#STL

Seasonal Decomposition of Time Series

"Seasonal and Trend decomposition using Loess"

```
#load packages
library(readr)
library(plyr)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(ggplot2)
library(tsibble)
```

```
##
## Attaching package: 'tsibble'

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, union
```

```
library(fable)
```

```
## Loading required package: fabletools
```

```
library(zoo)
```

```
##
## Attaching package: 'zoo'

## The following object is masked from 'package:tsibble':
##
##     index

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

```
library(fpp3)
```

```
## -- Attaching packages --------------------------------------- fpp3 0.4.0 --

## v tibble      3.1.2      v tsibbledata 0.3.0
## v tidyr       1.1.3      v feasts      0.2.2
## v lubridate   1.7.10

## -- Conflicts ------------------------------------------------ fpp3_conflicts --
## x dplyr::arrange()      masks plyr::arrange()
## x dplyr::count()        masks plyr::count()
## x lubridate::date()     masks base::date()
## x dplyr::failwith()     masks plyr::failwith()
## x dplyr::filter()       masks stats::filter()
## x dplyr::id()           masks plyr::id()
## x zoo::index()          masks tsibble::index()
## x tsibble::intersect()  masks base::intersect()
## x lubridate::interval() masks tsibble::interval()
## x dplyr::lag()          masks stats::lag()
## x dplyr::mutate()       masks plyr::mutate()
## x dplyr::rename()       masks plyr::rename()
## x tsibble::setdiff()    masks base::setdiff()
## x dplyr::summarise()    masks plyr::summarise()
## x dplyr::summarize()    masks plyr::summarize()
## x tsibble::union()      masks base::union()
```

```
df <- read_csv(file = "data/data_interpolated_with_diesel.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]

##
## -- Column specification --------------------------------------------------------
## cols(
##   X1 = col_double(),
##   Mode = col_character(),
##   ORegionDAT = col_character(),
##   DRegionDAT = col_character(),
##   yw = col_character(),
##   sanitized_cost = col_double(),
##   prcp = col_double(),
##   tavg = col_double(),
##   tmax = col_double(),
##   tmin = col_double(),
##   approx_cost = col_double(),
##   diesel_price = col_double()
## )
```

```r
ts <- df %>%
  mutate(yw = yearweek(yw)) %>%
  as_tsibble(key = c(Mode,ORegionDAT, DRegionDAT), index = yw)
```

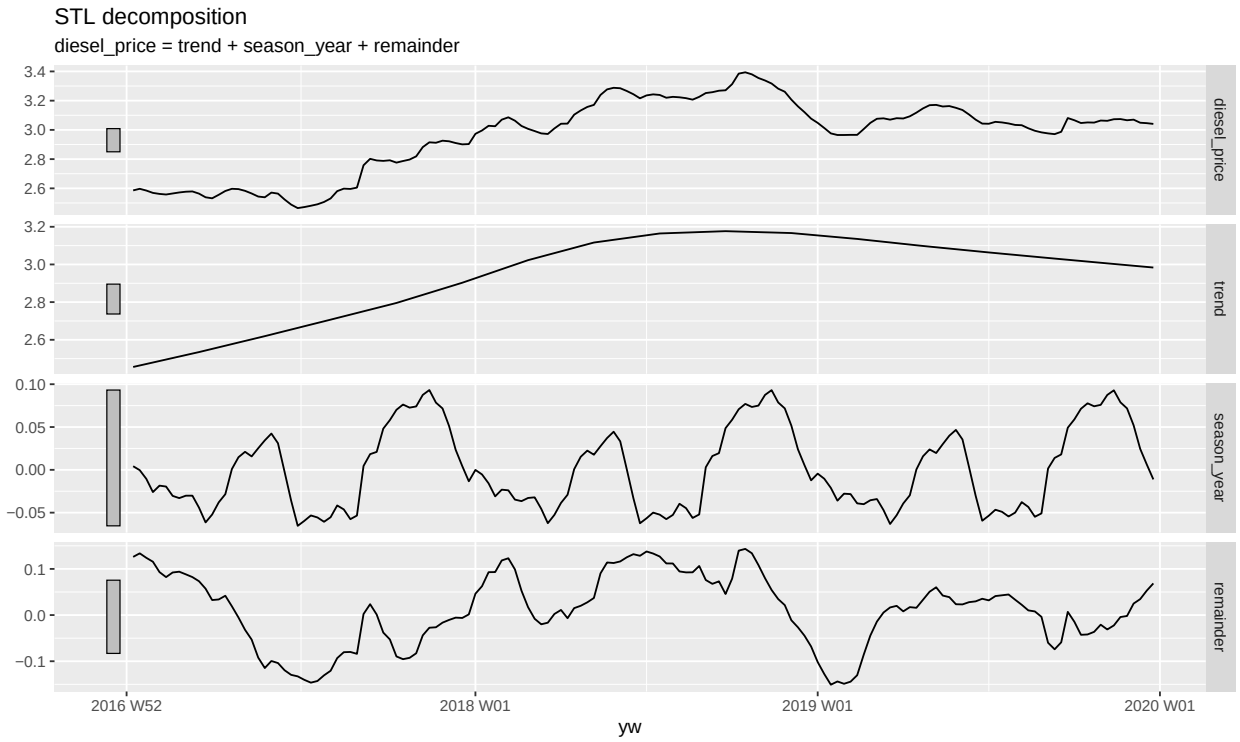
```r
#select diesel prices in just one lane (so it's not repeated)

dieselTS <- ts %>% select(diesel_price) %>% filter(Mode == "R", ORegionDAT == "CA_FRS", DRegionDAT == "
dieselTS
```

```
## # A tsibble: 235 x 5 [1W]
## # Key:       Mode, ORegionDAT, DRegionDAT [1]
##    diesel_price       yw Mode  ORegionDAT DRegionDAT
##           <dbl>   <week> <chr> <chr>      <chr>
##  1         2.59 2017 W01 R     CA_FRS     IL_CHI
##  2         2.60 2017 W02 R     CA_FRS     IL_CHI
##  3         2.58 2017 W03 R     CA_FRS     IL_CHI
##  4         2.57 2017 W04 R     CA_FRS     IL_CHI
##  5         2.56 2017 W05 R     CA_FRS     IL_CHI
##  6         2.56 2017 W06 R     CA_FRS     IL_CHI
##  7         2.56 2017 W07 R     CA_FRS     IL_CHI
##  8         2.57 2017 W08 R     CA_FRS     IL_CHI
##  9         2.58 2017 W09 R     CA_FRS     IL_CHI
## 10         2.58 2017 W10 R     CA_FRS     IL_CHI
## # ... with 225 more rows
```

```r
#create training set - up through 2020 of the time series
train <- dieselTS %>%
  filter_index(~ "2019 W52")
```

```r
#create and plot STL decompositions of training set
dcmp <- train %>% model(STL(diesel_price))
components(dcmp) %>% autoplot()
```



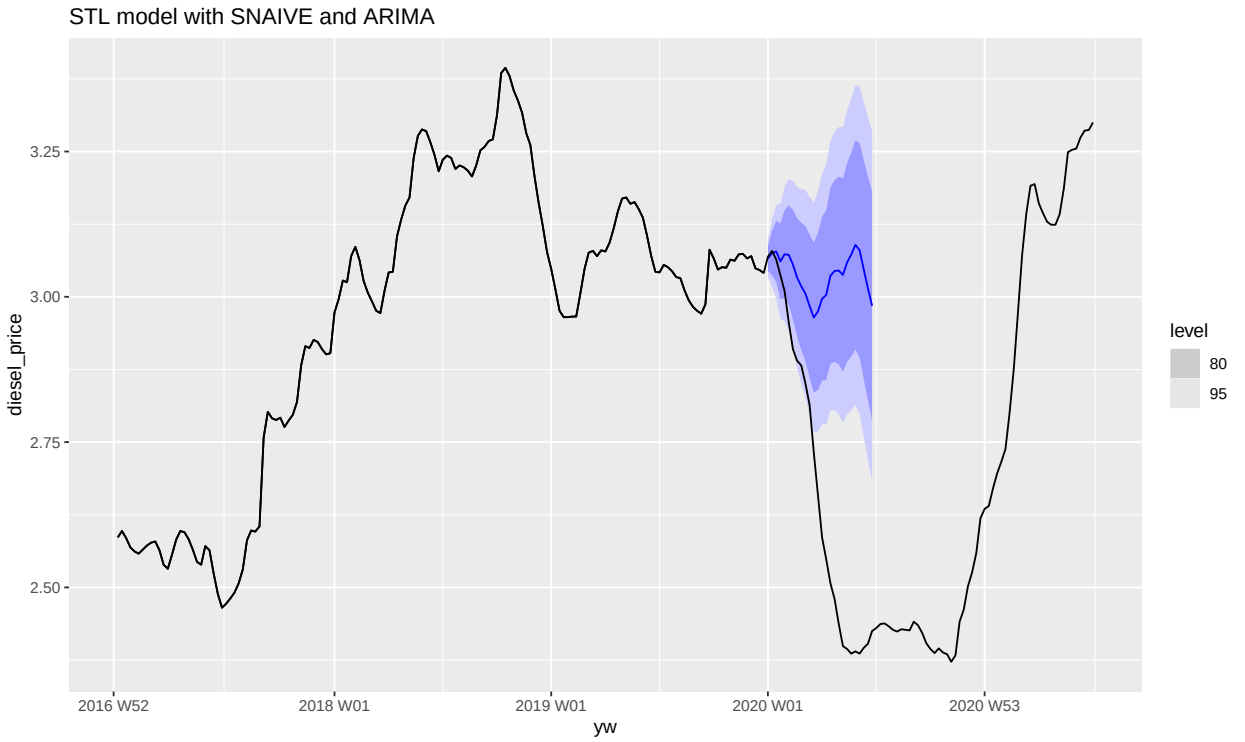

```
#fit model
dcmp <- train %>%
  model(stlf = decomposition_model(
    STL(diesel_price),
    SNAIVE(season_year),
    ARIMA(season_adjust)
  ))
```

```
## Warning in sqrt(diag(best$var.coef)): NaNs produced
```

```
fc <- dcmp %>%
  forecast(h=26)
```

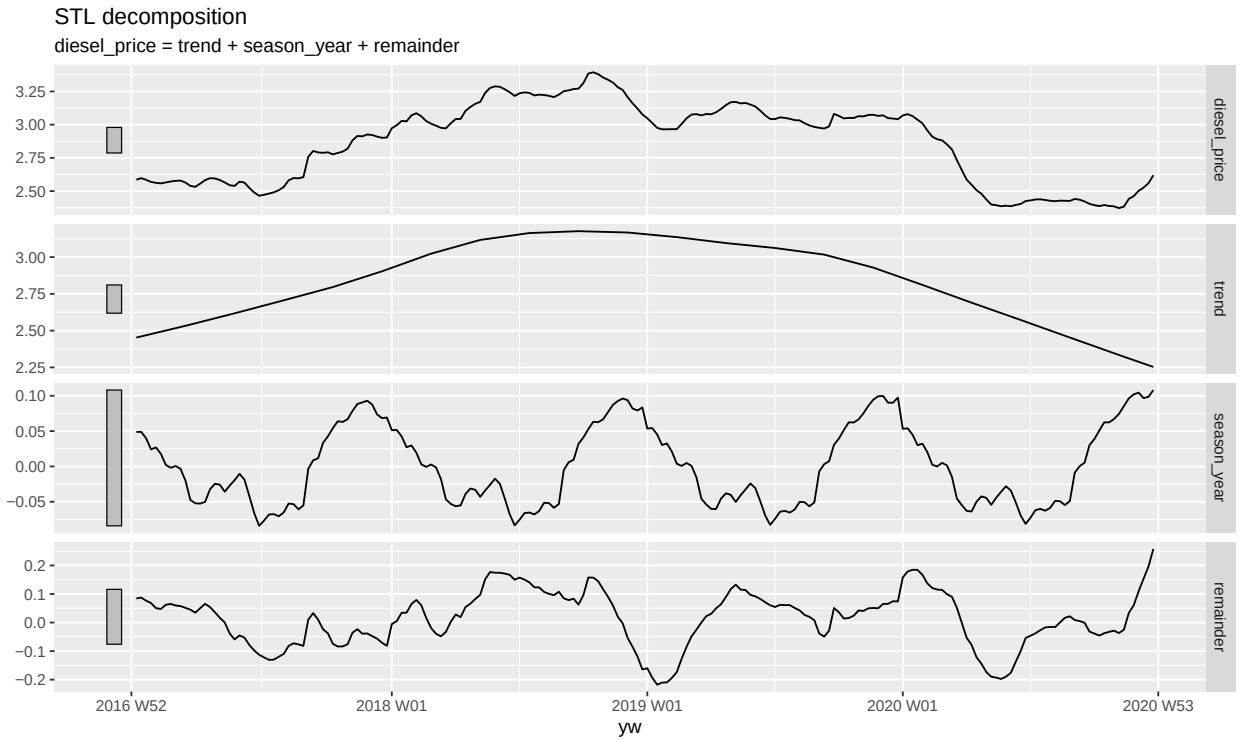
```
#plot
fc %>%
  autoplot(train) +
  autolayer(dieselTS, colour = "black") +
  labs(title="STL model with SNAIVE and ARIMA")
```

```
## Plot variable not specified, automatically selected `.vars = diesel_price`
```




```
#create training set - up through 2020 of the time series
train <- dieselTS %>%
  filter_index(~ "2020 W52")
```

```
#create and plot STL decompositions of training set
dcmp <- train %>% model(STL(diesel_price))
components(dcmp) %>% autoplot()
```

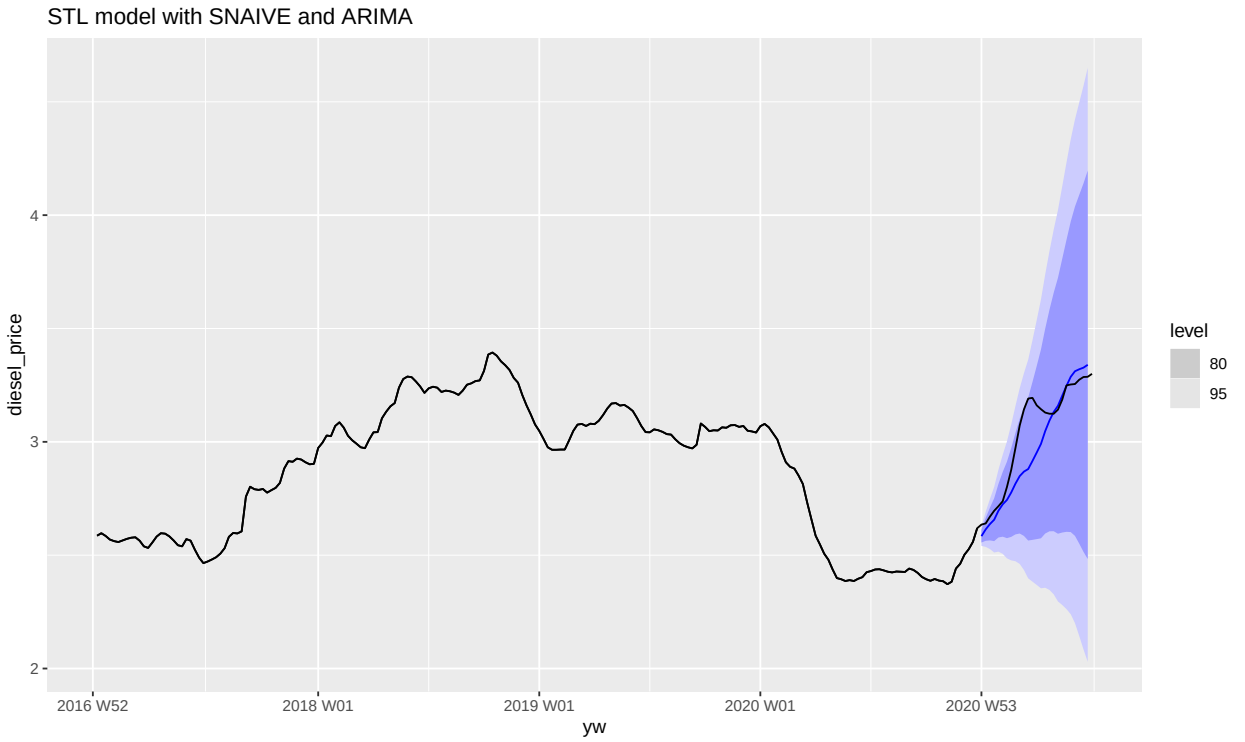



```
#fit model
dcmp <- train %>%
  model(stlf = decomposition_model(
    STL(diesel_price),
    SNAIVE(season_year),
    ARIMA(season_adjust)
  ))
```

```
fc <- dcmp %>%
  forecast(h=26)
```

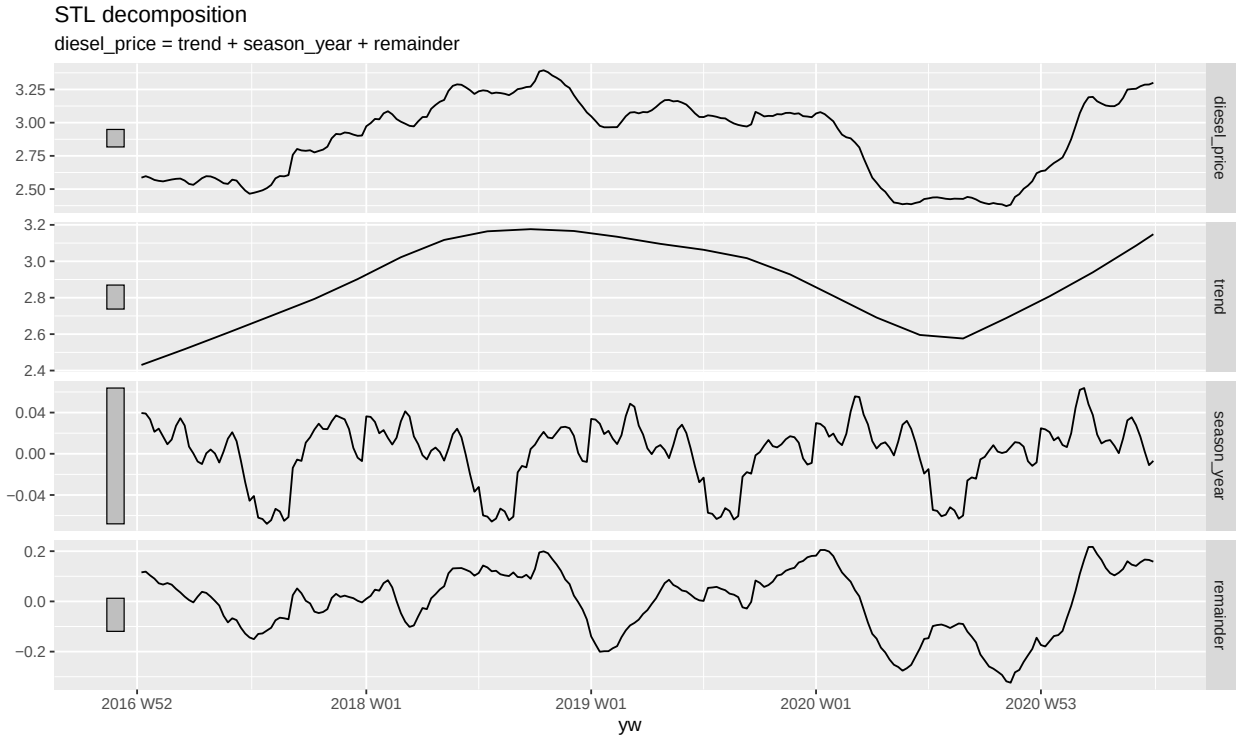
```
#plot
fc %>%
  autoplot(train) +
  autolayer(dieselTS, colour = "black") +
  labs(title="STL model with SNAIVE and ARIMA")
```

```
## Plot variable not specified, automatically selected `.vars = diesel_price`
```




```
#create training set - up through 2020 of the time series
train <- dieselTS %>%
  filter_index(~ "2021 W26")
```

```
#create and plot STL decompositions of training set
dcmp <- train %>% model(STL(diesel_price))
components(dcmp) %>% autoplot()
```

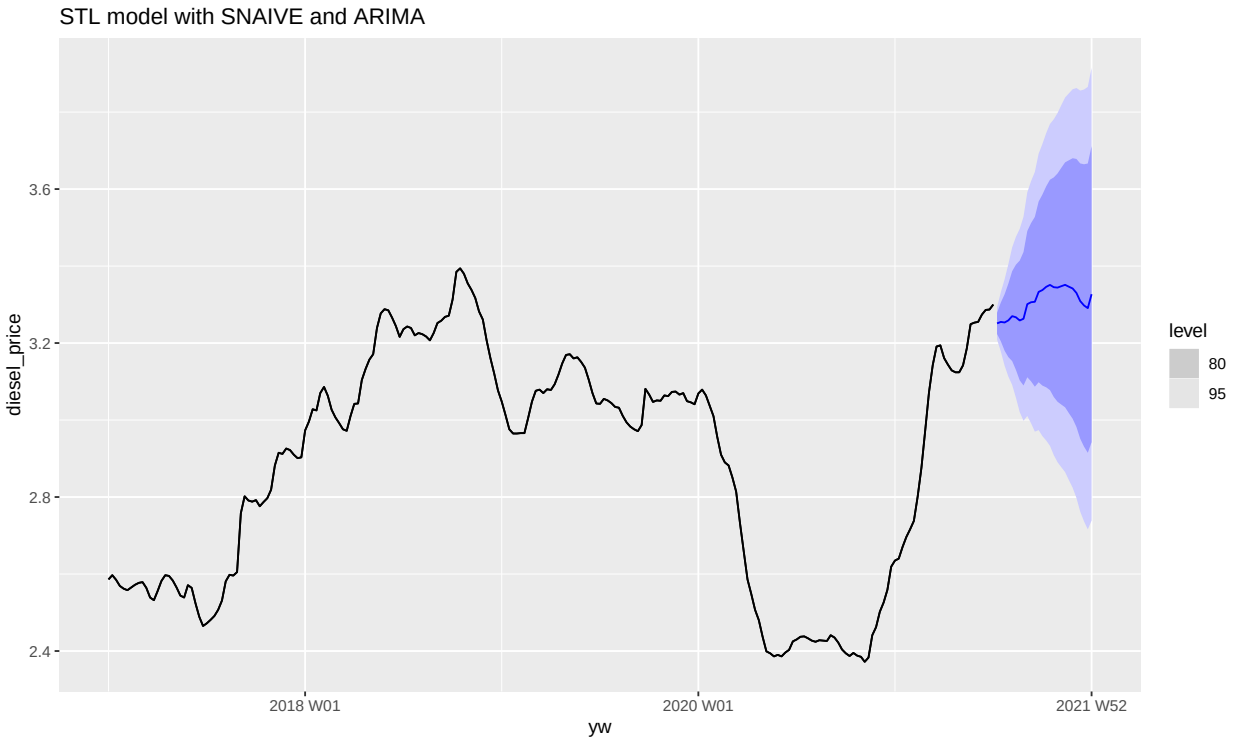



```
#fit model
dcmp <- train %>%
  model(stlf = decomposition_model(
    STL(diesel_price),
    SNAIVE(season_year),
    ARIMA(season_adjust)
  ))
```

```
fc <- dcmp %>%
  forecast(h=26)
```

```
#plot
fc %>%
  autoplot(train) +
  autolayer(dieselTS, colour = "black") +
  labs(title="STL model with SNAIVE and ARIMA")
```

```
## Plot variable not specified, automatically selected `.vars = diesel_price`
```


STL model with SNAIVE and ARIMA
diesel_price
3.6
3.2
2.8
2.4
2018 W01
2020 W01
2021 W52
yw
level
80
95