product supplied",
        y = "mln. barrels per day") +
  geom_label(
    aes(x = yearweek("1993 W52"), y = 9,
        label = paste0("AICc = ", format(AICc), "| CV = ", format(CV))),
    data = glance(fit %>% select(poi))
  )
```



```
gg_tsresiduals(fit %>%
    select("K = 06"))
```

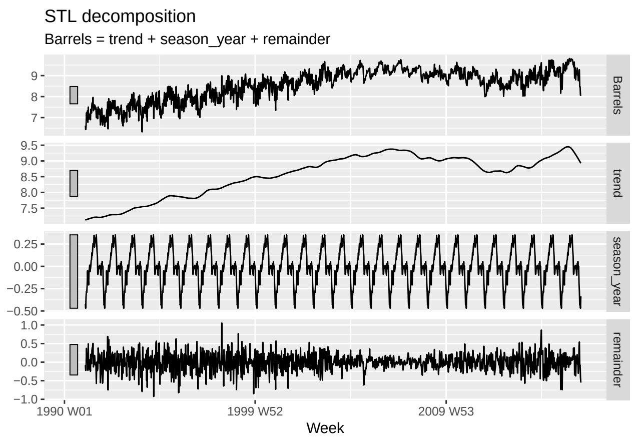
3.3 code

```
lambda <- us_gasoline %>% #lambda is two
  features(Barrels, features = guerrero) %>%
  pull(lambda_guerrero)

#us_gasoline %>% mutate(Barrels_t = BoxCox(Barrels, lambda)) # this did not work

fit <- model(us_gasoline,
             `ARIMA` = ARIMA( Barrels ~ trend (knots=elbows) + fourier(K=6) + PDQ(0:1,0,0:1))
             )
usgaspost2005 <- us_gasoline %>% filter(Week >= yearweek('2005 W10'))
fit_short <- model(usgaspost2005,
             `ARIMA post 2005` = ARIMA( Barrels ~ trend (knots=elbows) + fourier(K=6) + PDQ(0:
                  )
```

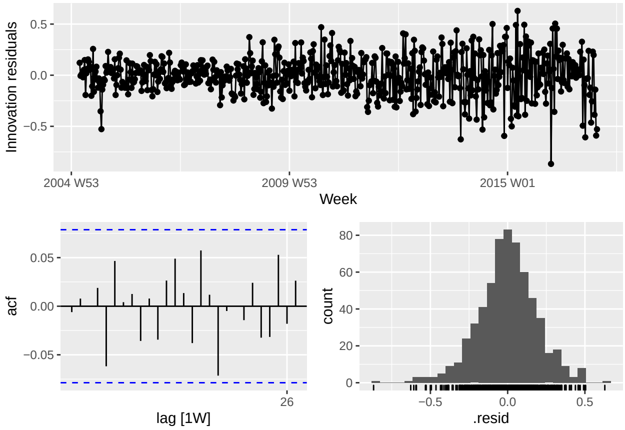
```
fit %>% glance() %>%
 select(-c(ar_roots,ma_roots)) %>%
   gt() %>%
  tab_header(
    title = md("**ARIMA HR w/ 6 Fourier terms and knots**")
  ) %>% opt_align_table_header(align = "center")

# fit %>% report()
# fit_short %>% report()

us_gasoline %>% autoplot(Barrels) + #plot
  geom_line(data = fitted(fit),
            aes(y = .fitted, colour = .model), size = 1, alpha = 0.5) +
  geom_line(data = fitted(fit_short),
            aes(y = .fitted, colour = .model), size = 1, alpha = 0.5)
```

```
rbind(augment(fit %>%
    select("ARIMA")) %>%
    features(.innov, ljung_box, dof = 21, lag = 53), augment(fit_short %>%
    select("ARIMA post 2005")) %>%
    features(.innov, ljung_box, dof = 21, lag = 53)) %>%
    gt() %>%
    tab_header(title = md("**Box-Ljung two ARIMA + HR models**"),
        subtitle = md("with 21 DoF and 53 lags")) %>%
    opt_align_table_header(align = "center")
```

```
us_gasoline %>%
  model(
    STL(Barrels ~ trend(window = 53) + #smooth out the whole year
      season(window = "periodic"),
        robust = TRUE)) %>%
  components() %>%
  autoplot()
```



```
gg_tsresiduals(fit_short %>% select('ARIMA post 2005'))
```



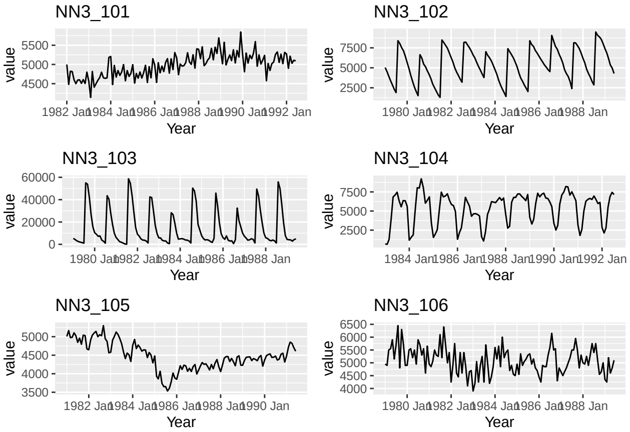
4.1 code

```
# the readxl method is passed in the ts with a range of
# rows referring to the location of the train/test split as
# well as the start argument which sets the year and month
# of the split.

data_path <- "/Users/emielsteegh/UvA/AFiCS/A3/NN3_REDUCED_DATASET_WITH_TEST_DATA.xls"

NN3_101_train <- ts(readxl::read_excel(data_path, range = "B18:B144"),
    start = c(1982, 1), frequency = 12) %>%
    as_tsibble()

NN3_102_train <- ts(readxl::read_excel(data_path, range = "C18:C144"),
    start = c(1979, 1), frequency = 12) %>%
    as_tsibble()

NN3_103_train <- ts(readxl::read_excel(data_path, range = "D18:D144"),
    start = c(1979, 1), frequency = 12) %>%
    as_tsibble()

NN3_104_train <- ts(readxl::read_excel(data_path, range = "E18:E133"),
    start = c(1983, 1), frequency = 12) %>%
    as_tsibble()

NN3_105_train <- ts(readxl::read_excel(data_path, range = "F18:F144"),
    start = c(1981, 1), frequency = 12) %>%
    as_tsibble()

NN3_106_train <- ts(readxl::read_excel(data_path, range = "G18:G144"),
    start = c(1979, 1), frequency = 12) %>%
    as_tsibble()

NN3_107_train <- ts(readxl::read_excel(data_path, range = "H18:H144"),
    start = c(1982, 1), frequency = 12) %>%
    as_tsibble()

NN3_108_train <- ts(readxl::read_excel(data_path, range = "I18:I133"),
    start = c(1983, 1), frequency = 12) %>%
    as_tsibble()

NN3_109_train <- ts(readxl::read_excel(data_path, range = "J18:J141"),
    start = c(1980, 1), frequency = 12) %>%
    as_tsibble()

NN3_110_train <- ts(readxl::read_excel(data_path, range = "K18:K144"),
    start = c(1978, 1), frequency = 12) %>%
    as_tsibble()

NN3_111_train <- ts(readxl::read_excel(data_path, range = "L18:L144"),
    start = c(1978, 1), frequency = 12) %>%
    as_tsibble()

NN3_101_test <- ts(readxl::read_excel(data_path, range = "B144:B162"),
    start = c(1992, 7), frequency = 12) %>%
    as_tsibble()

NN3_102_test <- ts(readxl::read_excel(data_path, range = "C144:C162"),
    start = c(1989, 7), frequency = 12) %>%
    as_tsibble()

NN3_103_test <- ts(readxl::read_excel(data_path, range = "D144:D162"),
    start = c(1989, 7), frequency = 12) %>%
    as_tsibble()

NN3_104_test <- ts(readxl::read_excel(data_path, range = "E132:E151"),
```

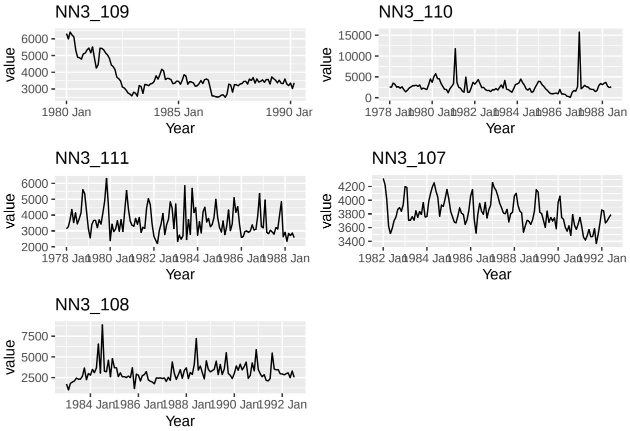
```r
    start = c(1992, 8), frequency = 12) %>%
    as_tsibble()

NN3_105_test <- ts(readxl::read_excel(data_path, range = "F144:F162"),
    start = c(1991, 7), frequency = 12) %>%
    as_tsibble()

NN3_106_test <- ts(readxl::read_excel(data_path, range = "G144:G162"),
    start = c(1989, 7), frequency = 12) %>%
    as_tsibble()

NN3_107_test <- ts(readxl::read_excel(data_path, range = "H144:H162"),
    start = c(1992, 7), frequency = 12) %>%
    as_tsibble()

NN3_108_test <- ts(readxl::read_excel(data_path, range = "I133:I152"),
    start = c(1992, 8), frequency = 12) %>%
    as_tsibble()

NN3_109_test <- ts(readxl::read_excel(data_path, range = "J141:J159"),
    start = c(1990, 4), frequency = 12) %>%
    as_tsibble()

NN3_110_test <- ts(readxl::read_excel(data_path, range = "K144:K162"),
    start = c(1988, 7), frequency = 12) %>%
    as_tsibble()

NN3_111_test <- ts(readxl::read_excel(data_path, range = "L144:L162"),
    start = c(1988, 7), frequency = 12) %>%
    as_tsibble()
```

```r
plot1 <- NN3_101_train %>%
    autoplot() + labs(title = "NN3_101", x = "Year")
plot2 <- NN3_102_train %>%
    autoplot() + labs(title = "NN3_102", x = "Year")
plot3 <- NN3_103_train %>%
    autoplot() + labs(title = "NN3_103", x = "Year")
plot4 <- NN3_104_train %>%
    autoplot() + labs(title = "NN3_104", x = "Year")
plot5 <- NN3_105_train %>%
    autoplot() + labs(title = "NN3_105", x = "Year")
plot6 <- NN3_106_train %>%
    autoplot() + labs(title = "NN3_106", x = "Year")
plot7 <- NN3_107_train %>%
    autoplot() + labs(title = "NN3_107", x = "Year")
plot8 <- NN3_108_train %>%
    autoplot() + labs(title = "NN3_108", x = "Year")
```

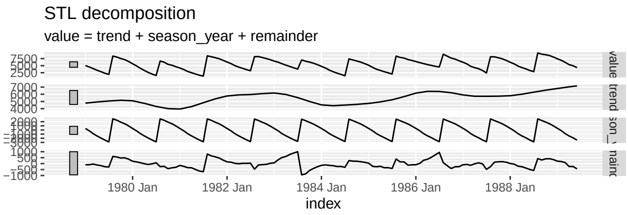
```
plot9 <- NN3_109_train %>%
    autoplot() + labs(title = "NN3_109", x = "Year")
plot10 <- NN3_110_train %>%
    autoplot() + labs(title = "NN3_110", x = "Year")
plot11 <- NN3_111_train %>%
    autoplot() + labs(title = "NN3_111", x = "Year")

grid.arrange(plot1, plot2, plot3, plot4, plot5, plot6, ncol = 2)
```

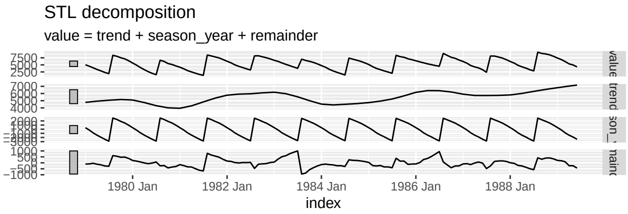


```
grid.arrange(plot9, plot10, plot11, plot7, plot8, ncol = 2)
```

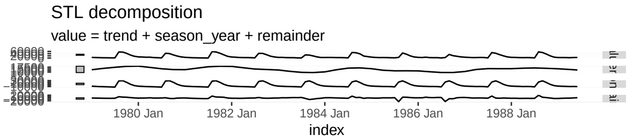


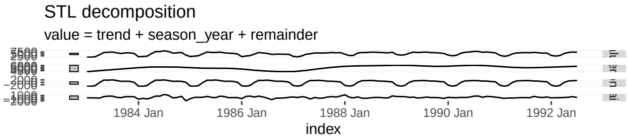
```
NN3_101_decomp <- NN3_101_train %>%
    model(STL(value))
NN3_102_decomp <- NN3_102_train %>%
    model(STL(value))
NN3_103_decomp <- NN3_103_train %>%
    model(STL(value))
NN3_104_decomp <- NN3_104_train %>%
    model(STL(value))
NN3_106_decomp <- NN3_106_train %>%
    model(STL(value))

grid.arrange(autoplot(components(NN3_102_decomp)), autoplot(components(NN3_102_decomp)),
    ncol = 1)
```

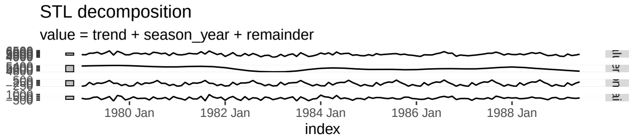




```
grid.arrange(autoplot(components(NN3_103_decomp)), autoplot(components(NN3_104_decomp)),
    autoplot(components(NN3_106_decomp)), ncol = 1)
```

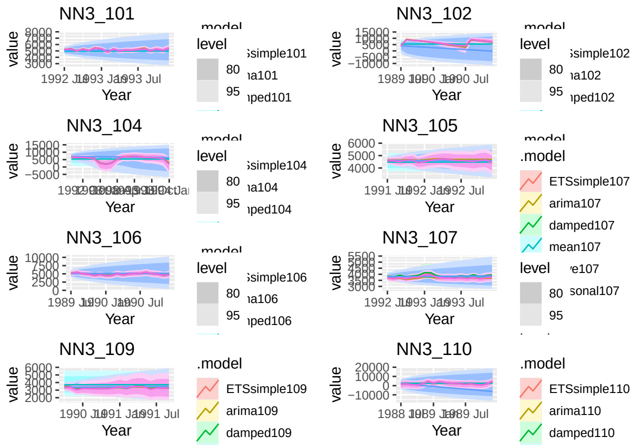






```
# 108, 104 and 111 were not plotted due to important scale
# transformations
fcplot1 <- fc_NN3_101 %>%
    autoplot() + labs(title = "NN3_101", x = "Year")
fcplot2 <- fc_NN3_102 %>%
    autoplot() + labs(title = "NN3_102", x = "Year")
fcplot4 <- fc_NN3_104 %>%
    autoplot() + labs(title = "NN3_104", x = "Year")
fcplot5 <- fc_NN3_105 %>%
    autoplot() + labs(title = "NN3_105", x = "Year")
fcplot6 <- fc_NN3_106 %>%
    autoplot() + labs(title = "NN3_106", x = "Year")
fcplot7 <- fc_NN3_107 %>%
    autoplot() + labs(title = "NN3_107", x = "Year")
fcplot9 <- fc_NN3_109 %>%
    autoplot() + labs(title = "NN3_109", x = "Year")
fcplot10 <- fc_NN3_110 %>%
    autoplot() + labs(title = "NN3_110", x = "Year")
```

```
grid.arrange(fcplot1, fcplot2, fcplot4, fcplot5, fcplot6, fcplot7,
    fcplot9, fcplot10, ncol = 2)
```



```
# the readxl method is passed in the ts with a range of
# rows referring to the location of the train/test split as
# well as the start argument which sets the year and month
# of the split.

data_path <- "/Users/emielsteegh/UvA/AFiCS/A3/NN3_REDUCED_DATASET_WITH_TEST_DATA.xls"

NN3_101_train <- ts(readxl::read_excel(data_path, range = "B18:B144"),
    start = c(1982, 1), frequency = 12) %>%
    as_tsibble()

NN3_102_train <- ts(readxl::read_excel(data_path, range = "C18:C144"),
    start = c(1979, 1), frequency = 12) %>%
    as_tsibble()

NN3_103_train <- ts(readxl::read_excel(data_path, range = "D18:D144"),
    start = c(1979, 1), frequency = 12) %>%
    as_tsibble()

NN3_104_train <- ts(readxl::read_excel(data_path, range = "E18:E133"),
    start = c(1983, 1), frequency = 12) %>%
    as_tsibble()

NN3_105_train <- ts(readxl::read_excel(data_path, range = "F18:F144"),
    start = c(1981, 1), frequency = 12) %>%
    as_tsibble()

NN3_106_train <- ts(readxl::read_excel(data_path, range = "G18:G144"),
    start = c(1979, 1), frequency = 12) %>%
    as_tsibble()
```

```
NN3_107_train <- ts(readxl::read_excel(data_path, range = "H18:H144"),
    start = c(1982, 1), frequency = 12) %>%
    as_tsibble()

NN3_108_train <- ts(readxl::read_excel(data_path, range = "I18:I133"),
    start = c(1983, 1), frequency = 12) %>%
    as_tsibble()

NN3_109_train <- ts(readxl::read_excel(data_path, range = "J18:J141"),
    start = c(1980, 1), frequency = 12) %>%
    as_tsibble()

NN3_110_train <- ts(readxl::read_excel(data_path, range = "K18:K144"),
    start = c(1978, 1), frequency = 12) %>%
    as_tsibble()

NN3_111_train <- ts(readxl::read_excel(data_path, range = "L18:L144"),
    start = c(1978, 1), frequency = 12) %>%
    as_tsibble()

NN3_101_test <- ts(readxl::read_excel(data_path, range = "B144:B162"),
    start = c(1992, 7), frequency = 12) %>%
    as_tsibble()

NN3_102_test <- ts(readxl::read_excel(data_path, range = "C144:C162"),
    start = c(1989, 7), frequency = 12) %>%
    as_tsibble()

NN3_103_test <- ts(readxl::read_excel(data_path, range = "D144:D162"),
    start = c(1989, 7), frequency = 12) %>%
    as_tsibble()

NN3_104_test <- ts(readxl::read_excel(data_path, range = "E132:E151"),
    start = c(1992, 8), frequency = 12) %>%
    as_tsibble()

NN3_105_test <- ts(readxl::read_excel(data_path, range = "F144:F162"),
    start = c(1991, 7), frequency = 12) %>%
    as_tsibble()

NN3_106_test <- ts(readxl::read_excel(data_path, range = "G144:G162"),
    start = c(1989, 7), frequency = 12) %>%
    as_tsibble()

NN3_107_test <- ts(readxl::read_excel(data_path, range = "H144:H162"),
    start = c(1992, 7), frequency = 12) %>%
    as_tsibble()
```

```r
NN3_108_test <- ts(readxl::read_excel(data_path, range = "I133:I152"),
    start = c(1992, 8), frequency = 12) %>%
    as_tsibble()

NN3_109_test <- ts(readxl::read_excel(data_path, range = "J141:J159"),
    start = c(1990, 4), frequency = 12) %>%
    as_tsibble()

NN3_110_test <- ts(readxl::read_excel(data_path, range = "K144:K162"),
    start = c(1988, 7), frequency = 12) %>%
    as_tsibble()

NN3_111_test <- ts(readxl::read_excel(data_path, range = "L144:L162"),
    start = c(1988, 7), frequency = 12) %>%
    as_tsibble()
```

4.2 code

```r
# list created below to iterate through all tsibbles and
# retrieve lambda value
ts_models <- list(NN3_101_train, NN3_102_train, NN3_103_train,
    NN3_104_train, NN3_105_train, NN3_106_train, NN3_107_train,
    NN3_108_train, NN3_109_train, NN3_110_train, NN3_111_train)
guerr <- c()  # Empty vector to retrieve lambda values
nr <- 0
for (i in ts_models) {
    ## loop to easily retrieve lambda values
    dummy <- i %>%
        features(value, features = guerrero) %>%
        pull(lambda_guerrero)
    guerr <- append(guerr, dummy)

    print(glue("TimeSeries no. {nr+1} has a lambda value of {round(dummy, 2)}"))

    nr <- nr + 1
}
```

```
## TimeSeries no. 1 has a lambda value of 0.71
## TimeSeries no. 2 has a lambda value of 1.6
## TimeSeries no. 3 has a lambda value of -0.6
## TimeSeries no. 4 has a lambda value of 1.45
## TimeSeries no. 5 has a lambda value of 1.21
## TimeSeries no. 6 has a lambda value of 0.92
## TimeSeries no. 7 has a lambda value of -0.9
## TimeSeries no. 8 has a lambda value of -0.77
## TimeSeries no. 9 has a lambda value of 0.39
## TimeSeries no. 10 has a lambda value of 1.3
## TimeSeries no. 11 has a lambda value of -0.9
```

```
# In the below time series, a lambda value of -0.5 was used
# whenever the value dips below -0.5 in order to avoid
# forecasting NaNs.

fc_NN3_101 <- model(NN3_101_train, mean101 = MEAN(box_cox(value,
    lambda = guerr[1])), naive101 = NAIVE(box_cox(value, lambda = guerr[1])),
    ETSsimple101 = ETS(box_cox(value, lambda = guerr[1])), seasonal101 = ETS(box_cox(value,
        lambda = guerr[1]) ~ error("A") + trend("N") + season("A")),
    damped101 = ETS(box_cox(value, lambda = guerr[1]) ~ error("A") +
        trend("Ad") + season("A")), arima101 = ARIMA(box_cox(value,
        guerr[1]), stepwise = TRUE)) %>%
    forecast(h = "18 months")

fc_NN3_102 <- model(NN3_102_train, mean102 = MEAN(box_cox(value,
    lambda = guerr[2])), naive102 = NAIVE(box_cox(value, lambda = guerr[2])),
    ETSsimple102 = ETS(box_cox(value, lambda = guerr[2])), seasonal102 = ETS(box_cox(value,
        lambda = guerr[2]) ~ error("A") + trend("N") + season("A")),
    damped102 = ETS(box_cox(value, lambda = guerr[2]) ~ error("A") +
        trend("Ad") + season("A")), arima102 = ARIMA(box_cox(value,
        guerr[2]), stepwise = TRUE)) %>%
    forecast(h = "18 months")

fc_NN3_103 <- model(NN3_103_train, mean103 = MEAN(box_cox(value,
    lambda = -0.5)), naive103 = NAIVE(box_cox(value, lambda = -0.5)),
    ETSsimple103 = ETS(box_cox(value, lambda = -0.5)), seasonal103 = ETS(box_cox(value,
        lambda = -0.5) ~ error("A") + trend("N") + season("A")),
    damped103 = ETS(box_cox(value, lambda = -0.5) ~ error("A") +
        trend("Ad") + season("A")), arima103 = ARIMA(box_cox(value,
        -0.5), stepwise = TRUE)) %>%
    forecast(h = "18 months")

fc_NN3_104 <- model(NN3_104_train, mean104 = MEAN(box_cox(value,
    lambda = guerr[4])), naive104 = NAIVE(box_cox(value, lambda = guerr[4])),
    ETSsimple104 = ETS(box_cox(value, lambda = guerr[4])), seasonal104 = ETS(box_cox(value,
        lambda = guerr[4]) ~ error("A") + trend("N") + season("A")),
    damped104 = ETS(box_cox(value, lambda = guerr[4]) ~ error("A") +
        trend("Ad") + season("A")), arima104 = ARIMA(box_cox(value,
        guerr[4]), stepwise = TRUE)) %>%
    forecast(h = "18 months")

fc_NN3_105 <- model(NN3_105_train, mean105 = MEAN(box_cox(value,
    lambda = guerr[5])), naive105 = NAIVE(box_cox(value, lambda = guerr[5])),
    ETSsimple105 = ETS(box_cox(value, lambda = guerr[5])), seasonal105 = ETS(box_cox(value,
        lambda = guerr[5]) ~ error("A") + trend("N") + season("A")),
    damped105 = ETS(box_cox(value, lambda = guerr[5]) ~ error("A") +
        trend("Ad") + season("A")), arima105 = ARIMA(box_cox(value,
        guerr[5]), stepwise = TRUE)) %>%
    forecast(h = "18 months")

fc_NN3_106 <- model(NN3_106_train, mean106 = MEAN(box_cox(value,
    lambda = guerr[6])), naive106 = NAIVE(box_cox(value, lambda = guerr[6])),
    ETSsimple106 = ETS(box_cox(value, lambda = guerr[6])), seasonal106 = ETS(box_cox(value,
        lambda = guerr[6]) ~ error("A") + trend("N") + season("A")),
    damped106 = ETS(box_cox(value, lambda = guerr[6]) ~ error("A") +
        trend("Ad") + season("A")), arima106 = ARIMA(box_cox(value,
        guerr[6]), stepwise = TRUE)) %>%
    forecast(h = "18 months")

fc_NN3_107 <- model(NN3_107_train, mean107 = MEAN(box_cox(value,
    lambda = -0.5)), naive107 = NAIVE(box_cox(value, lambda = -0.5)),
    ETSsimple107 = ETS(box_cox(value, lambda = -0.5)), seasonal107 = ETS(box_cox(value,
        lambda = -0.5) ~ error("A") + trend("N") + season("A")),
    damped107 = ETS(box_cox(value, lambda = -0.5) ~ error("A") +
        trend("Ad") + season("A")), arima107 = ARIMA(box_cox(value,
        -0.5), stepwise = TRUE)) %>%
    forecast(h = "18 months")

fc_NN3_108 <- model(NN3_108_train, mean108 = MEAN(box_cox(value,
    lambda = -0.5)), naive108 = NAIVE(box_cox(value, lambda = -0.5)),
    ETSsimple108 = ETS(box_cox(value, lambda = -0.5)), seasonal108 = ETS(box_cox(value,
        lambda = -0.5) ~ error("A") + trend("N") + season("A")),
    damped108 = ETS(box_cox(value, lambda = -0.5) ~ error("A") +
        trend("Ad") + season("A")), arima108 = ARIMA(box_cox(value,
        -0.5), stepwise = TRUE)) %>%
    forecast(h = "18 months")

fc_NN3_109 <- model(NN3_109_train, mean109 = MEAN(box_cox(value,
    lambda = guerr[9])), naive109 = NAIVE(box_cox(value, lambda = guerr[9])),
    ETSsimple109 = ETS(box_cox(value, lambda = guerr[9])), seasonal109 = ETS(box_cox(value,
        lambda = guerr[9]) ~ error("A") + trend("N") + season("A")),
    damped109 = ETS(box_cox(value, lambda = guerr[9]) ~ error("A") +
        trend("Ad") + season("A")), arima109 = ARIMA(box_cox(value,
        guerr[9]), stepwise = TRUE)) %>%
    forecast(h = "18 months")

fc_NN3_110 <- model(NN3_110_train, mean110 = MEAN(box_cox(value,
    lambda = guerr[10])), naive110 = NAIVE(box_cox(value, lambda = guerr[10])),
    ETSsimple110 = ETS(box_cox(value, lambda = guerr[10])), seasonal110 = ETS(box_cox(value,
        lambda = guerr[10]) ~ error("A") + trend("N") + season("A")),
    damped110 = ETS(box_cox(value, lambda = guerr[10]) ~ error("A") +
        trend("Ad") + season("A")), arima110 = ARIMA(box_cox(value,
        guerr[10]), stepwise = TRUE)) %>%
    forecast(h = "18 months")
```

```
fc_NN3_111 <- model(NN3_111_train, mean111 = MEAN(box_cox(value,
    lambda = -0.5)), naive111 = NAIVE(box_cox(value, lambda = -0.5)),
    ETSsimple111 = ETS(box_cox(value, lambda = -0.5)), seasonal111 = ETS(box_cox(value,
        lambda = -0.5) ~ error("A") + trend("N") + season("A")),
    damped111 = ETS(box_cox(value, lambda = -0.5) ~ error("A") +
        trend("Ad") + season("A")), arima111 = ARIMA(box_cox(value,
        -0.5), stepwise = TRUE)) %>%
    forecast(h = "18 months")
```

```
# own method for MAPE and MSE:

smape <- function(test, fc) {

    models <- sapply(fc[".model"], unique)  # retrieve models
    mape_out_score <- list()  # empty list for output retrieval
    mse_out_score <- list()  # empty list for output retrieval

    for (c in models) {
        # input ts is filtered according to model column
        data <- select(filter(fc_NN3_101, .model == c), .mean,
            value)
        # MAPE calculation:
        temp_mape <- mean(abs((test$value - data$.mean)/test$value)) *
            100
        mape_out_score[[c]] <- temp_mape
        # MSE calculation:
        temp_mse <- mean((test$value - data$.mean)^2)
        mse_out_score[[c]] <- temp_mse

    }
    # Merging lists
    mape_final <- do.call(rbind, mape_out_score)
    mse_final <- do.call(rbind, mse_out_score)
    # Merging Scores
    scores_final <- cbind(.MAPE = mape_final, .MSE = mse_final)
    return(scores_final)
}
```

```
res_smape_mse <- as.data.frame(rbind(smape(NN3_101_test, fc_NN3_101),
    smape(NN3_102_test, fc_NN3_102), smape(NN3_103_test, fc_NN3_103),
    smape(NN3_104_test, fc_NN3_104), smape(NN3_105_test, fc_NN3_105),
    smape(NN3_106_test, fc_NN3_106), smape(NN3_107_test, fc_NN3_107),
    smape(NN3_108_test, fc_NN3_108), smape(NN3_109_test, fc_NN3_109),
    smape(NN3_110_test, fc_NN3_110), smape(NN3_111_test, fc_NN3_111))) %>%
    rownames_to_column()

names(res_smape_mse)[1] <- ".model"
names(res_smape_mse)[2] <- "SMAPE"
names(res_smape_mse)[3] <- "MSE"


res_acc <- rbind(accuracy(fc_NN3_101, NN3_101_test), accuracy(fc_NN3_102,
    NN3_102_test), accuracy(fc_NN3_103, NN3_103_test), accuracy(fc_NN3_104,
    NN3_104_test), accuracy(fc_NN3_105, NN3_105_test), accuracy(fc_NN3_106,
    NN3_106_test), accuracy(fc_NN3_107, NN3_107_test), accuracy(fc_NN3_108,
    NN3_108_test), accuracy(fc_NN3_109, NN3_109_test), accuracy(fc_NN3_110,
    NN3_110_test), accuracy(fc_NN3_111, NN3_111_test)) %>%
    select(.model, MAE, MAPE, RMSE)

final_res <- merge(res_acc, res_smape_mse, by = ".model")
final_res %>%
    arrange(MSE) %>%
    head() %>%
    gt() %>%
    tab_header(title = md("**Accuracy measures**"), subtitle = md("sorted by MSE")) %>%
    opt_align_table_header(align = "center")
```

Accuracy measures

sorted by MSE

.model	MAE	MAPE	RMSE	SMAPE	MSE
seasonal101	110	2.07	137	2.07	18811
damped101	112	2.10	139	2.10	19185
ETSsimple101	112	2.12	141	2.12	19854
arima101	121	2.27	150	2.27	22624
naive101	179	3.34	224	3.34	50275
mean101	292	5.45	336	5.45	112883

```
res_acc %>%
    select(MAE) %>%
    unlist() %>%
    which.min()  # Refers to ARIMA 101
```

```
## MAE6
##    6
```

```
res_acc %>%
    select(MAPE) %>%
    unlist() %>%
    which.min()  # Refers to ARIMA 101
```

```
## MAPE6
##     6
```

```
res_acc %>%
    select(RMSE) %>%
    unlist() %>%
    which.min()  # Refers to ETSsimple105
```

```
## RMSE25
##     25
```

```
res_smape_mse %>%
    select(SMAPE) %>%
    unlist() %>%
    which.min()  # Refers to seasonal 101
```

```
## SMAPE4
##      4
```

```
res_smape_mse %>%
    select(MSE) %>%
    unlist() %>%
    which.min()  # Refers to to ETSsimple105
```

```
## MSE4
##    4
```