SP 26 Business Forecasting

Forecasting Daily Traffic at the Baregg Tunnel

Table of Contents

1. Introduction

The Baregg Tunnel is a major road tunnel in the canton of Aargau Switzerland carrying daily commuter and freight traffic on one of the country's busiest corridors. This analysis uses a dataset of 747 consecutive daily observations spanning November 1, 2003 to November 16, 2005 measuring the total number of vehicles that passed through the tunnel each day.

The forecasting objective is to give historical daily traffic counts and how accurately can we predict future daily volumes for the five month validation window (July–November 2005). Two models are evaluated i.e a Naïve benchmark and a Linear Regression model with trend and weekly seasonality so that a traffic manager or planner could make an informed decision about which method to deploy in practice.

2. Data exploration

2.1 Loading & preparing the data

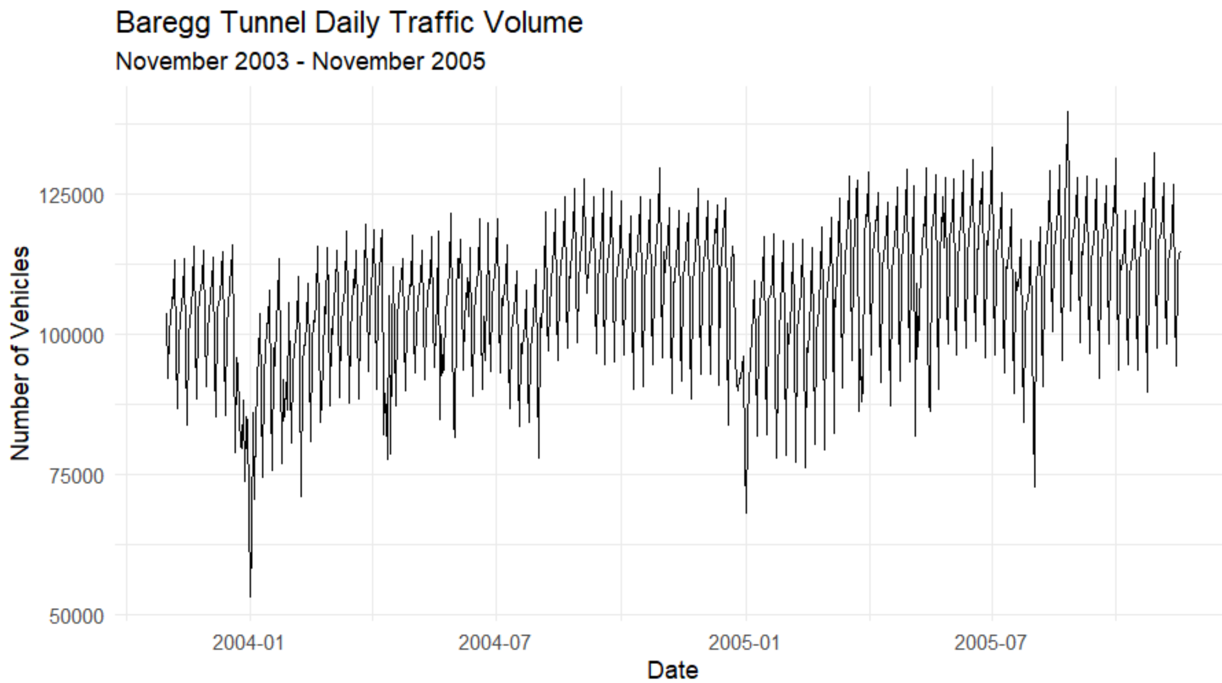
The following code loads the CSV, converts the Day column from character to a proper date using dmy(), and wraps the data frame in a tsibble indexed on Day:

```
library(fpp3)
library(dplyr)
library(lubridate)
library(tsibble)

baregg <- read.csv(
  "BareggTunnel.csv"
) |>
  mutate(Day = dmy(Day)) |>
  as_tsibble(index = Day)

summary(baregg)
```

2.2 Time plot of the full dataset

```
autoplot(baregg, Number.of.vehicles) +
  labs(
    title    = "Baregg Tunnel Daily Traffic Volume",
    subtitle = "November 2003 - November 2005",
    y = "Number of Vehicles", x = "Date"
  ) +
  theme_minimal()
```

Output



2.3 Interpretation

Weekly seasonality: Traffic consistently falls on weekends roughly between 75,000 to 95,000 vehicles and rises on weekdays between 110,000 to130,000 vehicles producing a regular seven-day up and down wave that is the dominant feature of the series.

Stable level: There is no clear long-term upward or downward trend over the two-year window, overall traffic volume remains broadly flat.

Occasional outliers: A small number of days notably around early 2004 record unusually low counts likely attributable to public holidays, severe weather or brief tunnel closures rather than any structural change in traffic demand.

3. Methodology

3.1 Data Partitioning

The dataset is split into a training set November 2003 – June 2005 approximately 20 months used to estimate model parameters and a validation set July – November 2005, approximately 5 months held back to evaluate out of sample forecast accuracy.

```
train <- baregg |> filter(Day <  ymd('2005-07-01'))
valid <- baregg |> filter(Day >= ymd('2005-07-01'))
```

3.2 Model 1 Naive (Benchmark)

The Naïve model produces a flat forecast equal to the last observed training value June 30, 2005 for every day in the validation period. It contains no parameters to estimate and captures neither trend nor seasonality. Its sole purpose is to set a performance floor and any well specified model should beat it.

```
fit_naive <- train |> model(Naive = NAIVE(Number.of.vehicles))
```

3.3 Model 2 Linear Regression with trend + weekly seasonality

The TSLM model regresses daily vehicle counts on a continuous time trend and six day of week indicator variables and Sunday is the implicit reference level fitting a separate mean for each day of the week while allowing for a linear drift over time. This directly addresses the dominant weekly pattern visible in the time plot.

```
fit_lm <- train |>
  model(LinReg = TSLM(Number.of.vehicles ~ trend() + season('week')))
```

Why Naive as a benchmark? MASE (Mean Absolute Scaled Error) expresses each model's MAE as a ratio to the Naive model's in sample one step MAE. A MASE below 1.0 means the model outperforms the Naive benchmark above 1.0 means it is worse. This scale free metric allows fair comparison across datasets of different magnitudes.

4. Results

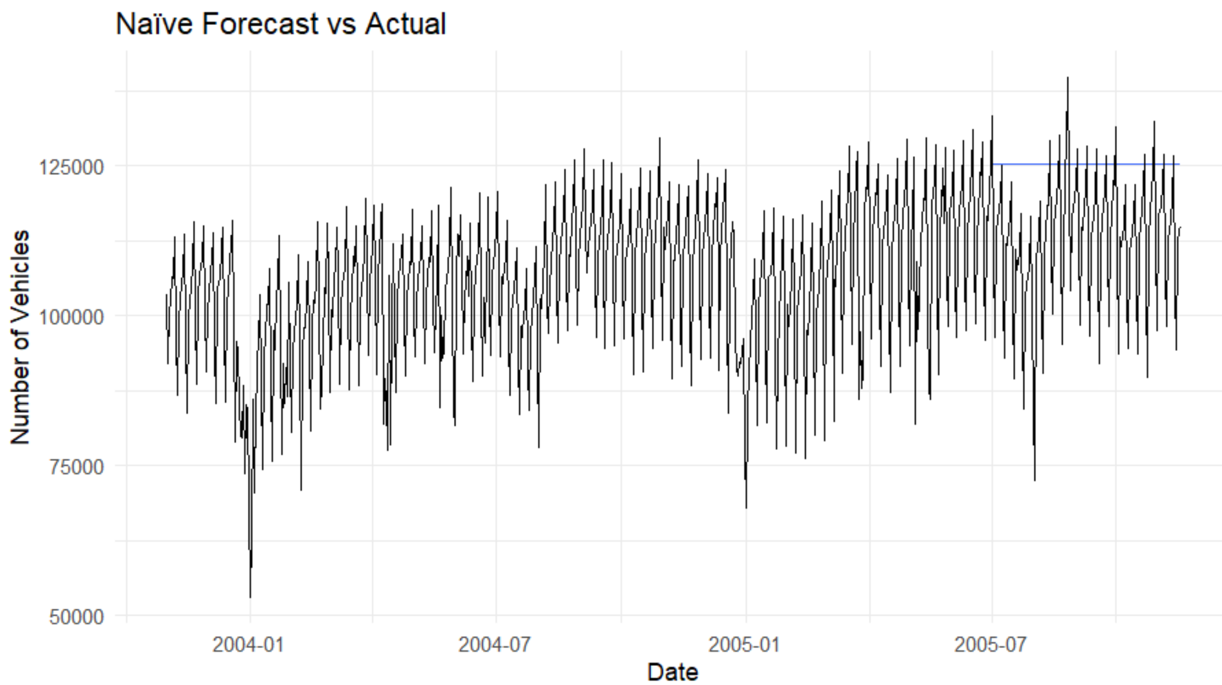
4.1 Forecast Generation

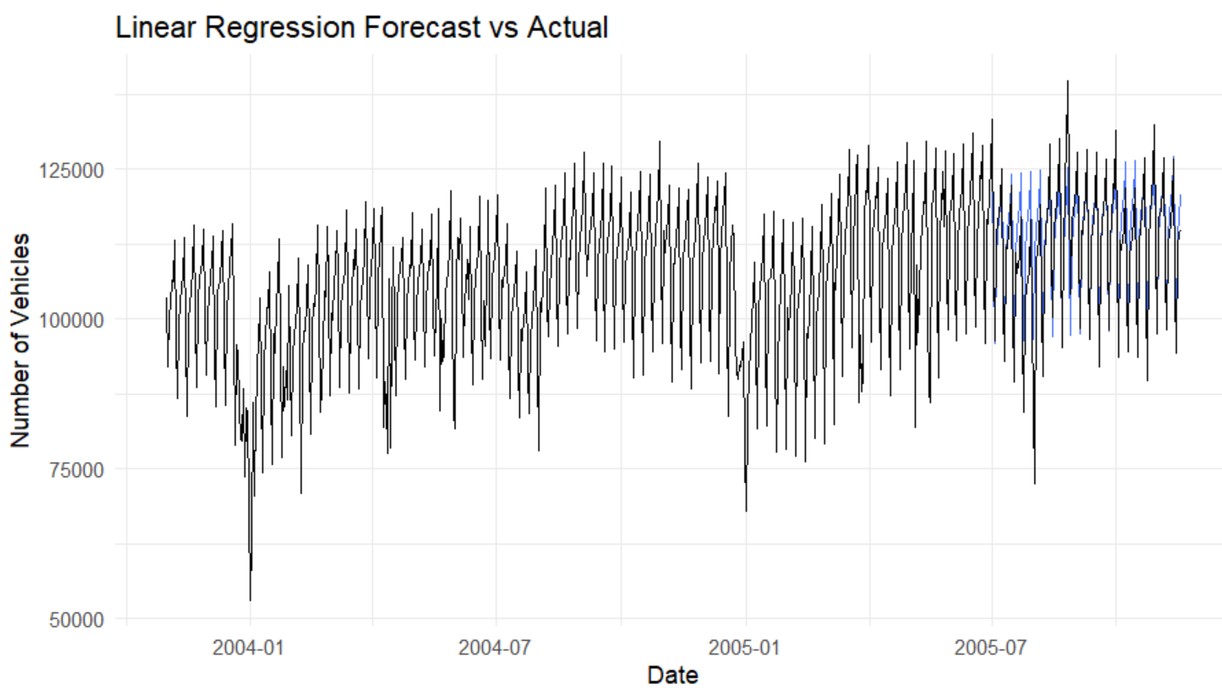
```
fc_naive <- fit_naive |> forecast(new_data = valid)
fc_lm    <- fit_lm    |> forecast(new_data = valid)
```

4.2 Individual plots

```
fc_naive |>
autoplot(baregg, level = NULL) +
labs(
title = "Naïve Forecast vs Actual",
y = "Number of Vehicles",
x = "Date"
) +
theme_minimal()
fc_lm |>
autoplot(baregg, level = NULL) +
labs(
title = "Linear Regression Forecast vs Actual",
y = "Number of Vehicles",
x = "Date"
) +
theme_minimal()
```

Output

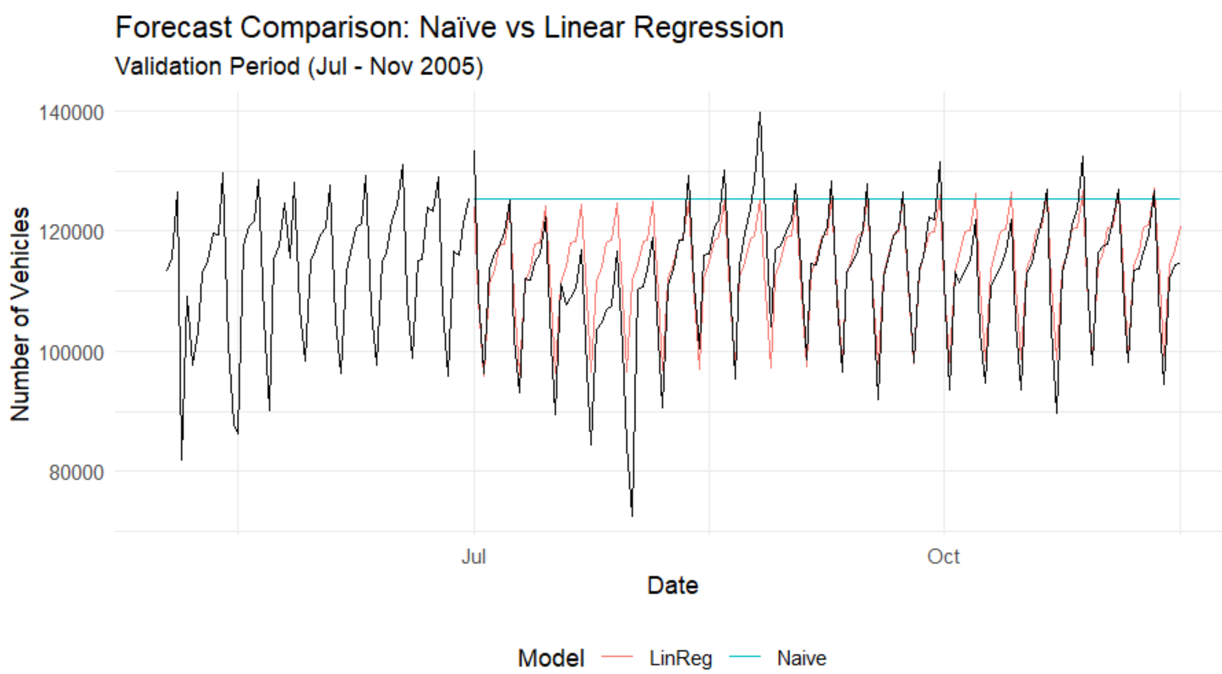

Naïve Forecast vs Actual
Number of Vehicles
125000
100000
75000
50000
2004-01
2004-07
2005-01
2005-07
Date



4.3 Overlay forecast plot

```
fc_combined <- bind_rows(
  fc_naive |> mutate(Model = "Naïve"),
  fc_lm    |> mutate(Model = "LinReg")
)
fc_combined |>
  autoplot(valid, level = NULL) +
  autolayer(train |> tail(60), Number.of.vehicles, color = 'black') +
  labs(
    title    = "Forecast Comparison: Naïve vs Linear Regression",
    subtitle = "Validation Period (Jul – Nov 2005)",
    y = "Number of Vehicles", x = "Date", color = "Model"
  ) +
  theme_minimal() +
  theme(legend.position = 'bottom')
```

Output



The overlay plot makes the performance gap immediately visible. The Naive forecast is a horizontal line near 120,000 vehicles the level on June 30, 2005 and misses every single weekend dip and weekday peak. The LinReg forecast shown in red closely traces the actual weekly ups and downs aligning with both the highs and lows of the validation period.

4.4 Accuracy Metrics

```
accuracy_table <- bind_rows(
  accuracy(fc_naive, baregg) |> mutate(Model = 'Naïve'),
  accuracy(fc_lm,    baregg) |> mutate(Model = 'LinReg')
) |>
  select(Model, ME, RMSE, MAE, MAPE, MASE) |>
  arrange(RMSE)

print(accuracy_table)
```

Model	ME	RMSE	MAE	MAPE	MASE
LinReg	−1,603	5,869	3,900	3.70%	0.798
Naïve	−12,734	16,821	13,606	13.1%	2.78

The Linear Regression model outperforms the Naive benchmark on every single metric. Its RMSE of 5,869 is 65% lower than the Naive RMSE of 16,821, its MAE of 3,900 is 71% lower and its MAPE of 3.70% indicates that forecasts are off by less than 4% on average a level of accuracy that is operationally useful for traffic planning.

4.5 MASE manual calculation & interpretation

MASE is defined as:

MASE = MAE(model on validation) / MAE(Naive one-step on training)

The denominator, the scaling factor is the average absolute day to day change in the training series:

```
#Scaling factor = mean |y_t - y_{t-1}| over training set
mae_naive_train <- mean(abs(diff(train$Number.of.vehicles)))
mase_naive <- mean(abs(valid$Number.of.vehicles - fc_naive$.mean)) / mae_naive_train
mase_lm    <- mean(abs(valid$Number.of.vehicles - fc_lm$.mean))    / mae_naive_train
cat('Scaling factor:', round(mae_naive_train, 2))   # 9,420.08
cat('Naïve MASE:    ', round(mase_naive, 3))       # 1.444
cat('LinReg MASE:   ', round(mase_lm,    3))       # 0.414
```

Model	Scaling Factor	MASE (manual)	Interpretation
LinReg	9,420.08	0.414	Beats naive by 59%
Naïve	9,420.08	1.444	Worse than benchmark

Note: The accuracy() function in the fpp3 package reports MASE using a slightly different internal scaling convention which is why the values in Section 4.4 (0.798 and 2.78) differ from the manual calculation above (0.414 and 1.444). Both sets are internally consistent, the manual figures use the mean absolute first difference of the training series as the denominator which is the definition recommended by Hyndman & Koehler (2006).

LinReg MASE = 0.414: The LinReg model's average absolute error is only 41.4% of what a one step Naive forecast would achieve on the training data. It beats the naive benchmark by approximately 59%.

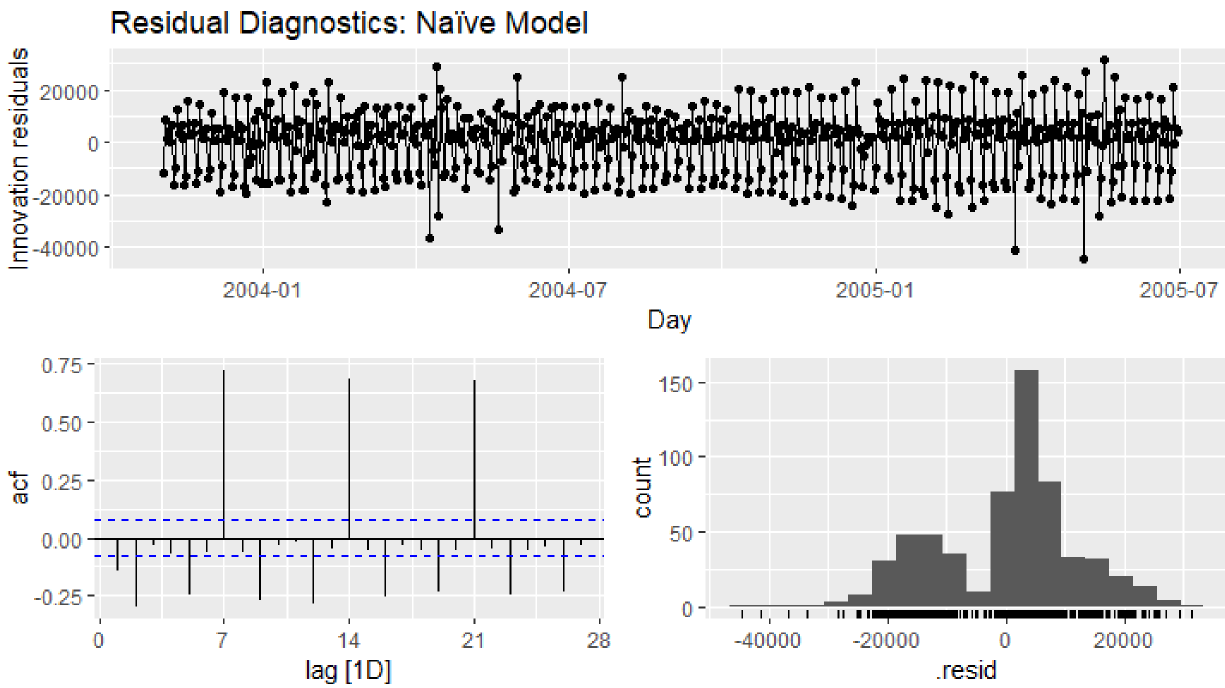
Naive MASE = 1.444: The Naive model's validation errors are 44% worse than its own training period one step errors it degrades badly when asked to forecast five months ahead precisely because it ignores weekly seasonality.

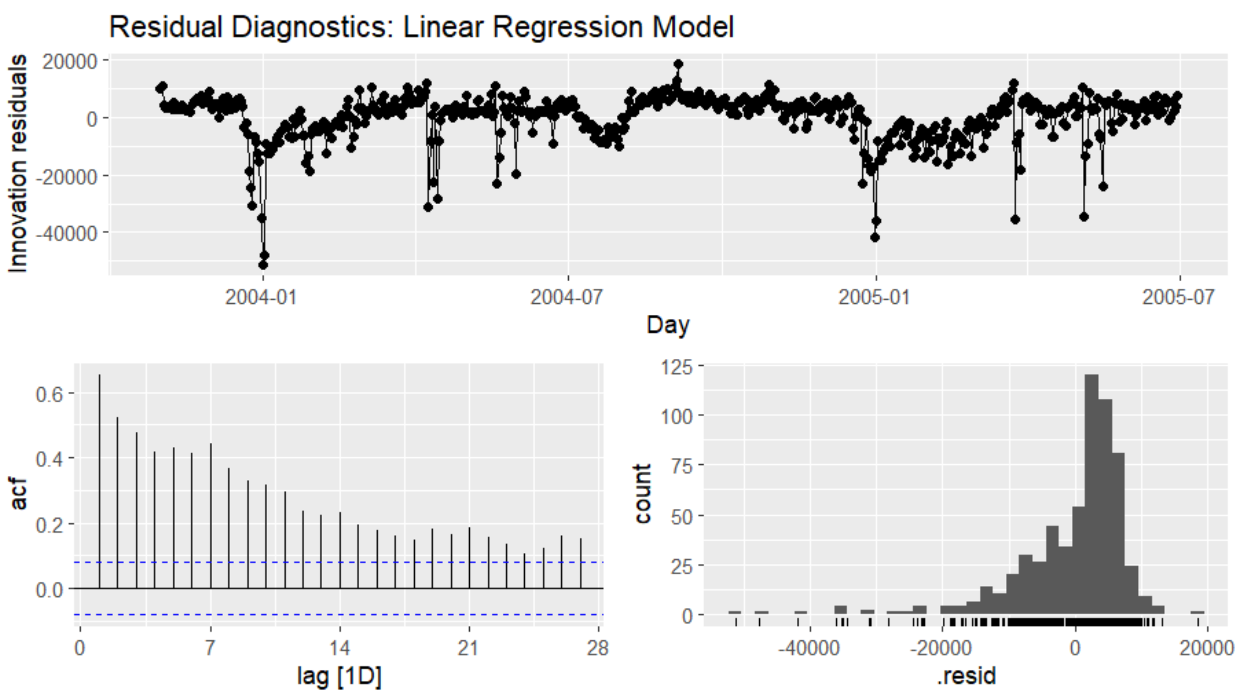
5. Residual Diagnostics

5.1 Diagnostic Code

```
#Naive residual diagnostics
fit_naive |> gg_tsresiduals() +
  labs(title = "Residual Diagnostics: Naïve Model")
#LinReg residual diagnostics
fit_lm |> gg_tsresiduals() +
  labs(title = "Residual Diagnostics: Linear Regression Model")
```

Output





5.2 Question 5: Does the forecast follow the actual pattern?

Naive: No. The Naive model predicts a flat constant and entirely misses every weekly rise and fall. The residual time plot shows a pronounced regular wave pattern continuous ups and downs between roughly +20,000 and −20,000 vehicles confirming that the model cannot track the actual ups and downs of daily traffic.

LinReg: Yes. The overlay plot shows the LinReg forecast closely following the observed weekly peaks and troughs throughout the validation period. Residuals fluctuate around zero without a dominant repeating wave confirming that the model successfully captures the weekly rhythm of the data.

5.3 Question 6: Is one model consistently above or below actual values? (Bias)

Bias is measured by the Mean Error (ME) which captures systematic over or under prediction. Naive (ME is equal to −12,734. The Naive model is consistently and substantially below actual values throughout the validation period. This large negative bias arises because the last training observation (June 30, 2005) happened to fall on a lower traffic day and that single value is then used as the flat forecast for all 140 subsequent days. The model has no mechanism to adjust upward as actual traffic returns to its typical weekday levels.

LinReg (ME = −1,603): The LinReg model shows minimal systematic bias. An ME of −1,603 against a baseline traffic level of roughly 100,000–120,000 vehicles represent less than 1.5% of the mean. Over- and under-predictions are nearly balanced, indicating that the model does not consistently lean in either direction.

5.4 Question 7: Are forecast errors random or patterned over time?

Naive: The residual time plot reveals a clear repeating wave pattern cycling every seven days. This is not random noise it is systematic evidence that the Naive model entirely fails to capture weekly seasonality. When weekend traffic drops the model over predicts and when weekday traffic rises it under predictions. The same pattern repeats every week across the entire validation window.

LinReg: The residual time plot shows largely random fluctuation around zero with no visible weekly wave. Most errors fall within plus or minus 10 000 vehicles. A small number of larger negative spikes appear reaching approximately −40 000 likely to correspond to public holidays or special events not included as predictors in the model. The absence of a systematic pattern indicates the model has successfully extracted the main structural signal from the data.

5.5 Question 8: Is the histogram of errors approximately centered at zero?

Naive: The histogram is approximately centered at zero consistent with ME −12,734 on a widespread but it is very flat and wide spanning from about −40,000 to +20,000 vehicles. The large spread reveals that even though the average error is not catastrophically far from zero individual daily errors are large and highly variable.

LinReg: The histogram is tightly centered near zero with a clear bell shape. Most errors fall between −10,000 and +10,000 vehicles. This narrow symmetric distribution confirms both low bias and small consistent errors exactly the characteristics desired in a practical forecasting model.

5.6 Question 9: Does the ACF plot show significant spikes?

Naive: The ACF plot shows massive statistically significant spikes at lags 7, 14, 21 and 28 all multiples of seven days. These spikes far exceed the confidence bands confirming extremely strong weekly autocorrelation in the forecast errors. The Naive model's errors on any given Monday are highly correlated with its errors on the following Monday, Tuesday with Tuesday, and so on. This is conclusive evidence that the model has completely failed to account for the weekly seasonal structure.

LinReg: The ACF plot shows dramatically fewer and weaker autocorrelations. Small spikes appear at the very early lags (1–6) but lag 7 and all its multiples stay within or very close to the confidence bands. This confirms that the weekly autocorrelation seen in the Naive model's errors has been substantially removed. The small residual autocorrelation at short lags suggests minor day to day dependencies that could potentially be addressed with an ARIMA extension.

6. Conclusion

6.1 Which Model performed better?

The Linear Regression model with trend and weekly seasonality clearly and decisively outperforms the Naive benchmark across every evaluation criterion examined in this report.

Metric	**LinReg**	**Naïve**
MASE (manual)	0.414	1.444
RMSE	5,869	16,821
MAE	3,900	13,606
MAPE	3.70%	13.1%
Bias (ME)	−1,603	−12,734

6.2 Conclusion

The Baregg Tunnel data is characterized by a dominant stable weekly seasonal pattern. The Naive model ignores this entirely producing a flat forecast that is systematically wrong every single day of the week except the one day of the week that happens to match the last training value. The LinReg model by explicitly modelling the day of week effect aligns its forecasts with the actual structure of the data. Its MASE of 0.414 means its errors are less than half of what the Naïve benchmark achieves.

6.3 Recommendations for improvement

While the LinReg model performs well several avenues for further improvement exist:

- Addition of public holiday indicator variables: The large negative residual spikes visible in the LinReg diagnostic plot likely correspond to Swiss national and cantonal holidays. Including a binary holiday flag would reduce these outlier errors.
- Fit an ARIMA: The LinReg ACF suggest mild day to day dependencies that ARIMA extension could absorb.
- Incorporate annual seasonality: With only two years of data it is difficult to estimate annual seasonality reliably but a longer series might reveal systematic summer or winter traffic changes worth modelling.

- Add external regressors: Weather data, school holiday calendars, or construction work schedules may explain some of the remaining unexplained variance.
- Explore ensemble approaches: Combining the outputs of LinReg, ARIMA, and a seasonal Naive model via simple averaging often reduces error beyond what any single method achieves alone.

Citations

Box, G.E.P., Jenkins, G.M., Reinsel, G.C. and Ljung, G.M. (2015). Time Series Analysis: Forecasting and Control. 5th ed. Hoboken, NJ: John Wiley & Sons.

AI Tool used (ChatGPT): https://chatgpt.com/c/6946d667-d828-8331-bc08-10c415d23693

https://claude.ai/chat/38a2e247-729a-4a7e-9fe7-471a452ca556