

Analysing Melbourne CBD pedestrian counts data with a focus on imputation methods

Gavin Chin

21 September 2017

Background

The Data: City of Melbourne Pedestrian Data

The City of Melbourne provides an open data platform to access council datasets, with the intention “to increase transparency, improve public services and support new economic and social initiatives”¹. This paper will focus primarily on the pedestrian data collected from pedestrian sensors placed throughout Melbourne’s CBD. In particular, we want to be able to model the pedestrian traffic at different locations in the CBD, and forecast the traffic. In addition, we aim to have an interactive, dynamic data visualisation of the model to help us to understand how Melbourne operates.

The dataset being investigated can be obtained from the official City of Melbourne’s open data². The pedestrian data is in the form of hourly pedestrian counts for 43 sensor locations. In total, there is 1425192 observations in the dataset, representing 26293 hours of data. It is available in .csv format, allowing for easy data import into R. We can also access the pedestrian data using the the `rwalkr` package by Earo Wang³. This package allows R to import the data from `data.melbourne.vic.gov.au` which is updated monthly, or from the data source which is used by `pedestrian.melbourne.vic.gov.au` which is updated daily.

Not even a single sensor location has complete data for the period between 1/1/2014 and 21/10/2017.

Imputation Literature Review Notes

Reviewing current research on imputation of missing values in traffic data, a paper on imputation of missing classified traffic data during winter season (Imputation of Missing Classified Traffic Data During Winter Season - *Hyuk-Jae Roh, Satish Sharma, Prasanta K. Sahu*) stated some methods which were found to be poor for imputation. The data they were attempting imputation for were traffic counters located on the highway network in Alberta, Canada with between 40% and 60% missing data.

¹from ‘About Melbourne Data’ page at <https://data.melbourne.vic.gov.au/about>

²the specific pedestrian data is available at <https://data.melbourne.vic.gov.au/Transport-Movement/Pedestrian-volume-updated-monthly-/b2ak-trbp>

³<https://github.com/earowang/rwalkr/>

Replacement with “good” historical values, or using historical average values was found to have resulted in very poor fit with high MARE/MARD (mean absolute relative error/difference). They did find that using a moving average worked the best out of all the heuristic methods used. The largest problem with heuristic methods, however, is the inherent inability to reflect sudden fluctuations/shocks during abnormal periods. With such a large proportion of missing values, pattern matching methods were investigated. This involves comparing the *study curve*, the pattern at the counter which is to be imputed, to candidate patterns (patterns at other locations).

For the purposes of that paper, it was found to be inappropriate with the data being analysed. However, unlike *Roh et al.* where patterns were compared to traffic volumes in different jurisdictions (large geographical distance), the geographic distances between the sensor locations in the Melbourne CBD pedestrian data are much smaller. Another difference is the randomness of the missing values, where the periods of missing data run longer in the Melbourne CBD pedestrian data compared to that in the Alberta traffic data. They proposed using non-parametric estimation methods with k-Nearest Neighbours for their imputation, although this is unlikely to work well with the lack of data at certain sensor locations in the Melbourne CBD Pedestrian data.

Methodology: Imputation of Missing Values

Basic GLM Approach

The first method used to impute missing values is to fit generalised linear models (GLMs) at each sensor location. Estimating a GLM with a Quasi-Poisson error distribution, where specify the model as:

$$E[\text{HourlyCounts}|\text{Sensor}, \text{HDay}, \text{Time}] = \exp(\mu_{\text{Sensor}, \text{HDay}, \text{Time}})$$

where $\mu_{\text{Sensor}, \text{HDay}, \text{Time}} \sim \text{Time} \times \text{HDay}$ (1)

Time is the time of the day, while **HDay** is the day of the week, with an additional factor level for public holidays. Both these variables are categorical/factors. We cannot treat time of the day as an integer or numeric because it does not have a linear relationship with counts.s

We use the Quasi-Poisson error distribution as opposed to normal/Gaussian errors due to the response variables being count data. The Poisson distribution allows for only non-negative integer values, while estimating a Quasi-Poisson model estimates an additional parameter for overdispersion. This allows more flexibility in the model by allowing the assumption that $E[Y] = \text{Var}[Y] = \mu$ to be relaxed. Instead, the Quasi-Poisson distribution allows for $\text{Var}[Y] = \theta\mu$.

While this works well with a small proportion of missing values, it also requires a large number of observations. Specifically, with this parameterisation, we have $23 + 7 + (23 \times 7) + 1 = 192$ coefficients to estimate. Another disadvantage of using this model is the lack of robustness to outliers due to the sensitivity of some parameters. This becomes particularly problematic at sensor locations which have been recently installed, where there is a lack of historical data.

Small-Large Split Approach

The proposed alternative approach to imputing the missing values in the data focuses on improving the models used at locations with a large proportion of missing values. A potential threshold value which could be used to class a sensor as having a “large” proportion of missing values vs a “small” proportion of missing values is 10%. At locations with a “small” proportion of missings, we can continue to use the GLM quasi-posson regression specified in (1). At the locations with a large proportion of missing values, we need to use information from neighbouring sensors.

The first issue which needs to be addressed is the potential of values which are actually missing due to sensor failure or malfunction having a zero count. This issue will cause bias in the estimates, as well as affect the classification of a sensor. Of course, we cannot simply classify all zero counts as `NA`, as true zero values are possible.

Table 1: Summary statistics of the sequence lengths of repeated values

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
2	2	2	16.98	2	19967

Instead, a simple check we can implement is to look for long sequences of zero counts. For example, we may want to classify any sequence of `Hourly_Counts = 0` running for longer than 24 hours as being considered `NA`. From the summary of lengths of repeated values (which are repeated at least once), we see that the distribution of the lengths are very positively skewed with a mean length of 16.98 hours of missing data. Classifying any sequence of repeated values, typically 0, which has a length greater than 6 as `NA` should allow true zero counts to avoid being misclassified as `NA`. We select the length of 6 hours as it would be unreasonable to assume any sensor location to have the exact same number of pedestrian counts over such a long period.

After we have replaced questionable zero values with the value `NA`, we can calculate the proportion of missing values for each sensor location. From these calculated proportion, we can classify sensors as either having a “large” or “small” proportion of missing values, where large is defined as $\text{NAprop}_{\text{sensor}} > \text{threshold}$ and $\text{threshold} = 0.1$.

Because the models for the sensors with a large proportion of missing values rely on having neighbouring sensors having complete cases (no missing values), we want to impute these values first. This will then allow us to have as much information to be available for the models which use neighbouring sensors to train on.

For sensors with a small proportion of missing values, we use a GLM quasipoisson model. To improve on the previous model used, we use the model:

$$E [\text{HourlyCounts} | \text{Sensor}, \text{Month}, \text{DayType}, \text{Time}] = \exp(\mu_{\text{Sensor}, \text{Month}, \text{DayType}, \text{Time}})$$

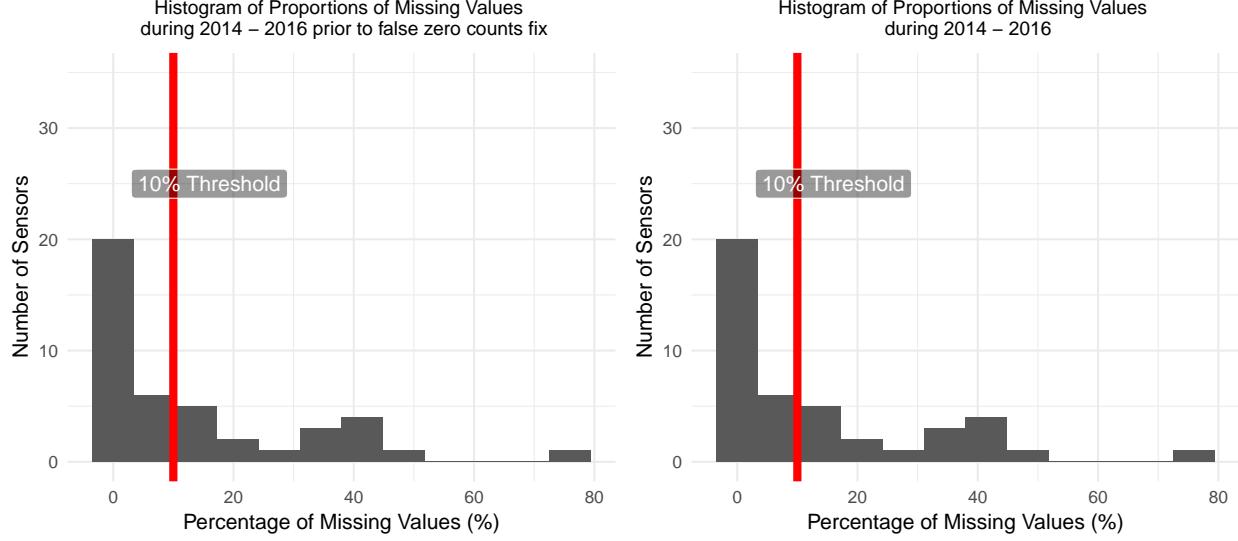


Figure 1: Comparison of distribution of proportion of missing values at each sensor is significantly different after correcting for false zero counts in the data

$$\text{where } \mu_{\text{Sensor}, \text{Month}, \text{DayType}, \text{Time}} \sim \text{Month} + \text{Time} \times \text{DayType} \quad (2)$$

Instead of `HDay`, we use `DayType` which classifies Tuesday, Wednesday and Thursday as a single factor level, Midweek. We do this as the daily patterns on these weekdays are similar. This reduces the number of parameters to estimate by 46. A result of this is we also have more robust estimates for the midweek days, being less sensitive to outliers. Because this model will be estimated for sensor locations with few missing values, we have enough data to add in month as an additive effect. This will help capture the annual seasonality.

Using these estimated models, we impute the missing values to produce complete data at all these sensor locations. For the locations with high proportions of missing values can be predicted using the now complete data from all the locations which had a small proportion of missing values. Using a threshold of 10%, this gives us 25 sensor locations to use.

For each of the sensor locations with high proportions of missings, we need to decide which sensor to use for prediction. A simple method of finding a geographical closest neighbour is to take the haversine/great-circle distance between the sensor to be imputed and all the possible candidates.

Using the two geographically closest sensors with complete cases, we use another GLM specified as:

$$E [\text{HourlyCounts}_{\text{sensor}} | \text{Time}, \text{HourlyCounts}_{\text{neighbours}}] = \mu_{\text{Time}, \text{HourlyCounts}_{\text{neighbours}}}$$

$$\text{where } \mu_{\text{Time}, \text{HourlyCounts}_{\text{neighbours}}} \sim \text{Time} \times \text{HourlyCounts}_{\text{neighbour 1}}^{SC} + \text{Time} \times \text{HourlyCounts}_{\text{neighbour 2}}^{SC}$$

We would expect that the amount of people passing through the neighbouring sensors will be a strong predictor of the pedestrian counts as they are likely to also pass through. Using

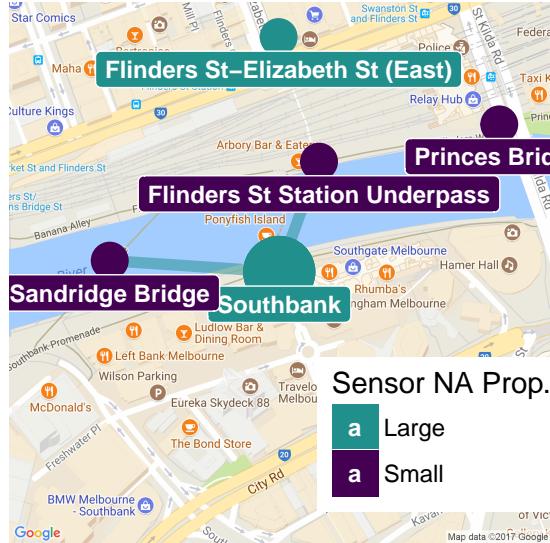
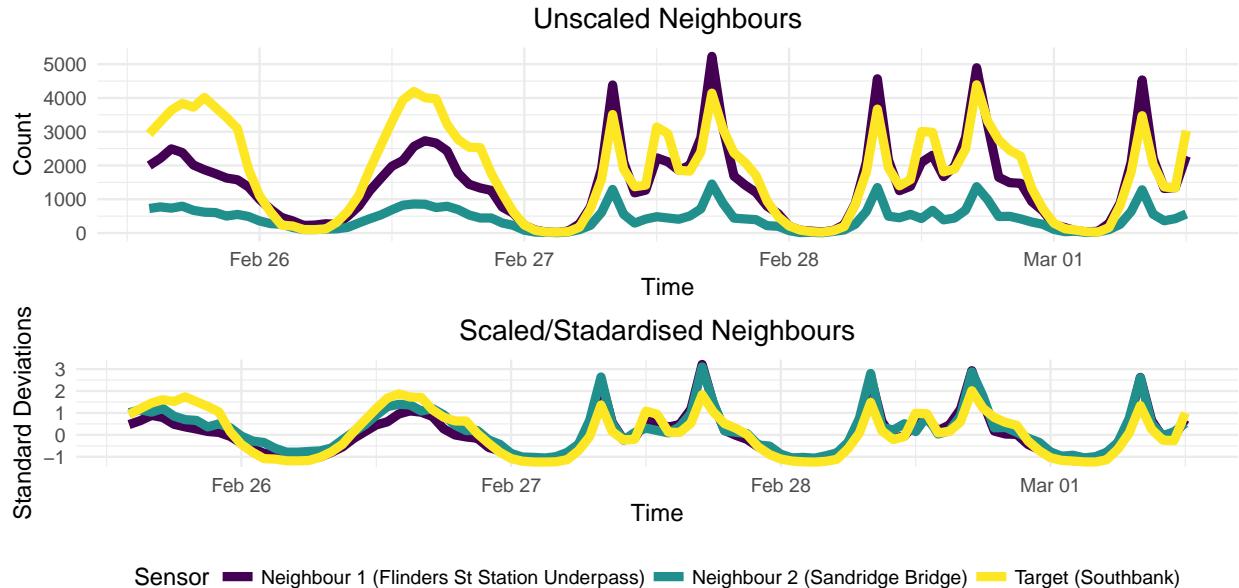


Figure 2: Map showing the algorithm selecting Flinders St Station Underpass and Sandridge Bridge as neighbours to be used to predict the pedestrian counts at Southbank

geographical neighbours, we can capture rare events effectively. Any large shocks to the counts at the neighbouring sensors will have an effect on the counts to be imputed.



In order to use a mixture of the patterns from two different sensors, we need to use the scaled counts at the neighbouring sensors to avoid the magnitude of the counts having an effect on the predicted counts. We scale the counts by standardising to $\mu = 0, \sigma^2 = 1$. Because the covariance with neighbouring sensors may vary over time, we also add an interaction term between the neighbouring sensor counts and the time of the day.

Imputation Results

Firstly, we evaluate the initial model used to impute:

$$\text{where } \mu_{\text{Sensor}, \text{HDay}, \text{Time}} \sim \text{Time} \times \text{HDay} \quad (1)$$

Note, this is also after the adjustments made to treat suspiciously long sequences of 0 values as NA in the actual counts. The simple GLM model is trained before this correction, while the improved model is trained on the adjusted data.

For the purposes of evaluating the imputation method at a location with a small proportion of missing values, we will first look at the counts from Southern Cross Station. At this sensor, the proportion of missing values (adjusted) is 0.02 Firstly, we evaluate the goodness of fit of the predictions made within the training period (01/01/2014 to 31/12/2016).

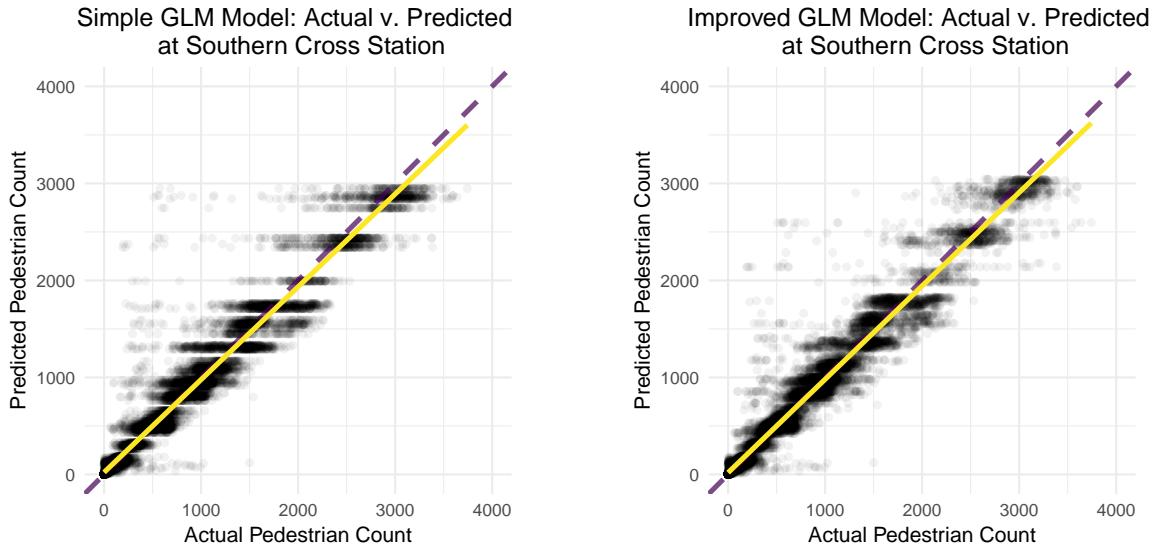


Figure 3: A plot of the in-sample Actual vs Predicted pedestrian counts at Southern Cross Station (sensor location with small proportion of missing values) for the simple GLM model and the improved GLM model. The dashed line is $x = y$, representing a reference for perfect fit, while solid line is the linear fit of the points.

Visually, we can see that the fitted values are good estimates as the relationship between the actuals vs fitted is very close to 1:1. This is seen when we perform a least squares estimate of $\widehat{\text{Fitted}}_t = \hat{\beta}_0 + \hat{\beta}_1 \text{Actual}_t$, and the estimated $\hat{\beta}_0$ is close to 0 and $\hat{\beta}_1$ is close to 1.

Here we see that the GLM model with only time and date based variables work well for fitting small periods of missing data.

However, if we try to use this same model specification at sensor locations with large proportions of missing values, the fit is poor. We use the sensor at Australia on Collins, where the proportion of missing values is 41.96%.

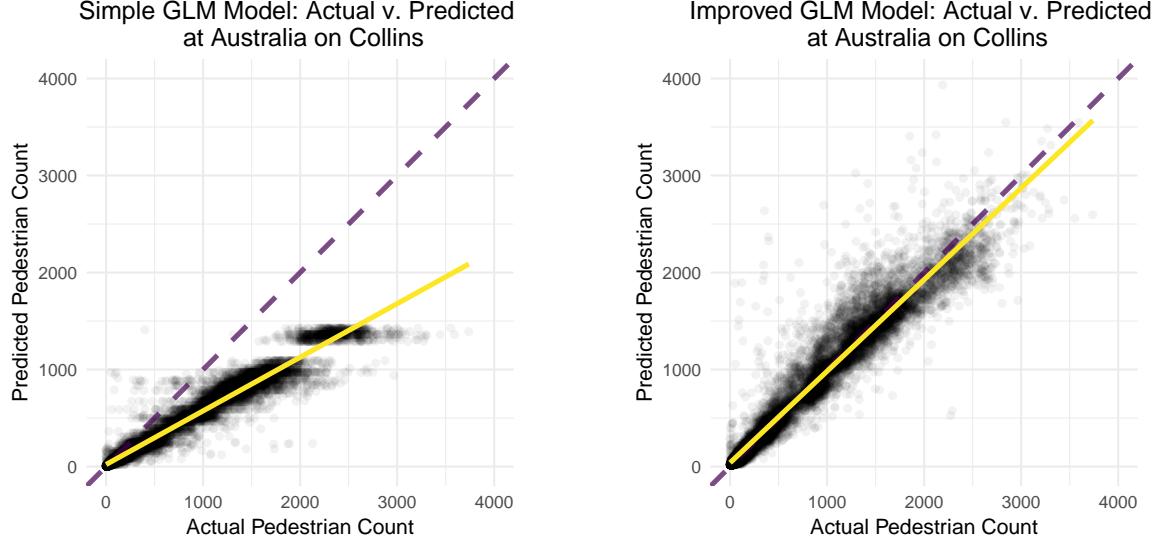


Figure 4: A plot of the in-sample Actual vs Predicted pedestrian counts at Australia on Collins (sensor location with large proportion of missing values) for the simple GLM model and the improved GLM model. The dashed line is $x = y$, representing a reference for perfect fit, while solid line is the linear fit of the points.

We can see that the simple model has bias caused by the false zero counts, resulting in the predicted values to be underestimated. This is shown by the slope of the fitted line on the actual counts against predicted counts being < 1 , emphasising the importance of the first step taken in the algorithm of the improved model to class the false zero counts as missing values.

The criterion we will use to measure goodness of fit is MARE. It is calculated by:

$$MARE_{sensor} = \frac{1}{T^2} \frac{\sum_{t=1}^T |\widehat{\text{HourlyCounts}}_{sensor,t} - \text{HourlyCounts}_{sensor,t}|}{\sum_{t=1}^T \text{HourlyCounts}_{sensor,t}}$$

We calculate a separate MARE for predictions in-sample (in the training set) and the predictions out of sample (test set of 2017 data). This will help to identify overfitting of the data. We use MARE as it will be a comparable measure between sensor locations as well as comparable between models.

While it is not very clear when comparing the distribution of in-sample MARE between imputation models, when we compare the in-sample MARE at each location we can see that the majority of locations had better in-sample fit with the improved model using the algorithm. We do see that some locations did have worse in-sample fit, but overall we see improvement with 34 having better fit with the improved model over the simple model.

Focusing on the date 04/07/2014, which has 3 hours of missing data at Southern Cross Station:

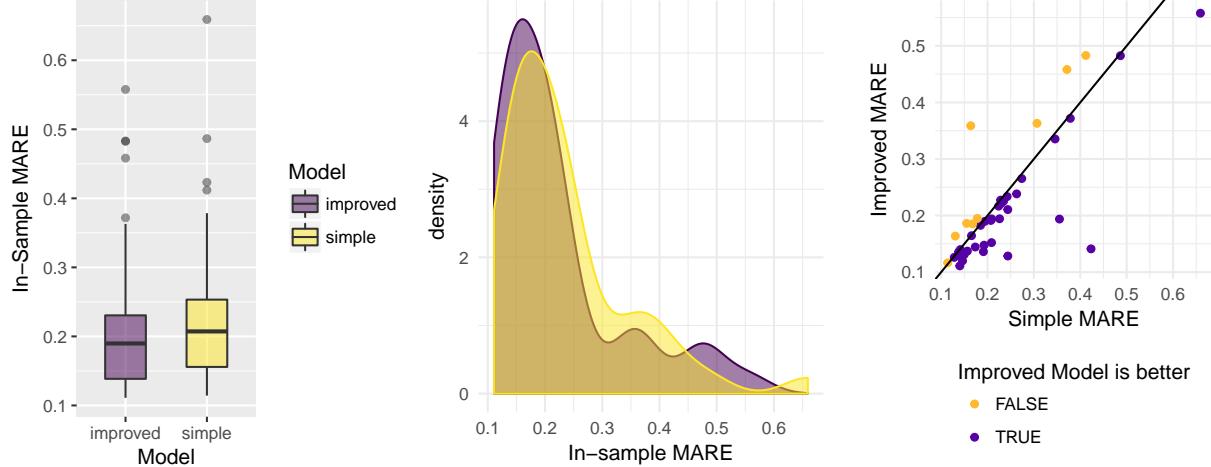


Figure 5: Distribution of in-sample (2014-2016 data) MARE by imputation model

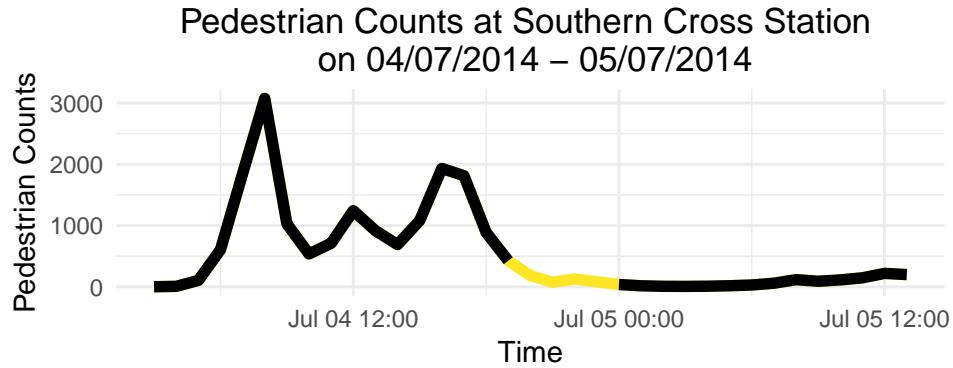


Figure 6: A plot of the pedestrian counts at Southern Cross Station. The black line is the observed counts, where the yellow line is the imputed counts.

More importantly, we can evaluate the performance of the model on the out of sample observations (2017 data) as cross validation. Due to potential false zero counts in the test data sourced from `rwalkr::walk_melb()`, we need to again apply the correction by long sequences of repeated values to be able to properly evaluate the goodness of fit of out-of-sample observations. The method used for this is identical to that used on the training set in the algorithm, replacing sequences of length 6 hours or longer with NA.

Again, we start by comparing the fit at a sensor location with a small proportion of missing values, Melbourne Central:

We find that the out-of-sample predictions of at Melbourne Central by the simple model has a tendency to underestimate the pedestrian counts. By comparison, the predictions made by the improved imputation model does not underestimate the counts as much. Comparing the out-of-sample MARE between models at Melbourne Central, MARE decreases from 21.43% to 18.55% with the use of the improved model.

At a sensor location with a large proportion of missing values, Collins Place (South), we

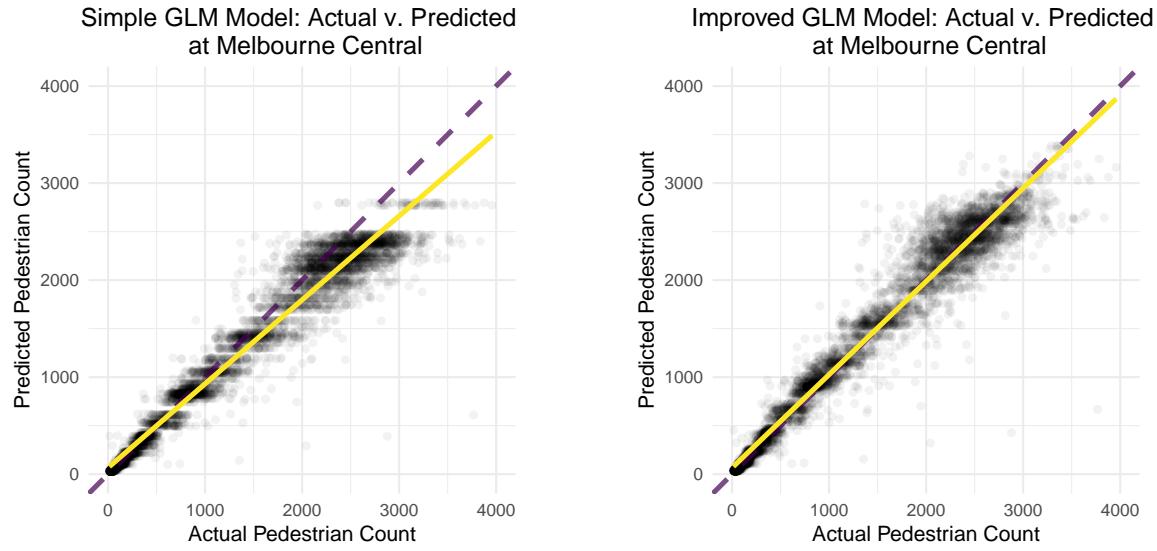


Figure 7: A plot of the out-of-sample (2017) Actual vs Predicted pedestrian counts at Melbourne Central (sensor location with small proportion of missing values) for the simple GLM model and the improved GLM model. The dashed line is $x = y$, representing a reference for perfect fit, while solid line is the linear fit of the points.

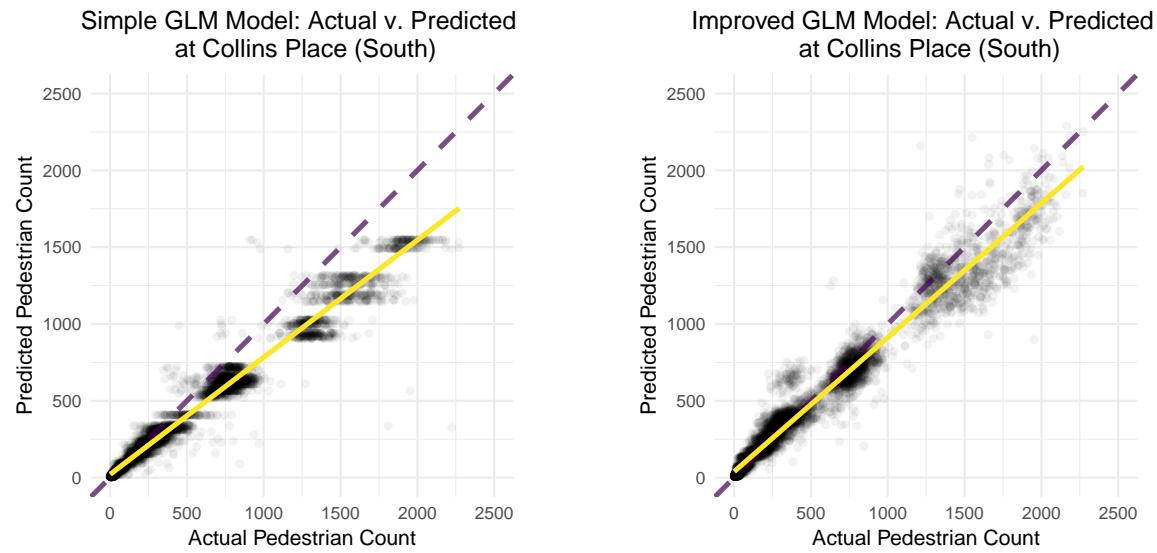


Figure 8: A plot of the out-of-sample (2017) Actual vs Predicted pedestrian counts at Collins Place (South) (sensor location with large proportion of missing values) for the simple GLM model and the improved GLM model. The dashed line is $x = y$, representing a reference for perfect fit, while solid line is the linear fit of the points.

again see a tendency to underestimate counts with the simple imputation model. The improved imputation method reduces the underestimation of the predictions and provides better predictions. The MARE is greatly improved, where it decreases from 21.39% to 14.65% with the use of the improved model.

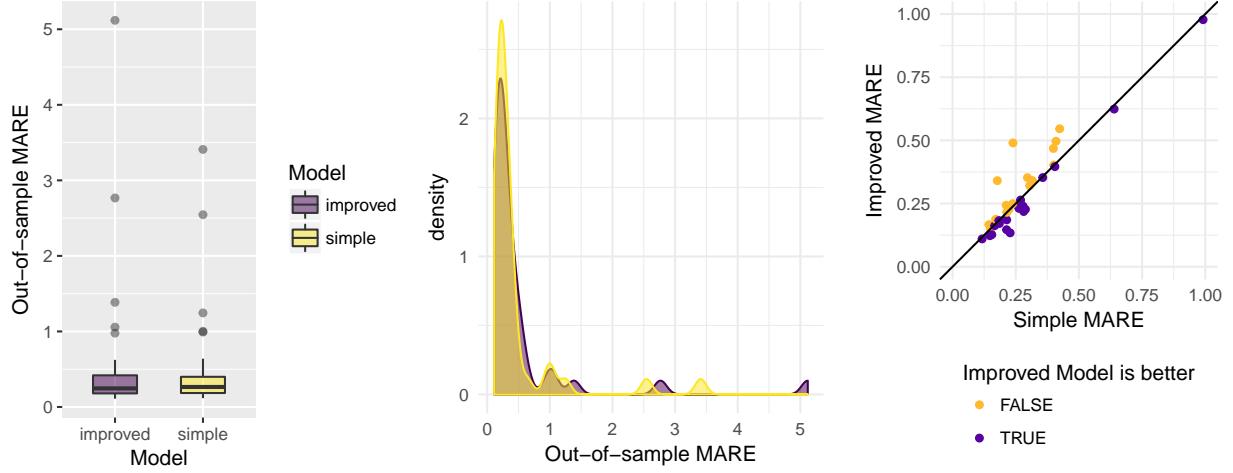


Figure 9: Distribution of out-of-sample (2017 data) MARE by imputation model, excluding Spencer St-Collins St (North)/(South) and Flinders St-Swanston St (West) sensors

Spencer St-Collins St (North)/(South) and Flinders St-Swanston St (West) was omitted from the the plots above as the out-of-sample MARE is extremely high at 2584.12%, 3278.74% and 3042.22% respectively.

As these sensor locations were both classified as locations with a large proportion of missing values, it would appear that the relationship between these locations and their neighbouring sensors has changed. This is because the in-saple MARE at these sensor locations were quite acceptable at 13.62%, 21.06% and 11.65% respectively.

The plot of the average counts by time of day and type of day and data set shows that this location saw an increase in counts on weekdays which did not occur at the neghbouring sensors. We also see that the pattern is slightly different during weekdays, with a higher afternoon/evening peak. These types of change in pedestrian traffic behaviour is hard to capture in our imputation model, especially when its neighbours did not have a similar change. While we can add a year variable to allow for year to year growth, the short period of training data doesn't make this appropriate at this stage.

Another way we can perform cross-validation is to remove data at a sensor, and evaluate the predictions made. This will allow us to evaluate the imputation model's performance within sample, which is the primary concern with them imputation process.

We select a sensor which has a small proportion of missing values, Bourke Street Mall (South), which has 99.981% of data available. As observed in the data, missing data is present in different ways: random, short periods of missing data (no more than 6 hours) or long periods of missing data (multiple months). We will simulate both types of missingness at this sensor location.

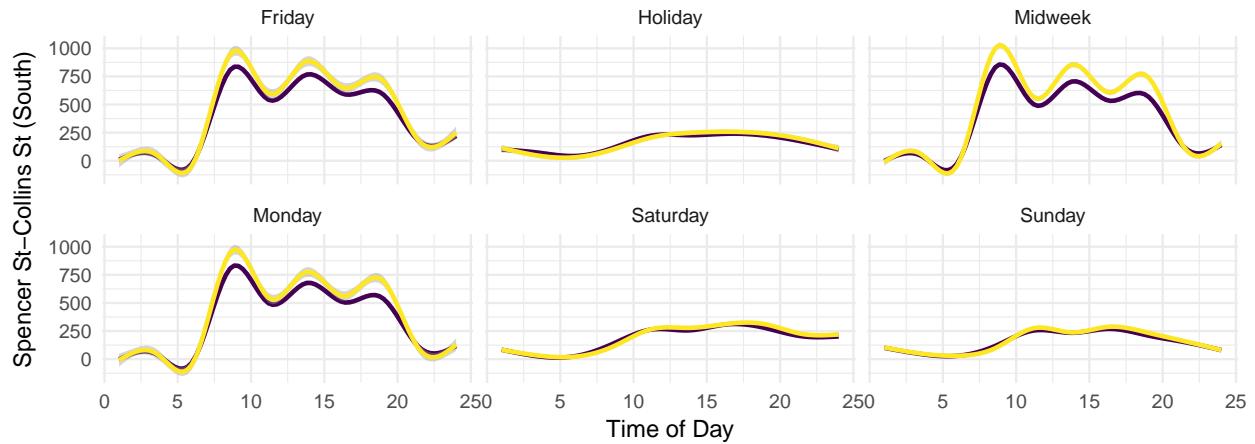


Figure 10: Average pedestrian counts by time of day and type of day (DayType) at Spencer St-Collins St (South) for training data (dark line) and test data (light line)

First, we remove randomly remove data such that 20% of the data is missing, then estimate the imputation model. Using this model, we calculate the MARE within the training data.

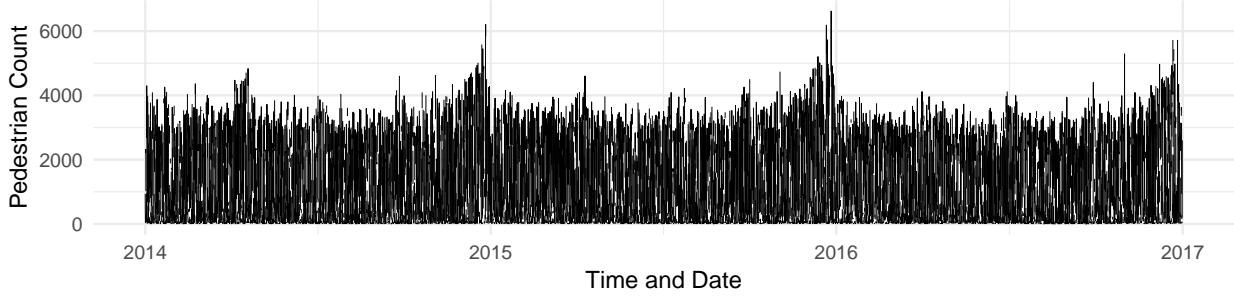


Figure 11: Plot of simulated missingness method 1 at Bourke Street Mall (South). 20% of the observations are removed from the traning data for the imputation model at random.

With random hours of missing data, we have very good predictions by using neighbouring sensors with a low MARE of 11.04%. Unfortunately, this type of missingness is not consistent with what we observe at most sensors with large proportions of missing data.

Instead, we need to remove 20% of the data in a single period to simulate what we typically observe. We take a random date and remove 2629 hours of data before and after this date. Then we proceed to estimate the imputation model again and calculate the MARE.

The MARE of the estimated model is 11.06%, which is similar to what was obtained when the missing values were random. This demonstrates that the model quite well when we remove 20% of the observations from the training data.

Removing 50% of the data using the same method, we find that the in-sample MARE is not affected very much by the significantly large proportion of missing training data where it is only 11.26%.

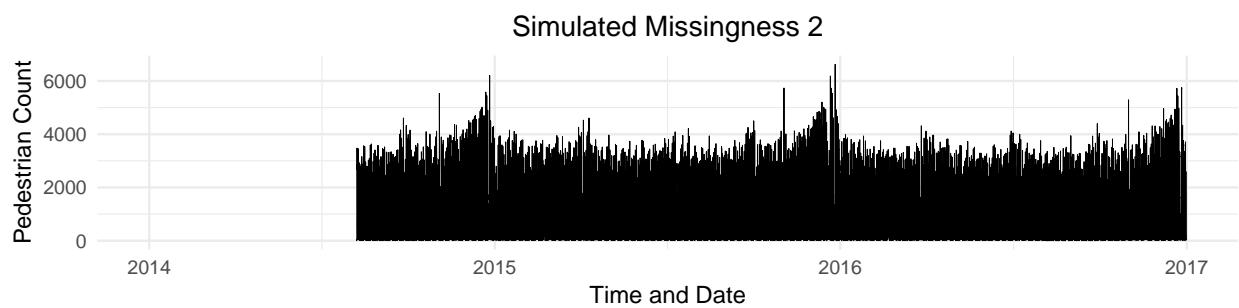


Figure 12: Plot of simulated missingness method 2 at Bourke Street Mall (South) with 20% data missing

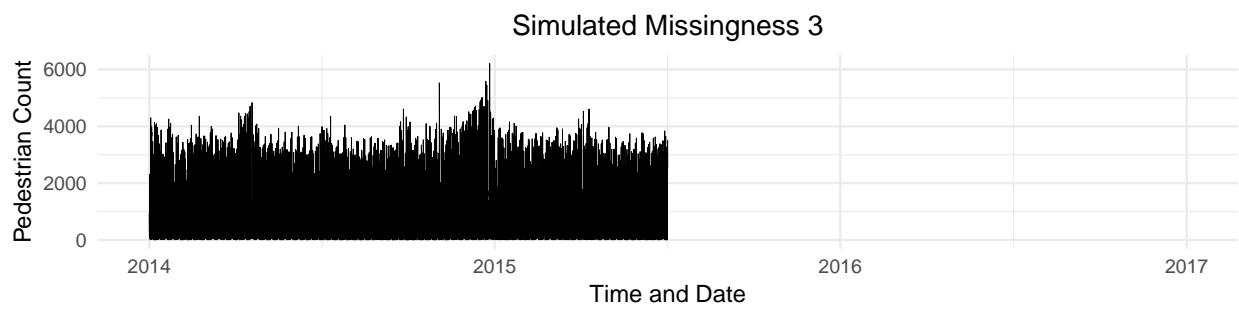


Figure 13: Plot of simulated missingness method 2 at Bourke Street Mall (South) with 50% data missing

Predictive Model

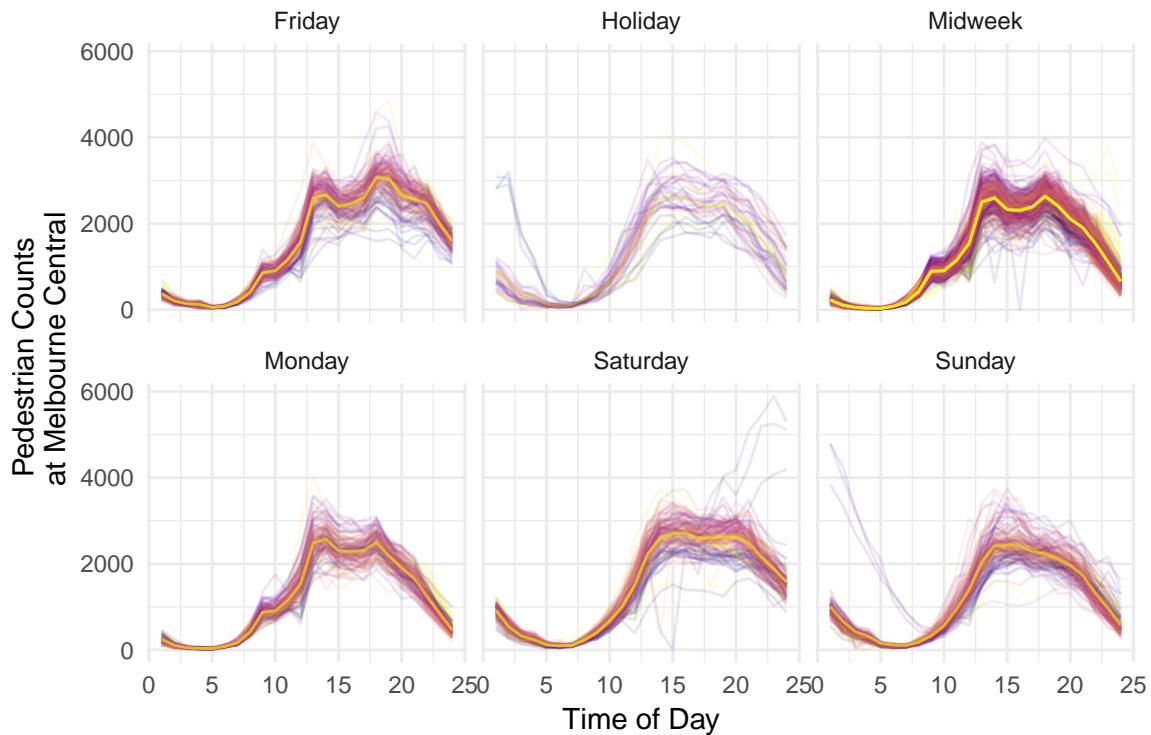
For the purpose of prediction, we want to use the most simplistic model possible to make quick predictions of the expected pedestrian counts at each location. The model should not be expected to be able to predict large deviations from time and date based estimates.

Similar to the simple imputation model used, we will also use a generalised linear model with quasi-poisson errors using time and date based variables as predictors. The model specification used is:

$$E[\text{HourlyCounts}|\text{Sensor}, \text{Month}, \text{DayType}, \text{Time}] = \exp(\mu_{\text{Sensor}, \text{Month}, \text{DayType}, \text{Time}})$$

$$\text{where } \mu_{\text{Sensor}, \text{Month}, \text{DayType}, \text{Time}} \sim \text{Month} + \text{Time} \times \text{DayType} \quad (2)$$

Using a GLM predicts well compared to estimating with other models such as AR(p) or other time series models when considering the tradeoff between model complexity and prediction accuracy.



As

When we estimate the GLM at each sensor location with R, we only need to save the elements in the model object which are required in order to retain the functionality of R's `predict.glm()` function. Saving the entire GLM object as given by the base R `glm()` function is highly inefficient when working with large datasets. This is due to the model object providing redundancies. In particular, elements which are unnecessarily saved for our

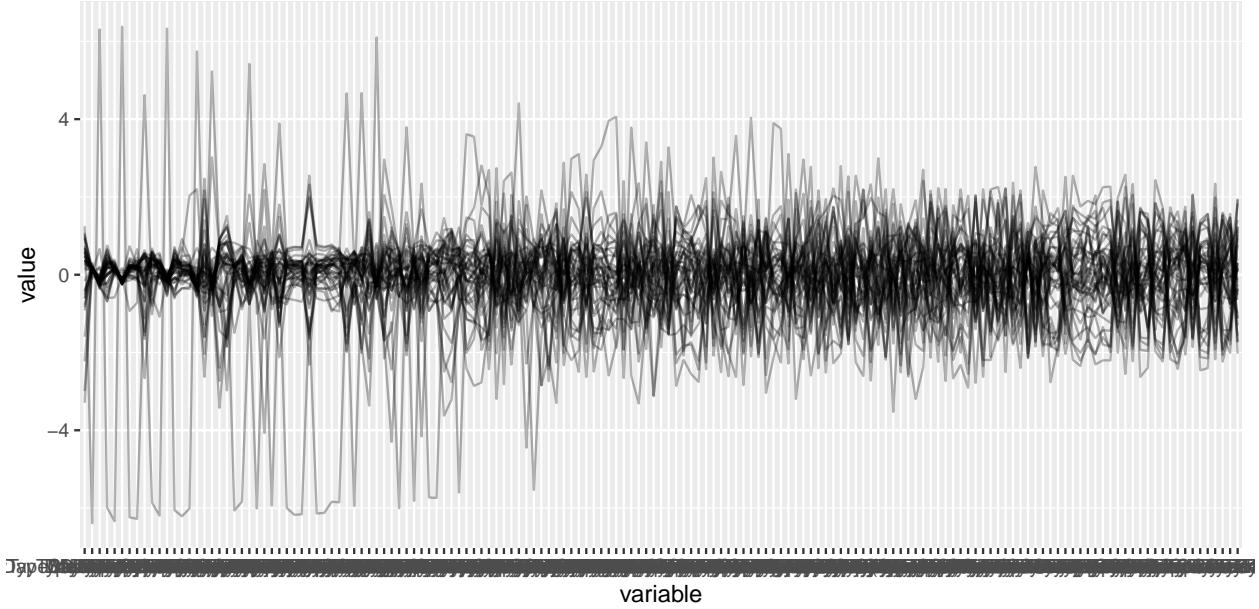


Figure 14: Plot of the estimated coefficients of the GLM at each sensor for each variable. Variables (x-axis) is ordered by the scagnostic measure, Outlying.

purposes: the training data, the predictors in a model matrix format (the matrix of \mathbf{X} after being formatted to be used for regression, such as recoding categorical variables into dummy variables), the response variable as another vector (\mathbf{Y}), the fitted values of the model after estimation, the residuals of the estimates, as well as the estimated effects.

The result of removing all the unnecessary redundant data elements in the GLM object is a storage size reduction from 29.9 Mb to 510 Kb. This is only 1.66% of the size of the original GLM object. With 43 sensors (and models), this represents a reduction in model object size from 1.3 Gb to 21.4 Mb.

The estimated model parameters are saved so that they can be used for on-the-fly predictions by end users without needing to estimate the model themselves. Using the estimated model parameters, we can write a new function to facilitate the predictions in an easy to use format.

```
ped_predict(pred_date = "2017-12-26", t_hour = 13, is_pub_hol = TRUE)
```

This function will return tibble (trimmed down version of `data.frame()` in R) containing the predicted counts at each sensor location for the 13th hour (13:00/1:00 PM) on Boxing Day (26 December) 2017. This tibble can then be easily used for visualisations or analysis. Due to the lack of data on future public holiday dates, the function assumes the date given is not a public holiday unless `is_pub_hol = TRUE` is provided.

Another function which uses the `ped_predict()`, `ped_predict_day()`, returns a tibble containing the predicted counts at each sensor location for all 24 hours of the date given to predict.