Time series regression: Bergen traffic (lab)

Print-friendly PDF version

The data

Forecast hourly traffic through Bergen's *Danmarksplass* intersection — a real, live-updating case with a primary series, an external predictor, and a calendar effect all in one place. The traffic counts come from two sensors run by Statens vegvesen's Trafikkdata API, extended all the way through today, plus a Norwegian public holiday calendar that's *computed* rather than downloaded (so it never goes stale either).

```
install.packages("remotes")
remotes::install_github("holleland/ban430data")
```

```
library(fpp3)
library(ban430data)
data(regressionlab)
```

Two objects come in. The two traffic sensors arrive **pre-merged** into one table — you'll spend today's time on modelling and evaluation, not on rejoining raw series by datetime.

`traffic_danmarksplass` — one row per hour, 2018 → today

Column	Contains
`datetime`	date-time of the hour (Europe/Oslo)
`vehicles`	vehicles counted at Danmarksplass ("ved ladestasjon") in that hour
`moberg`	vehicles counted at Moberg v/Lekven — Halhjem ferry-arrival traffic, a candidate external predictor — in that hour

`holidays_no` — one row per Norwegian public holiday, 2018–2027

Column	Contains
`date`	the calendar date of the holiday

Cheat sheet: `TSLM()` syntax

Term	Syntax
Weekly seasonal dummies	`season(period = "week")`
Fourier terms	`fourier(K = ..., period = "week")`
An extra 0/1 or numeric predictor	add the column directly, e.g. `+ is_holiday`
Fit	`model(name = TSLM(log(y) ~ ...))`
Forecast a known window	`forecast(fit, new_data = test)`
Accuracy vs. the actuals	`accuracy(fc, full_data)`

Part 1: Shared foundation

Task 1: Build the tsibble

Code hint

```
traffic_ts <- traffic_danmarksplass %>%
  as_tsibble(index = datetime) %>%
  fill_gaps()
```

Task 2: Time plot

Plot the whole series. What does the trend look like over 8+ years? Where do you see the strongest seasonality — daily, weekly, or yearly?

Task 3: Choosing a transformation

Look back at your Task 2 plot — does the variance of `vehicles` look roughly constant over time, or does it grow with the level? Use `guerrero` to find the Box-Cox transformation that stabilises it best, and compare it to a plain log.

Code hint

```
traffic_ts %>% features(vehicles, features = guerrero)

lambda <- traffic_ts %>% features(vehicles, features = guerrero) %>% pull(lambda_guerrero)

traffic_ts %>% autoplot(box_cox(vehicles, lambda))
```

Q: How close is `lambda` to 0 (a log transform)? Given how close (or not) it is, would you expect Box-Cox to meaningfully beat a plain log once you get to modelling — or is log probably "close enough"? Keep `lambda` around; Task 1 in Part 2 asks you to check.

Task 4: A new variable, derived from a date

`holidays_no` is just a list of dates. Turn it into a 0/1 `is_holiday` column on `traffic_ts`, marking every hour that falls on a Norwegian public holiday.

Code hint

```
traffic_ts <- traffic_ts %>%
  mutate(is_holiday = as.numeric(as_date(datetime) %in% holidays_no$date))
```

Task 5: Train/test split

Use this exact split — everyone needs the same one, or the leaderboard comparison won't make sense:

- **train**: everything before 2025-12-20
- **test**: 2025-12-20 up to (not including) 2026-01-20 — a four-week window spanning Christmas, Boxing Day *and* New Year's Day

Code hint

```
train <- traffic_ts %>% filter(datetime < ymd("2025-12-20"))
test  <- traffic_ts %>% filter(datetime >= ymd("2025-12-20"), datetime < ymd("2026-01-20"))
```

Task 6: A closer look

Facet the **training** data by weekday, with hour of day on the x-axis and one line per date (colour by date). Where do weekends visibly differ from weekdays? Do any weekday dates look like they're "acting like" a weekend?

Code hint

```
train %>%
  as_tibble() %>%
  mutate(wday = wday(datetime, label = TRUE), hour = hour(datetime), date = as_date(datetime)
  ggplot(aes(x = hour, y = vehicles, group = date, colour = date)) +
  geom_line(alpha = 0.3) +
  facet_wrap(~wday)
```

Keep that plot in mind — it's the visual version of the problem every task below is trying to solve numerically.

Part 2: Build every model

For each task, build the model described, forecast the test window, and write down its **test-set RMSE and MASE**.

```
fc <- fit %>% forecast(new_data = test)
fc %>% accuracy(traffic_ts) %>% select(.model, RMSE, MASE)
```

Task 1: Weekly seasonal dummies

Fit `TSLM()` with `season(period = "week")` — that's 168 dummy predictors, one per hour-of-week. Try it on the raw response, the log-transformed response, and the Box-Cox transform from Part 1 (`lambda`).

Code hint

```
fit_A <- train %>%
  model(
    raw    = TSLM(vehicles ~ season(period = "week")),
    log    = TSLM(log(vehicles) ~ season(period = "week")),
    boxcox = TSLM(box_cox(vehicles, lambda) ~ season(period = "week"))
  )
```

Q: Which of the three fits best on the test set? Did Box-Cox actually beat log, or was your Part 1 hunch about `lambda` being "close enough" to 0 correct?

Task 2: Fourier terms

Same idea as Task 1, but replace the 168 dummies with a handful of `fourier(K = ..., period = "week")` terms. Try a couple of values of `K`.

Code hint

```
fit_B <- train %>%
  model(
    fourier_10 = TSLM(log(vehicles) ~ fourier(K = 10, period = "week")),
    fourier_40 = TSLM(log(vehicles) ~ fourier(K = 40, period = "week"))
  )
glance(fit_B) %>% select(.model, AIC, AICc, BIC)
```

Q: How does `AICc` change with `K`? How few parameters can you get away with before test accuracy visibly gets worse?

Task 3: Holidays

Add the `is_holiday` column you built in Part 1 to a weekly-seasonal model.

Code hint

```
fit_C <- train %>%
  model(
    no_holiday   = TSLM(log(vehicles) ~ season(period = "week")),
    with_holiday = TSLM(log(vehicles) ~ season(period = "week") + is_holiday)
  )
fc_C <- fit_C %>% forecast(new_data = test)
```

Q: Zoom in on 24 Dec – 2 Jan specifically. How badly does the `no_holiday` model miss on those days? What other dates might deserve their own dummy, beyond what `holidays_no` gives you (Christmas Eve? New Year's Eve? the day *before* a holiday)?

Task 4: External predictor (the ferry-traffic sensor)

`moberg` (Halhjem ferry-arrival traffic) is already a column on `train`/`test` — add it as a predictor.

Code hint

```
fit_D <- train %>% model(ext = TSLM(log(vehicles) ~ season(period = "week") + log(1 + moberg
fc_D  <- fit_D %>% forecast(new_data = test)
```

Q: This forecast only works because `test` already *has* real `moberg` values for every test hour. In a genuine forecasting situation, would you have those? What kind of forecast is this and what could you do instead?

Part 3: Combine and submit

1. Look back at your four RMSE/MASE numbers. Which idea helped most — the log transform, Fourier vs. dummies, the holiday term, the external predictor?
2. Fit **one combined model** that borrows the best pieces. Two kinds of submission are both welcome on the leaderboard — just label which one you're making:
   - **ex-ante**: only uses information a real forecaster would actually have at the time (e.g. Fourier + the holiday dummy — *not* `moberg`'s real test-window values)
   - **ex-post**: allowed to use the real observed test-window values (e.g. `moberg` as-is). Not a genuine forecast, but a useful ceiling to compare against
3. Compare against a seasonal naïve benchmark:

```
train %>%
  model(snaive = SNAIVE(vehicles)) %>%
  forecast(new_data = test) %>%
  accuracy(traffic_ts) %>%
  select(.model, RMSE, MASE)
```

4. Submit your combined model's score to the live leaderboard, labelled:

```
leaderboard_score(fc_final, team = "Your team name", type = "ex-ante")
```

If you have time, submit **both**: your best genuine ex-ante model, and an ex-post version that's allowed to peek at `moberg`. The gap between your own two scores is worth having a real number for.

What's leaderboard_score()?

A small helper that scores a forecast the same way `accuracy()` does throughout this lab, and prints a submission card you can read off:

```
leaderboard_score <- function(fc, team, type = c("ex-ante", "ex-post")) {
  type <- match.arg(type)
  acc <- fc %>% accuracy(traffic_ts)
  if (nrow(acc) != 1) {
    stop("fc has ", nrow(acc), " models (", paste(acc$.model, collapse = ", "),
         ") - filter to one with filter(.model == \"...\") first.")
  }
  cat("================================\n")
  cat(" LEADERBOARD SUBMISSION\n")
  cat("================================\n")
  cat(" Team:", team, "\n")
  cat(" Type:", type, "\n")
  cat(" RMSE:", round(acc$RMSE, 1), "\n")
  cat(" MASE:", round(acc$MASE, 3), "\n")
  cat("================================\n")
  invisible(acc)
}
```

Run this once at the top of your session, then call `leaderboard_score(fc_final, "Your team name", type = "ex-ante")` (or `"ex-post"`) on any forecast object — Task 1–4 or your combined one. Report the printed RMSE, MASE and type in the form below. Resubmitting the same team name **and** type replaces that entry, so submit early and keep improving — a team can hold one ex-ante *and* one ex-post entry at the same time.

Nobody's checking that your combined model is *novel* — reusing one task's model as-is is a perfectly fine first submission. The point is having a number on the board to improve on.

The live leaderboard only works on the web version of this page — open it there to see the current standings and submit a score.

#	Team	RMSE	MASE

#	Team	RMSE	MASE

Wrap-up discussion

Once a few teams are on the board:

- Who's leading the **ex-ante** board, and with what kind of model? Was it the most complex one, or did a simpler combination do just as well?
- If your pair submitted both: how big was your own gap between ex-ante and ex-post? Is everyone's gap roughly the same size, or does it vary a lot by team?
- Did the holiday term help everyone by roughly the same amount? What about the external predictor, for those who tried an ex-post version?
- Revisit the ex-ante/ex-post question from Task 4 — `moberg` only helps once you're allowed to see it. What would a genuinely ex-ante use of `moberg` look like (a lag? forecasting `moberg` itself first?), and would it actually help?
- What would you try next if you had another hour? (residual diagnostics on the leading model, a longer forecast horizon, a different external predictor, a stronger benchmark)