Walmart's weekly sales

Print-friendly PDF version



Figure 1: Image generated by OpenAI's DALL-E

You have just been hired as consultants by Walmart, the global retail giant. Your challenge is to create a forecasting model for the weekly sales of 45 of Walmart's stores. Accurate forecasts drive decisions on inventory, staffing, marketing and financial planning.

You will work through the whole forecasting process: explore the data and the effect of holidays, temperature and the state of the economy, fit and compare benchmark, ETS, ARIMA and dynamic regression models, and finally forecast all the stores. One methodological point of this lab is the distinction between a *bottom-up* approach, where you build a model for each store and add the forecasts to a forecast of the total sales, and a *top-down* approach, where you add the data to total sales first, and build one model for that. There are pros and cons with both, and at the end we compare them.

Objective: *create a forecasting model for the individual stores' weekly sales and for the total weekly sales.*

This lab uses most of the course: time series graphics, the forecaster's toolbox, ETS, regression, ARIMA and dynamic regression. Work in groups of two or three. You should have time for the first eight exercises in a session, and the rest can be done at home.

The data

```
install.packages("remotes")
remotes::install_github("holleland/ban430data")   # to get the walmart data
```

```
library(fpp3)
library(ban430data)
data(walmart)
```

The data (from Kaggle, under a CC0 licence) are weekly sales for 45 Walmart stores, from week 5 of 2010 to week 43 of 2012. The columns are

Column	Description
`Store`	the store number (1 to 45)
`Date`	the date of the Friday of the sales week
`Weekly_Sales`	sales for the store in the week
`Holiday_Flag`	1 if the week contains a special holiday, otherwise 0
`Temperature`	temperature on the day of sale
`Fuel_Price`	the cost of fuel in the region
`CPI`	the prevailing consumer price index
`Unemployment`	the prevailing unemployment rate

The holidays that are flagged are the Super Bowl (12 Feb 2010, 11 Feb 2011, 10 Feb 2012), Labour Day (10 Sep 2010, 9 Sep 2011, 7 Sep 2012), Thanksgiving (26 Nov 2010, 25 Nov 2011) and Christmas (31 Dec 2010, 30 Dec 2011).

Cheat sheet: functions for this lab

Task	Syntax
Time index with weekly data	`yearweek(Date)`, `as_tsibble(index = week, key = Store)`
Seasonal plot, features	`gg_season(y, period = "year")`, `features(y, unitroot_kpss)`
Differencing	`difference(y)` (first), `difference(y, 52)` (seasonal)
Benchmarks	`NAIVE(y)`, `SNAIVE(y)`, `MEAN(y)`, `RW(y ~ drift())`
ETS and ARIMA	`ETS(log(y))`, `ARIMA(log(y))`, `ARIMA(log(y) ~ pdq(1, 1, 1) + PDQ(0, 1, 0))`
Dynamic regression	`ARIMA(log(y) ~ x + PDQ(0, 0, 0))`; forecasts need `new_data` with `x`
Forecast and accuracy	`forecast(fit, new_data = test)`, `accuracy(fc, full_data)`
Aggregate over the stores	`index_by(week) %>% summarise(total = sum(Weekly_Sales))`

Exercise 1: Make a tsibble

Load the data and convert it to a tsibble. It should have a well-formatted index column of class *yearweek*, and the key should be *Store*.

Code hint

```
walmart_ts <- walmart %>%
  mutate(week = yearweek(Date)) %>%
  select(-Date) %>%
  as_tsibble(index = week, key = Store)
```

Exercise 2: A first look at the data

Do a preliminary analysis. You may focus on stores 1–5 to keep the figures simple. Discuss whether

- trend or seasonality is present,
- there is any visible effect of the holidays, the temperature, the fuel price or the unemployment,
- you would prefer to inflation adjust the time series (what does the `CPI` column look like?),
- it is relevant to apply a mathematical transformation of the data.

Use relevant graphics, and justify your decisions.

Code hint

```
walmart_ts %>% filter(Store %in% 1:5) %>% autoplot(Weekly_Sales)
walmart_ts %>% filter(Store %in% 1:5) %>%
  gg_season(Weekly_Sales, period = "year")

# the holiday weeks in red
walmart_ts %>% filter(Store %in% 1:5) %>%
  ggplot(aes(x = week, y = Weekly_Sales)) + geom_line() +
  geom_point(data = . %>% filter(Holiday_Flag == 1), colour = "red") +
  facet_wrap(~ Store, scales = "free_y")

# sales against the other variables
walmart_ts %>% filter(Store %in% 1:5) %>% as_tibble() %>%
  pivot_longer(c(Temperature, Fuel_Price, Unemployment, CPI)) %>%
  ggplot(aes(x = value, y = Weekly_Sales)) + geom_point(size = 0.6) +
  facet_grid(Store ~ name, scales = "free")
```

Q: Which of the holidays seem to matter for the sales? Is there any reason to expect that a single holiday dummy will work well?

Exercise 3: Train and test sets

Make a train-test split of the data. We will use the last 20 weeks as the test set.

Code hint

```
train <- walmart_ts %>% filter(week <= max(week) - 20)
test  <- walmart_ts %>% filter(week > max(week) - 20)
```

In the following exercises 4–7, we focus on Walmart store number 3.

Exercise 4: Differencing

Consider a seasonal difference or a first difference of the logarithmic weekly sales. Do a KPSS test for this setup (you may also use the `unitroot_ndiffs()` and `unitroot_nsdiffs()` functions).

Code hint

```
store3 <- train %>% filter(Store == 3)
store3_test <- test %>% filter(Store == 3)
store3 %>% features(log(Weekly_Sales), list(unitroot_kpss, unitroot_ndiffs, unitroot_nsdi
store3 %>% gg_tsdisplay(difference(log(Weekly_Sales), 52), plot_type = "partial")
```

Q: What are the downsides of applying a seasonal difference in this particular situation? Plotting the seasonally differenced series may help. (How many weeks do we have in the training set, and how long is the seasonal period?)

Exercise 5: Choose an ARIMA model

Plot the ACF and PACF of the (possibly differenced) logarithmic weekly sales, and suggest p and q for an ARIMA(p,d,q) model.

Code hint

```
store3 %>% gg_tsdisplay(difference(difference(log(Weekly_Sales), 52)), plot_type = "parti
```

Exercise 6: Compare models

Fit the following models to store 3:

- benchmark methods of your choice (naive, seasonal naive, mean or drift),
- ETS,
- ARIMA (the one from exercise 5, and the automatic),
- a dynamic regression model with a holiday effect.

Select the best model by the RMSE on the test set. Assess whether the model assumptions are reasonable (both the required and the "nice-to-have" assumptions). You may add other candidate models as you like.

Code hint

```
fit3 <- store3 %>%
  model(
    naive = NAIVE(Weekly_Sales),
    snaive = SNAIVE(Weekly_Sales),
    mean = MEAN(Weekly_Sales),
    drift = RW(Weekly_Sales ~ drift()),
    ets = ETS(log(Weekly_Sales)),
    arima = ARIMA(log(Weekly_Sales)),
    # your own p, d, q:
    arima_manual = ARIMA(log(Weekly_Sales) ~ 0 + pdq(1, 1, 1) + PDQ(0, 1, 0)),
    dynreg = ARIMA(log(Weekly_Sales) ~ Holiday_Flag + PDQ(0, 0, 0)))
fc3 <- fit3 %>% forecast(new_data = store3_test)
fc3 %>% accuracy(walmart_ts) %>% select(.model, RMSE, MAE, MAPE) %>% arrange(RMSE)

fit3 %>% select(arima) %>% gg_tsresiduals()
```

Q: `ETS()` gives a warning about the seasonal period. What does it say, and what does that mean for the ETS model? Is the best model on the test set better than the mean method? Is the dynamic regression with the holiday dummy better than the ARIMA without it, and why (not)?

Exercise 7: Ex-ante or ex-post?

The dynamic regression model uses a predictor. Discuss briefly whether this is an ex-ante or an ex-post forecast model. Would your conclusion change if the price of fuel was also included as a predictor?

Exercise 8: All the stores

Fit the same candidate models (as in exercise 6) to *all* the stores, using only the automatic selection procedures. For each store, choose the model with the lowest RMSE on the test set. Plot the forecasts for the test data for a selection of the stores (for example stores 15–20).

Code hint

```
fit_all <- train %>%
  model(
    snaive = SNAIVE(Weekly_Sales),
    ets = ETS(log(Weekly_Sales)),
    arima = ARIMA(log(Weekly_Sales)),
    dynreg = ARIMA(log(Weekly_Sales) ~ Holiday_Flag + PDQ(0, 0, 0)))
```

```
fc_all <- fit_all %>% forecast(new_data = test)
acc_all <- fc_all %>% accuracy(walmart_ts) %>% select(Store, .model, RMSE)
best <- acc_all %>% group_by(Store) %>% slice_min(RMSE, n = 1) %>% ungroup()
table(best$.model)

fc_all %>%
  inner_join(best %>% select(Store, .model), by = c("Store", ".model")) %>%
  filter(Store %in% 15:20) %>%
  autoplot(walmart_ts %>% filter(Store %in% 15:20, week >= yearweek("2012 W01")), level =
  facet_wrap(~ Store, scales = "free_y")
```

Exercise 9: Bottom-up forecast of the total

Based on the best models from exercise 8, make a forecast of the total weekly sales in the test set, for all the stores together. You may focus on the point forecasts. *Extra challenge:* get the prediction intervals right (optional; the forecast errors of the stores are correlated, so you cannot just add the variances).

Code hint

```
bottom_up <- fc_all %>%
  inner_join(best %>% select(Store, .model), by = c("Store", ".model")) %>%
  as_tibble() %>%
  group_by(week) %>%
  summarise(forecast = sum(.mean))
```

Exercise 10: Top-down, and the comparison

Exercise 9 is a *bottom-up* approach. Another approach is to build a model for the total sales of all the stores. Add the training and test data to total sales, build a (non-benchmark) model of your choice, and compare it with the bottom-up forecast from exercise 9. Which approach do you prefer?

Code hint

```
total <- walmart_ts %>% index_by(week) %>% summarise(total = sum(Weekly_Sales))
total_train <- total %>% filter(week <= max(week) - 20)
total_test  <- total %>% filter(week > max(week) - 20)
fit_tot <- total_train %>% model(snaive = SNAIVE(total), ets = ETS(log(total)), arima = A
fc_tot <- fit_tot %>% forecast(new_data = total_test)
fc_tot %>% accuracy(total) %>% select(.model, RMSE, MAPE)

# the RMSE of the bottom-up forecast
bu <- left_join(bottom_up, as_tibble(total_test), by = "week")
sqrt(mean((bu$total - bu$forecast)^2))
```

Q: Which approach gave the best forecast of the total? Is the comparison fair? (Think about how we chose the model for each store in exercise 8. What happens to the comparison

if we choose the models differently, for example by using the same method for all stores?) What are the advantages of the top-down approach, and what do we lose?

Exercise 11: The final forecast

Plot the final forecast for the total sales together with the test data, with a short comment. If you did not manage to do exercise 9 or 10, plot another forecast.