Tidy Time Series Analysis and Forecasting in R

Pride Inn Azure Hotel, January 17-19, 2022

John Karuitha

1 Background

Prepare data for use in R for time series analysis and forecasting.

- Create a tsibble objects for time series analysis.
- Use tsibble functions to prepare data for use in R for time series analysis and forecasting.

```r
if(!require(pacman)){
        install.packages("pacman")
}
```

```
Loading required package: pacman
```

```r
pacman::p_load(tidyverse, tsibble, fable,
feasts, tsibbledata, fpp3, here, janitor,
forecast)
```

2 Data preparation

- Do data preparation in `tsibble`.
- We have the tsibble the package and the data structure.
- A tsibble has a time index and the measurement variable or variables. The time index is mandatory.
- The key variable, eg region, type of vaccine help us to identify unique time series. This key is optional.

- A tsibble works with tidyverse functions unlike zoo and xts.

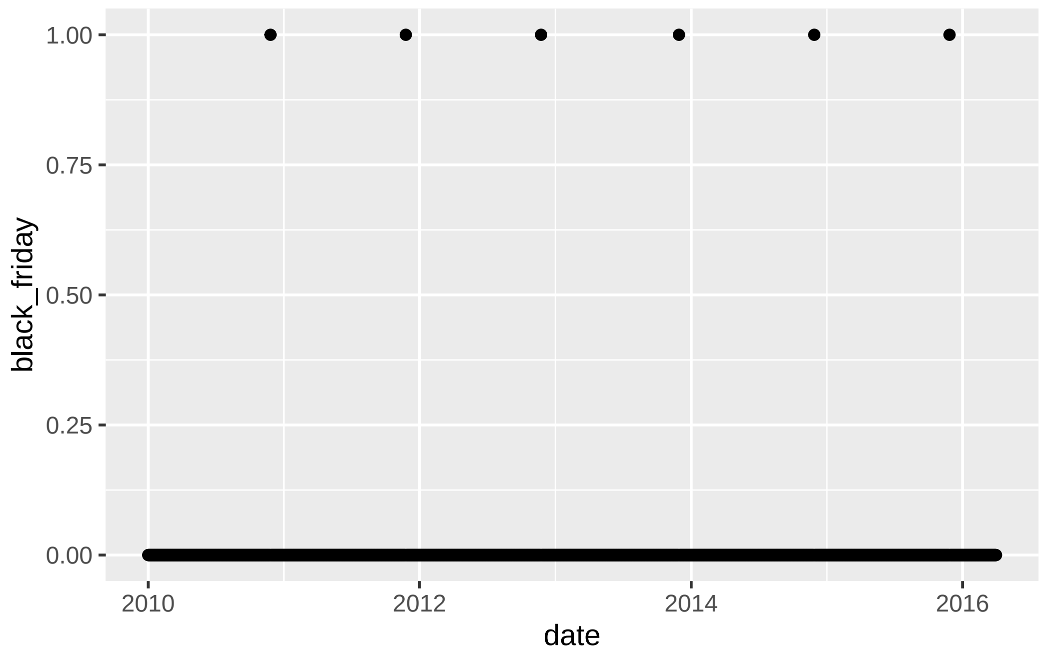
Explicit missing values: no one knows. Implicit missing values: use as_tsibble to create a tsibble- takes index and key. Exclamation mark means tsibble detects irregular intervals. Fix this by making it regular. check gaps. has_gaps, count_gaps, scan_gaps, fill_gaps. In our example - fill_gaps(trips = 0L) where trips is the measurement of interest.

From here we can work with tidyverse functions on a tsibble. When grouping by the index- use index_by, not group_by.

2.1 lab 1

```r
read_csv("forecasting_workshop_labs/data/holiday.csv") %>%
        clean_names() %>%
        mutate(date = dmy(date)) %>%
        as_tsibble(index = date) %>%
        fill_gaps() %>%
        ggplot(mapping = aes(x = date,
        y = black_friday)) +
        geom_point()
```

```
Rows: 2282 Columns: 6
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (1): date
dbl (5): Black Friday, Boxing Day, Christmas Day, Halloween Day, New Years Day

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```



```
??ggsubseriesplot
??ggACF
```

2.2 Forecasting techniques

- Average method: The mean is the forecast: Gives a flat line.
- Naive method: Previous observation used as forecast
- Seasonal Naive (snaive): Previous season observation used as the forecast. You have to know the cycle of seasonality- but R checks this season automatically. In `fable` we use similar notation with the `lm` function for linear models.

```
model_name(response_variable ~ term1 + term2 + ...)
SNAIVE(Beer ~ lag("year"))
```

We can train many models at once using the `model()` function to model many methods.

```
my_mable <- my_data %>%

        model(
                model1 = model_name(response_variable ~ term1 + term2 + ...),
                model2 = model_name(response_variable ~ term1 + term2 + ...),
                model3 = model_name(response_variable ~ term1 + term2 + ...)
        )
```

The result is a `mable` or model table object.

if we have new data or updated data, we can use the `stream()` function to update the models.

```
updated_models <- my_mable %>% stream(new_data)
```

The mable has columns equivalent to the number of models. The rows represent the time series. If you have many time series then you get many rows. To explore the mable, we use `report`, `tidy`, and `glance` from the broom package.

```
my_mable %>% select(model) %>% report()
my_mable %>% tidy()
my_mable %>% glance()
```

We can still use the tidyverse verbs to explore the mable.

2.3 Forecasting

To forecast, we pass the mable to the forecast function.

```
## If data is monthly, we can forecast 3 years ahead in the following ways
my_mable %>%
        forecast(n = "3 years")

my_mable %>%
        forecast(n = 12)
```

The result is a `fable`, a forecast table with point forecasts and distributions. The `.mean` is the point forecast. To get model details, use the augment function.

```
my_mable %>%
        select(model) %>%
        augment()
```

From this detailed table, we can plot the forecasts versus actual values.

Residuals should be a white noise process- the ACF should lie between the blue lines in the ACF plot.

Forecast intervals can be extracted using the `hilo()` function, specifying the level argument to control coverage.

```
my_mable %>%
        select(model1) %>%
        augment() %>%
        hilo(level = c(80, 95))
```

Point forecasts give an illusion of certainty. They are always faulty. Always report the uncertainty around a point estimate. Probabilistic forecasts are always the best - gives all the scenarios likely to materialize- the most likely, the best case, and the worst case.

- Check `desired service level` to determine the actual amount to purchase from the distribution.
- It is important to monitor the forecasts periodically- what you missed and why?

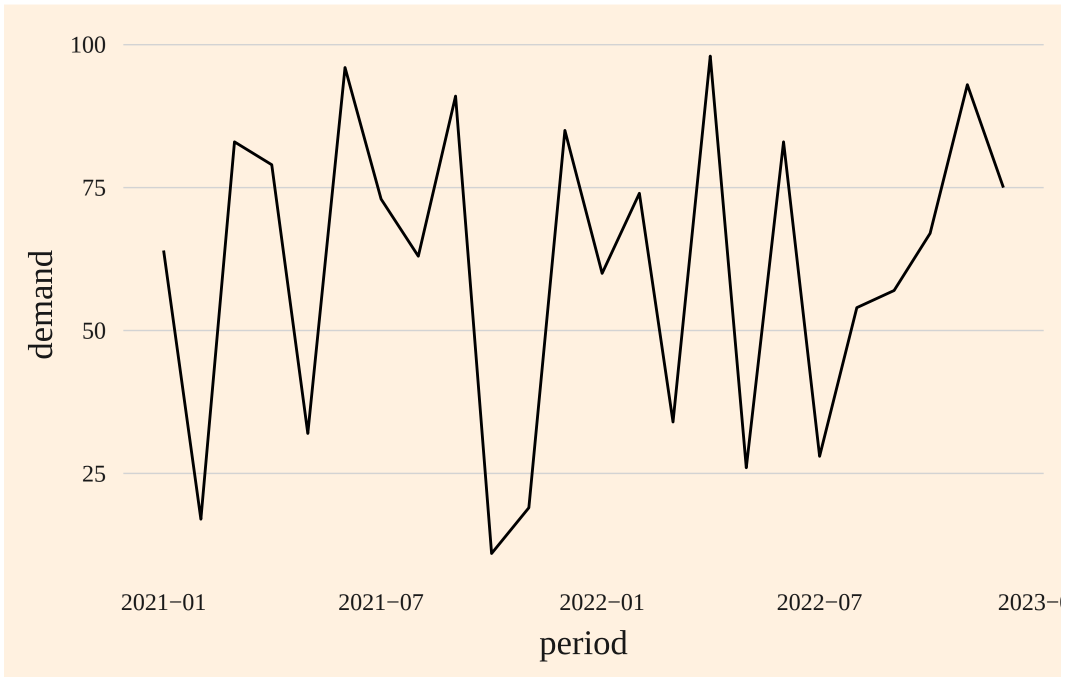
2.4 Exercise

```
library(tsibble)
library(fable)
library(tidyverse)
library(readxl)
library(janitor)


random_data <- readxl::read_xlsx("forecasts_random_data.xlsx") %>%
        clean_names()

############################################
head(random_data)
```

```
# A tibble: 6 x 2
  period              demand
  <dttm>               <dbl>
1 2021-01-01 00:00:00     64
2 2021-01-02 00:00:00     17
```

```
3 2021-01-03 00:00:00     83
4 2021-01-04 00:00:00     79
5 2021-01-05 00:00:00     32
6 2021-01-06 00:00:00     96
```

```
random_data_tsibble <- random_data %>%
        mutate(period = lubridate::ydm(period)) %>%
        as_tsibble(index = period)

##########################################
random_data_tsibble %>%
        ggplot(mapping = aes(x = period, y = demand)) +
        geom_line() +
        artyfarty::theme_ft()
```



2.5 Forecasting example

1. Split the data into training and testing set.
2. Forecast accuracy is computed on new data.

3. Model evaluation metrics- MAE, MSE, MAPE, RMSE, MASE and so on. If you have many time series, use a scale independent metric. MAPE penalizes overestimate more than underestimate.

2.6 Creating Rolling Training Sets

These are cross validation datasets from the training set.

- split the data: use filter_index.
- Create cross validation sets: use stretch_tssibble.
- Pass the cross validated data to model function.
- The accuracy function in fable is used to calculate forecast accuracy.
- Winkler score asseses how good our prediction intervals are. It takes into account the coverage and width of the intervals. Lower Winklers scores, the better.
- the function, `gg_tsresiduals` -
- Check all the assumptions:
- residuals uncorrelated.
- Residual unbiased, the mean close to zero.
- Other useful properties: - Residuals have constant variance (homoscedascicity). - Residuals are normally distributed.