TimeSeries_Forecast

RJ

8/16/2020

Time Series - Forecast

This document summarizes the learning process on Time Series analysis. The original source of the learning material is from Datacamp.

Preparation

Load the dataset and the required packages

```
library(tidyverse)
```

```
## -- Attaching packages ------------------------------------------------------------------------------
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.2     v dplyr   1.0.0
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## -- Conflicts ----------------------------------------------------------------------------------------
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

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

```
library(forecast)
```

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

```r
library(Hmisc)
```

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
## 
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:dplyr':
## 
##     src, summarize
```

```
## The following objects are masked from 'package:base':
## 
##     format.pval, units
```

```r
library(magrittr)
```

```
## 
## Attaching package: 'magrittr'
```

```
## The following object is masked from 'package:purrr':
## 
##     set_names
```

```
## The following object is masked from 'package:tidyr':
## 
##     extract
```

```r
ptt_price <- read.csv(choose.files())
head(ptt_price)
```

```
##   adm0_id adm0_name adm1_id            adm1_name mkt_id mkt_name cm_id
## 1     205    Rwanda   21973 $West/Iburengerazuba   1045  Birambo   148
## 2     205    Rwanda   21973 $West/Iburengerazuba   1045  Birambo   148
## 3     205    Rwanda   21973 $West/Iburengerazuba   1045  Birambo   148
## 4     205    Rwanda   21973 $West/Iburengerazuba   1045  Birambo   148
## 5     205    Rwanda   21973 $West/Iburengerazuba   1045  Birambo   148
## 6     205    Rwanda   21973 $West/Iburengerazuba   1045  Birambo   148
##            cm_name cur_id cur_name pt_id pt_name um_id um_name mp_month mp_year
## 1 Potatoes (Irish)     77      RWF    15  Retail     5      KG       11    2010
## 2 Potatoes (Irish)     77      RWF    15  Retail     5      KG       12    2010
## 3 Potatoes (Irish)     77      RWF    15  Retail     5      KG        1    2011
## 4 Potatoes (Irish)     77      RWF    15  Retail     5      KG        2    2011
## 5 Potatoes (Irish)     77      RWF    15  Retail     5      KG        3    2011
## 6 Potatoes (Irish)     77      RWF    15  Retail     5      KG        4    2011
##   mp_price mp_commoditysource
## 1 157.0000            MINAGRI
## 2 133.3333            MINAGRI
## 3  96.5000            MINAGRI
## 4  97.0000            MINAGRI
## 5 107.8000            MINAGRI
## 6 125.5000            MINAGRI
```

```r
# Show unique value in each column
summarise_all(ptt_price, n_distinct)
```

```
##   adm0_id adm0_name adm1_id adm1_name mkt_id mkt_name cm_id cm_name cur_id
## 1       1         1       5         5     63       63     1       1      1
##   cur_name pt_id pt_name um_id um_name mp_month mp_year mp_price
## 1        1     1       1     1       1       12       8     1452
##   mp_commoditysource
## 1                  1
```

```r
# Select column with more than one unique value
ptt_price_selected <- ptt_price %>%
  select_if(summarise_all(., n_distinct) > 1) %>%
  rename(region_id = adm1_id,
         region = adm1_name,
         market_id = mkt_id,
         market = mkt_name,
         month = mp_month,
         year = mp_year,
         price = mp_price) %>%
  mutate(date = ymd(paste(year, month, "01"))) %>%
  select(region_id, market, date, price)

describe(ptt_price_selected)
```

```
## ptt_price_selected
##
##  4  Variables      4320  Observations
## --------------------------------------------------------------------------------
## region_id
##        n  missing distinct     Info     Mean      Gmd
##     4320        0        5    0.951    20262     3120
##
## lowest :  2587 21969 21971 21972 21973, highest:  2587 21969 21971 21972 21973
##
## Value       2587 21969 21971 21972 21973
## Frequency    381   883  1030   919  1107
## Proportion 0.088 0.204 0.238 0.213 0.256
## --------------------------------------------------------------------------------
## market
##        n  missing distinct
##     4320        0       63
##
## lowest : Base      Birambo   Bugarama  Buhanda   Bumazi
## highest: Rukomo    Rushashi  Rusine    Rwagitima Rwamagana
## --------------------------------------------------------------------------------
## date
##          n    missing   distinct       Info       Mean        Gmd        .05
##       4320          0         96          1 2012-07-01      933.9 2008-07-01
##        .10        .25        .50        .75        .90        .95
## 2009-02-01 2010-11-01 2012-11-01 2014-05-08 2015-05-01 2015-09-01
##
## lowest : 2008-01-01 2008-02-01 2008-03-01 2008-04-01 2008-05-01
## highest: 2015-08-01 2015-09-01 2015-10-01 2015-11-01 2015-12-01
## --------------------------------------------------------------------------------
## price
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##     4320        0     1452        1    162.6    47.77    96.25   108.33
##      .25      .50      .75      .90      .95
##   132.50   160.00   192.64   219.14   235.00
##
## lowest :  62.5000  64.5000  66.0000  70.0000  70.6667
## highest: 301.7500 301.8571 390.0000 415.7500 430.0000
## --------------------------------------------------------------------------------
```

```r
location <- list.dirs(choose.dir())
allfile <- list.files(path = location, pattern = c(".csv"))

datalist <- list()
for (x in 1:length(allfile)) {
  # Read dataset
  datalist[[x]] <- read_csv(paste(location, allfile[x], sep = "\\"))
}
```

```
## Parsed with column specification:
## cols(
##   adm0_id = col_double(),
##   adm0_name = col_character(),
##   adm1_id = col_double(),
##   adm1_name = col_character(),
##   mkt_id = col_double(),
##   mkt_name = col_character(),
##   cm_id = col_double(),
##   cm_name = col_character(),
##   cur_id = col_double(),
##   cur_name = col_character(),
##   pt_id = col_double(),
##   pt_name = col_character(),
##   um_id = col_double(),
##   um_name = col_character(),
##   mp_month = col_double(),
##   mp_year = col_double(),
##   mp_price = col_double(),
##   mp_commoditysource = col_character()
## )
## Parsed with column specification:
## cols(
##   adm0_id = col_double(),
##   adm0_name = col_character(),
##   adm1_id = col_double(),
##   adm1_name = col_character(),
##   mkt_id = col_double(),
##   mkt_name = col_character(),
##   cm_id = col_double(),
##   cm_name = col_character(),
##   cur_id = col_double(),
##   cur_name = col_character(),
##   pt_id = col_double(),
##   pt_name = col_character(),
##   um_id = col_double(),
##   um_name = col_character(),
##   mp_month = col_double(),
##   mp_year = col_double(),
##   mp_price = col_double(),
##   mp_commoditysource = col_character()
## )
## Parsed with column specification:
## cols(
##   adm0_id = col_double(),
##   adm0_name = col_character(),
##   adm1_id = col_double(),
##   adm1_name = col_character(),
##   mkt_id = col_double(),
##   mkt_name = col_character(),
##   cm_id = col_double(),
##   cm_name = col_character(),
##   cur_id = col_double(),
##   cur_name = col_character(),
##   pt_id = col_double(),
##   pt_name = col_character(),
##   um_id = col_double(),
##   um_name = col_character(),
##   mp_month = col_double(),
##   mp_year = col_double(),
##   mp_price = col_double(),
##   mp_commoditysource = col_character()
## )
## Parsed with column specification:
## cols(
##   adm0_id = col_double(),
##   adm0_name = col_character(),
##   adm1_id = col_double(),
##   adm1_name = col_character(),
##   mkt_id = col_double(),
##   mkt_name = col_character(),
##   cm_id = col_double(),
##   cm_name = col_character(),
##   cur_id = col_double(),
##   cur_name = col_character(),
##   pt_id = col_double(),
##   pt_name = col_character(),
##   um_id = col_double(),
##   um_name = col_character(),
##   mp_month = col_double(),
##   mp_year = col_double(),
##   mp_price = col_double(),
##   mp_commoditysource = col_character()
## )
## Parsed with column specification:
## cols(
##   adm0_id = col_double(),
##   adm0_name = col_character(),
##   adm1_id = col_double(),
##   adm1_name = col_character(),
##   mkt_id = col_double(),
##   mkt_name = col_character(),
##   cm_id = col_double(),
##   cm_name = col_character(),
##   cur_id = col_double(),
##   cur_name = col_character(),
##   pt_id = col_double(),
##   pt_name = col_character(),
##   um_id = col_double(),
##   um_name = col_character(),
##   mp_month = col_double(),
##   mp_year = col_double(),
##   mp_price = col_double(),
##   mp_commoditysource = col_character()
## )
## Parsed with column specification:
## cols(
##   adm0_id = col_double(),
##   adm0_name = col_character(),
##   adm1_id = col_double(),
##   adm1_name = col_character(),
##   mkt_id = col_double(),
##   mkt_name = col_character(),
##   cm_id = col_double(),
##   cm_name = col_character(),
##   cur_id = col_double(),
##   cur_name = col_character(),
##   pt_id = col_double(),
##   pt_name = col_character(),
##   um_id = col_double(),
##   um_name = col_character(),
##   mp_month = col_double(),
##   mp_year = col_double(),
##   mp_price = col_double(),
##   mp_commoditysource = col_character()
## )
## Parsed with column specification:
## cols(
##   adm0_id = col_double(),
##   adm0_name = col_character(),
##   adm1_id = col_double(),
##   adm1_name = col_character(),
##   mkt_id = col_double(),
##   mkt_name = col_character(),
##   cm_id = col_double(),
##   cm_name = col_character(),
##   cur_id = col_double(),
##   cur_name = col_character(),
##   pt_id = col_double(),
##   pt_name = col_character(),
##   um_id = col_double(),
##   um_name = col_character(),
##   mp_month = col_double(),
##   mp_year = col_double(),
##   mp_price = col_double(),
##   mp_commoditysource = col_character()
## )
## Parsed with column specification:
## cols(
##   adm0_id = col_double(),
##   adm0_name = col_character(),
##   adm1_id = col_double(),
##   adm1_name = col_character(),
##   mkt_id = col_double(),
##   mkt_name = col_character(),
##   cm_id = col_double(),
##   cm_name = col_character(),
##   cur_id = col_double(),
##   cur_name = col_character(),
##   pt_id = col_double(),
##   pt_name = col_character(),
##   um_id = col_double(),
##   um_name = col_character(),
##   mp_month = col_double(),
##   mp_year = col_double(),
##   mp_price = col_double(),
##   mp_commoditysource = col_character()
## )
## Parsed with column specification:
## cols(
##   adm0_id = col_double(),
##   adm0_name = col_character(),
##   adm1_id = col_double(),
##   adm1_name = col_character(),
##   mkt_id = col_double(),
##   mkt_name = col_character(),
##   cm_id = col_double(),
##   cm_name = col_character(),
##   cur_id = col_double(),
##   cur_name = col_character(),
##   pt_id = col_double(),
##   pt_name = col_character(),
##   um_id = col_double(),
##   um_name = col_character(),
##   mp_month = col_double(),
##   mp_year = col_double(),
##   mp_price = col_double(),
##   mp_commoditysource = col_character()
## )
```

```
allfood <- reduce(datalist, bind_rows) %>%
              select_if(summarise_all(., n_distinct) > 1) %>%
              rename(region_id = adm1_id,
                     region = adm1_name,
                     market_id = mkt_id,
                     market = mkt_name,
                     food = cm_name,
                     month = mp_month,
                     year = mp_year,
                     price = mp_price) %>%
              mutate(date = ymd(paste(year, month, "01"))) %>%
              select(region_id, market, date, food, price)

describe(allfood)
```

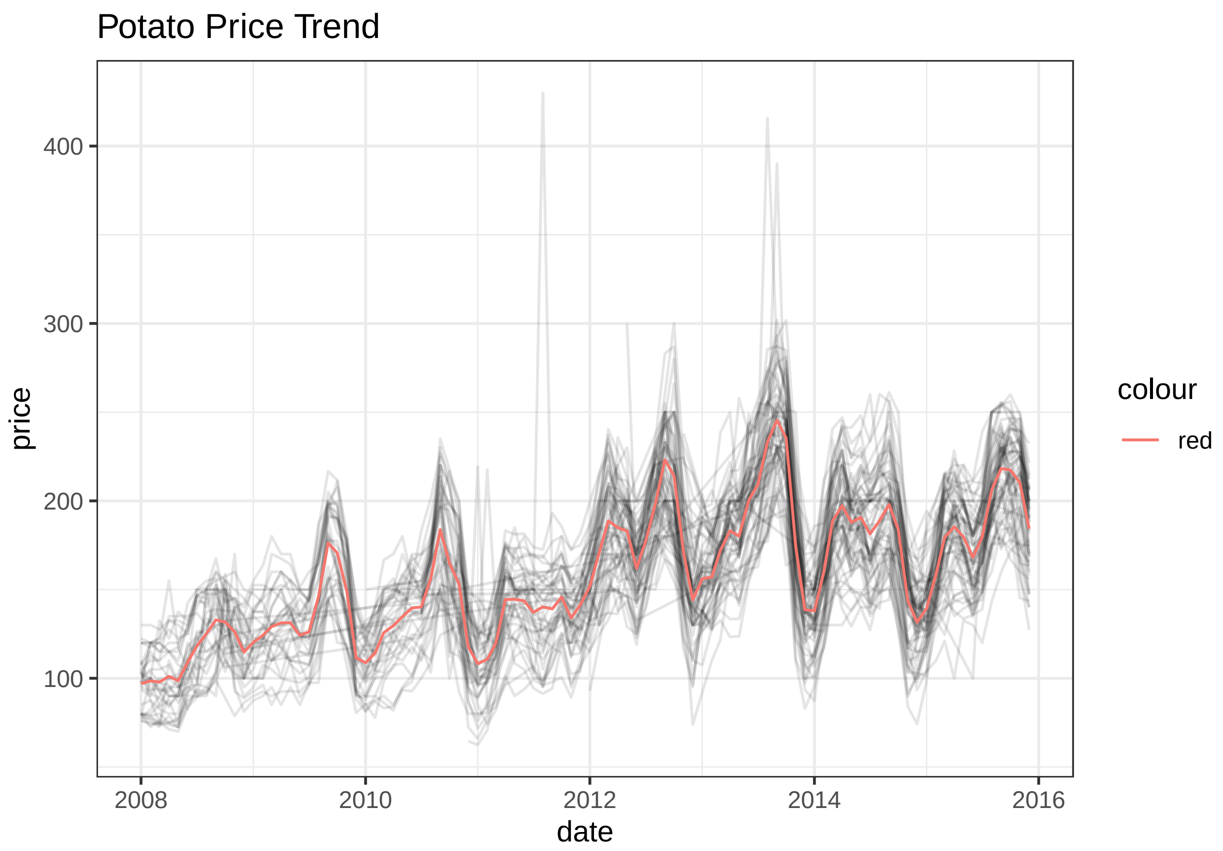
```
## allfood
##
##  5  Variables      27387  Observations
## --------------------------------------------------------------------------------
## region_id
##        n  missing distinct     Info     Mean      Gmd
##    27387        0        5    0.951    20332     3003
##
## lowest :  2587 21969 21971 21972 21973, highest:  2587 21969 21971 21972 21973
##
## Value       2587 21969 21971 21972 21973
## Frequency   2316  5843  6631  5804  6793
## Proportion 0.085 0.213 0.242 0.212 0.248
## --------------------------------------------------------------------------------
## market
##        n  missing distinct
##    27387        0       63
##
## lowest : Base      Birambo   Bugarama  Buhanda   Bumazi
## highest: Rukomo    Rushashi  Rusine    Rwagitima Rwamagana
## --------------------------------------------------------------------------------
## date
##          n    missing   distinct       Info       Mean        Gmd        .05
##      27387          0         96          1 2012-12-03      881.2 2008-09-01
##        .10        .25        .50        .75        .90        .95
## 2009-07-01 2011-07-01 2013-06-01 2014-09-01 2015-06-01 2015-09-01
##
## lowest : 2008-01-01 2008-02-01 2008-03-01 2008-04-01 2008-05-01
## highest: 2015-08-01 2015-09-01 2015-10-01 2015-11-01 2015-12-01
## --------------------------------------------------------------------------------
## food
##        n  missing distinct
##    27387        0        9
##
## lowest : Beans (dry)        Cassava            Chili (red)        Maize              Oranges (big si
## highest: Oranges (big size) Peas (fresh)       Potatoes (Irish)   Sorghum            Tomatoes
##
```

```
## Beans (dry) (4357, 0.159), Cassava (2766, 0.101), Chili (red) (1938, 0.071),
## Maize (4024, 0.147), Oranges (big size) (1843, 0.067), Peas (fresh) (1893,
## 0.069), Potatoes (Irish) (4320, 0.158), Sorghum (4099, 0.150), Tomatoes (2147,
## 0.078)
## --------------------------------------------------------------------------------
## price
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##    27387        0     6208        1    385.8    290.3    129.8    150.0
##      .25      .50      .75      .90      .95
##    200.0    280.0    416.0    800.0   1055.6
##
## lowest :   50.0000   62.5000   64.5000   66.0000   67.0833
## highest: 2900.0000 3000.0000 3283.3333 3500.0000 6500.0000
## --------------------------------------------------------------------------------
```

Exploration

This

```r
allfood %>%
  filter(food == "Potatoes (Irish)") %>%
  group_by(year(date), month(date)) %>%
  mutate(average_price = mean(price)) %>%
  ggplot() +
    geom_line(aes(x = date,
                  y = price,
                  group = market),
              alpha = 0.1) +
    geom_line(aes(x = date,
                  y = average_price,
                  color = "red")) +
    theme_bw() +
    labs(title = "Potato Price Trend")
```



Forecasting

Forecasting preparation. As mentioned in the source, best way to conduct forecasting is by using a `ts` object. Further research should be done to ease this process.

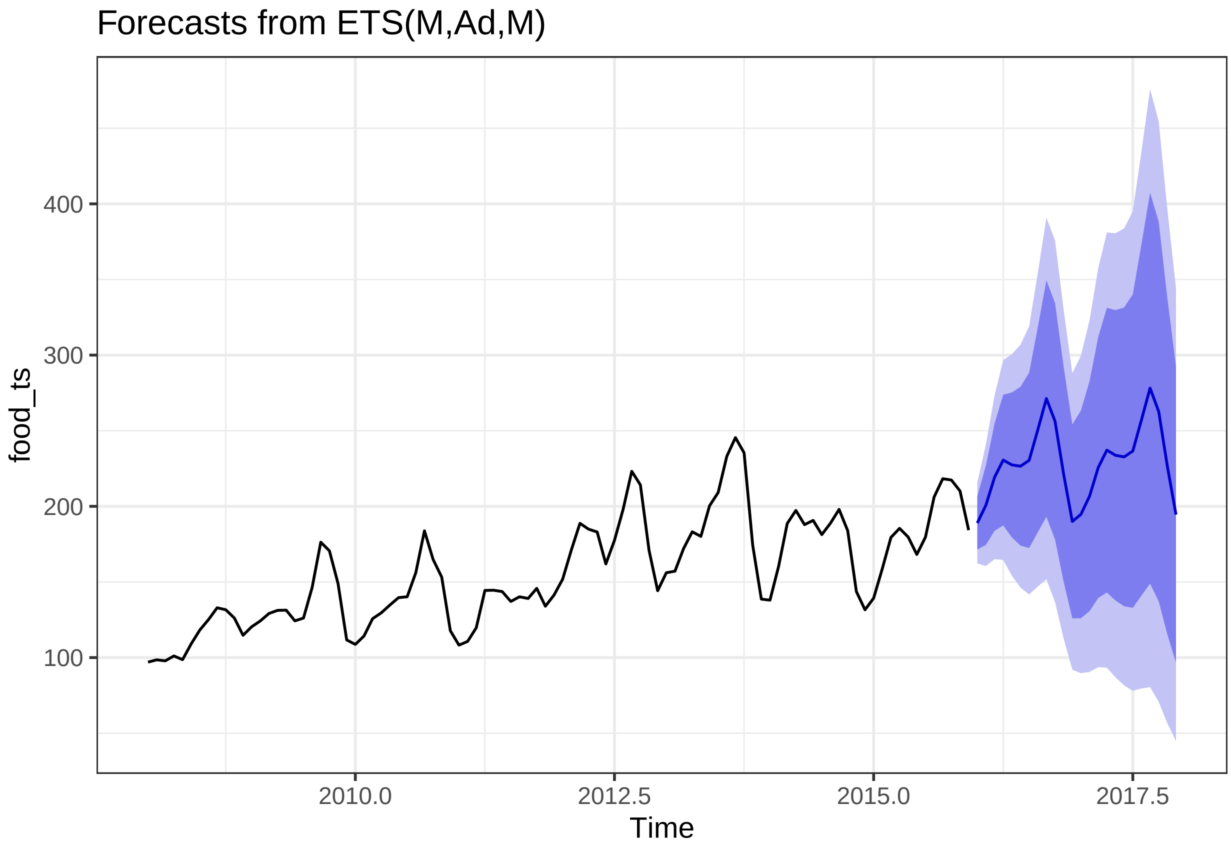
```
# Make list
food_price_list <- allfood %>%
  group_by(food, date) %>%
  summarise(price = mean(price)) %>%
  group_split(food, .keep = F) %>%
  setNames(unique(allfood$food))
```

```
## 'summarise()' regrouping output by 'food' (override with '.groups' argument)

## Warning: ... is ignored in group_split(<grouped_df>), please use
## group_by(..., .add = TRUE) %>% group_split()
```

```
# Create ts object for forecasting
food_ts <- food_price_list$`Potatoes (Irish)` %$%
              ts(price,
                 start = c(year(min(date)), month(min(date))),
                 end = c(year(max(date)), month(max(date))),
                 frequency = 12)

food_ts
```

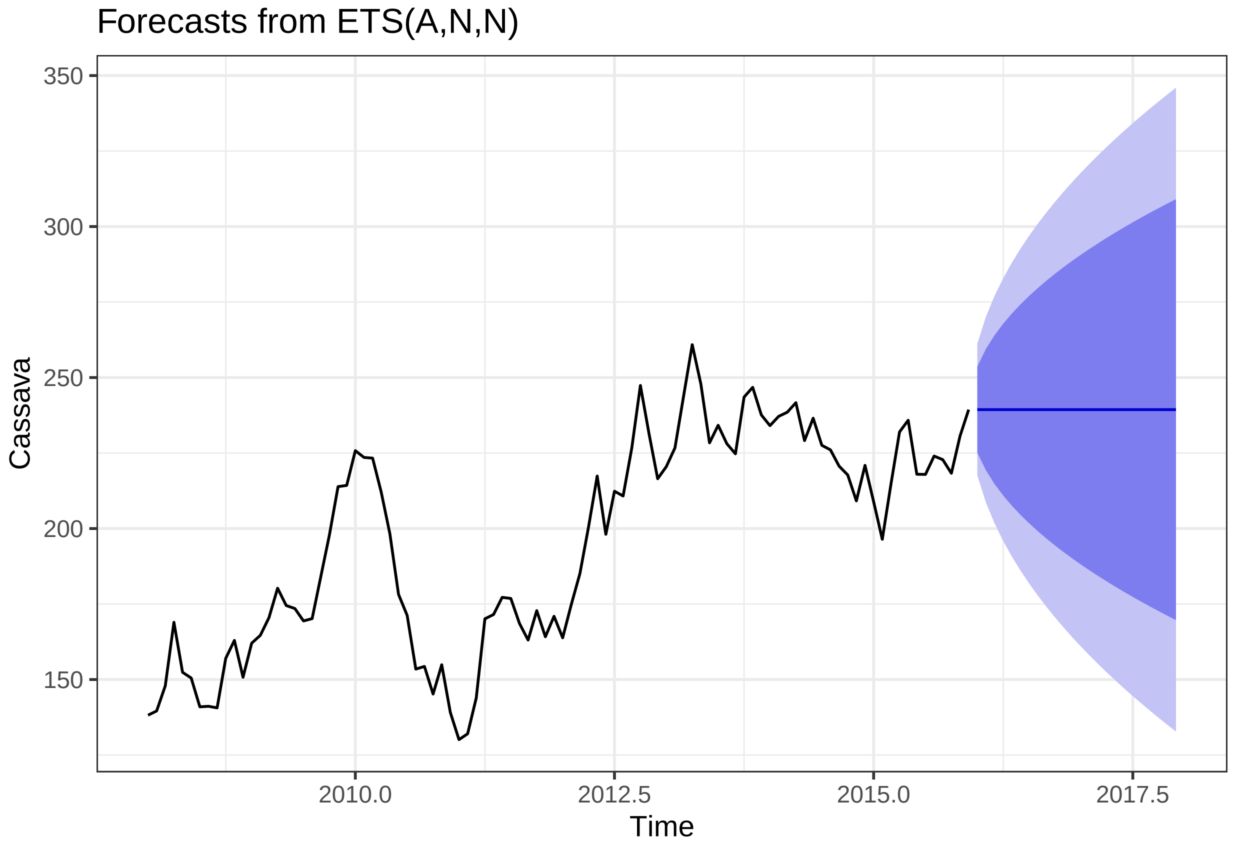
```
##            Jan       Feb       Mar       Apr       May       Jun       Jul
## 2008  97.00000  98.52273  97.84091 101.09848  98.61364 109.09091 118.24243
## 2009 120.37500 124.25000 129.16667 131.21944 131.35484 124.28472 126.13548
## 2010 108.72656 114.24059 125.73334 129.57393 134.74327 139.62608 140.22014
## 2011 108.22805 110.70811 119.71144 144.38943 144.52988 143.65701 137.11667
## 2012 151.84952 171.07966 188.77012 184.90020 183.04400 161.90685 177.52511
## 2013 156.10126 157.13208 172.10822 183.19044 180.13387 200.35022 209.11998
## 2014 137.98415 160.31585 188.66735 197.28938 187.88636 190.71303 181.32434
## 2015 139.30230 158.76986 179.43795 185.43599 179.71824 168.17693 179.65607
##            Aug       Sep       Oct       Nov       Dec
## 2008 125.11364 132.91667 131.64815 126.10417 114.74444
## 2009 146.32930 176.28522 170.61989 148.90484 111.67473
## 2010 156.15862 183.81879 164.90942 153.19605 117.75421
## 2011 140.21399 139.08074 145.76252 133.94809 141.34497
## 2012 198.02347 223.17500 214.16667 170.93782 144.20070
## 2013 233.07375 245.46842 235.40024 174.39085 138.67057
## 2014 188.84630 197.97793 183.84359 143.67184 131.59355
## 2015 206.16530 218.25415 217.41502 210.15317 184.24260
```

```
food_forcast <- forecast(object = food_ts)
autoplot(food_forcast) +
  theme_bw()
```



Simplify by making a function Using the food_price_list

```r
plot_predict <- function(dataset, foodLabel = "Food") {
dataset %$%
  ts(price,
  start = c(year(min(date)), month(min(date))),
  end = c(year(max(date)), month(max(date))),
  frequency = 12) %>%
  forecast() %>%
  autoplot() +
    theme_bw() +
    labs(y = foodLabel)
}

plot_predict(dataset = food_price_list$Cassava, foodLabel = "Cassava")
```


Forecasts from ETS(A,N,N)
Cassava
Time