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

A thesis submitted for the degree of

Bachelor of Economics Advanced with Honours

by

Gavin Chin

Department of Econometrics and Business Statistics

Monash University

Australia

October 2017

Contents

Acknowledgements

I would like to thank my supervisor, Professor Dianne Cook, for assisting with my research and offering great advice. I would also like to thank Earo Wang for her support, as well as creating the extremely helpful `rwalkr` package.

Thank you to Professor Xueyan Zhao for coordinating a great Econometrics Honours course! And lastly, all my fellow Honours Class of 2017 for making the course so enjoyable!

0.1 Software used

R and RStudio were used for all analysis in this paper. The following R packages were used: `tidyverse`, containing the packages: `ggplot2`, `dplyr`, `tidyr`, `readr`, `purr`, `tibble`, `lubridate`, `rwalkr`, `knitr`, `plyr`, `foreach`, `doSNOW`, `pander`, `gridExtra`, `ggmap`, `viridis`, `broom`, `bookdown`

Abstract

The City of Melbourne provides access pedestrian data collected from pedestrian sensors placed throughout Melbourne's CBD. Through data analysis and exploration, the aim is analyse the pedestrian behaviour at different locations in the CBD in order to predict pedestrian counts. Issues in the dataset, with missing values and false zero counts, are investigated and solutions proposed. An improved imputation method is proposed, using different models based on the proportion of missing values. For sensor locations with a small proportion of missing values used generalised linear models using only time and date based variables as predictors, while sensors with a large proportion of missings used neighbouring sensor counts as predictors. Using this imputed data, predictive models can be trained more effectively, producing better estimates. A predictive model for on-the-fly predictions was wrapped into a function to facilitate point estimates in an easy and tidy format.

Chapter 1

Introduction

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". This paper will focus primarily on the pedestrian data collected from pedestrian sensors placed throughout Melbourne's CBD. In particular, the aim is to model the pedestrian behaviour at different locations in the CBD, and predict pedestrian counts.

The main issue which is encountered with traffic data in general is the presence of missing values. For the Melbourne CBD pedestrian data, data is missing as a result of sensors being installed over time, as well as technical issues. As a consequence, data imputation, the estimation of missing data, is required.

Imputation methods are explored due to the lack of complete data in the pedestrian dataset. Without imputation on the data, some sensors lack information about pedestrian activity. This makes statistical inference and model estimation difficult to compare between locations.

1.1 Motivation

Modelling pedestrian activity in the Melbourne CBD will provide insights to the social and consumer behaviour of the people within the city. In particular, prediction of pedestrian traffic will provide information about the way the city operates. Examples of some uses of

such information include infrastructure planning, or security planning (which has become quite important in recent times) within the government sector. Private uses of such information include marketing, such as finding optimal locations to launch advertising campaigns, resource management for businesses to improve staffing based on pedestrian traffic, and investment planning to analyse business plan feasibility.

Throughout the data exploration and analysis, issues with respect to missing values in the pedestrian data are explored and solutions to these issues are proposed. These issues have been seen to cause biased estimates when building models for prediction. Efficient methods for solving these issues is also a priority due to the size of the data. As such, the imputation method used should be computationally feasible.

Chapter 2

Background

2.0.1 The Data: City of Melbourne Pedestrian Data

The dataset being analysed can be obtained from the official City of Melbourne's online open data platform. The pedestrian data is in the form of hourly pedestrian counts for 43 sensor locations. In total, there is 1435512 observations in the dataset, representing 26293 hours of data. It is available in `.csv` format, allowing for easy data import into R. Access ti the pedestrian data using the the `rwalkr` package. 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.

The data source used in this paper is accessed with the `rwalkr::walk_melb()` function, which imports data from `pedestrian.melbourne.vic.gov.au`. This data source was chosen due to missing values being explicit (all possible rows in the data are given, even if data is missing). It is also updated more frequently.

The period of data which is being used for analysis is between 01/01/2014 to 22/10/2017 (date of this paper). This is due to the poor quality of the dataset pre-2014 with many missing values with few sensors being installed during at the start of pedestrian counts recording (in 2009).

Even with a truncated period for the data, no sensor location has complete data for the period between between 01/01/2014 and 31/12/2016 (the training period to be used

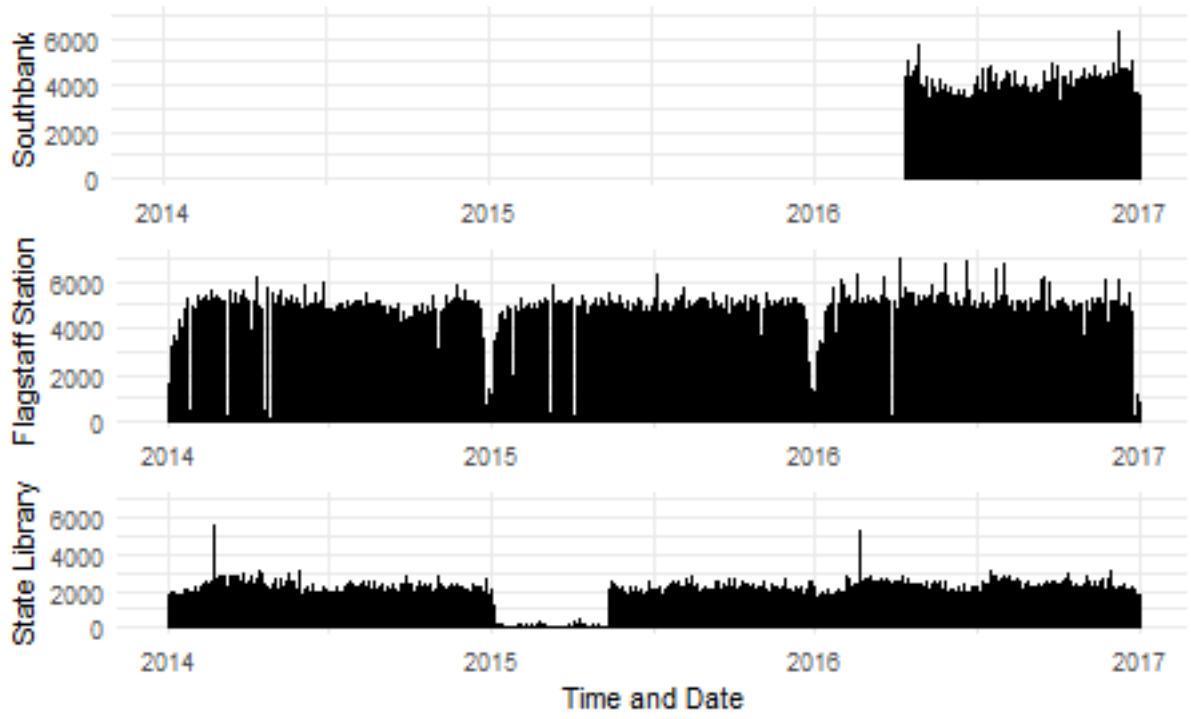

Figure 2.1: *Plot of hourly counts for Flagstaff Station, Southbank and State Library sensors during the training period of 2014-2016. It is observed that Southbank has a period without any data (pre installation), while Flagstaff Station appears to have complete data. State Library appears complete, although the first half of 2015 shows suspeciously low hourly counts for a long period.*

for predictive models in the paper). As a result, imputation of the data is required for meaningful analysis of pedestrian behaviour.

The sensor data is collected by sensors located throughout the Melbourne CBD under awnings and street poles. The sensors record the number of people passing through a counting zone each hour, sending this data to a server to be saved. The date of installation varies, as new sensors are constantly being installed over time.

2.0.2 Literature Review

Reviewing current research on imputation of missing values in traffic data, a paper on imputation of missing classified traffic data during winter season (Roh et al., 2016) stated some methods which were found to be poor for imputation. The data used in this research were from traffic counters located on the highway network in Alberta, Canada with between 40% and 60% missing data.

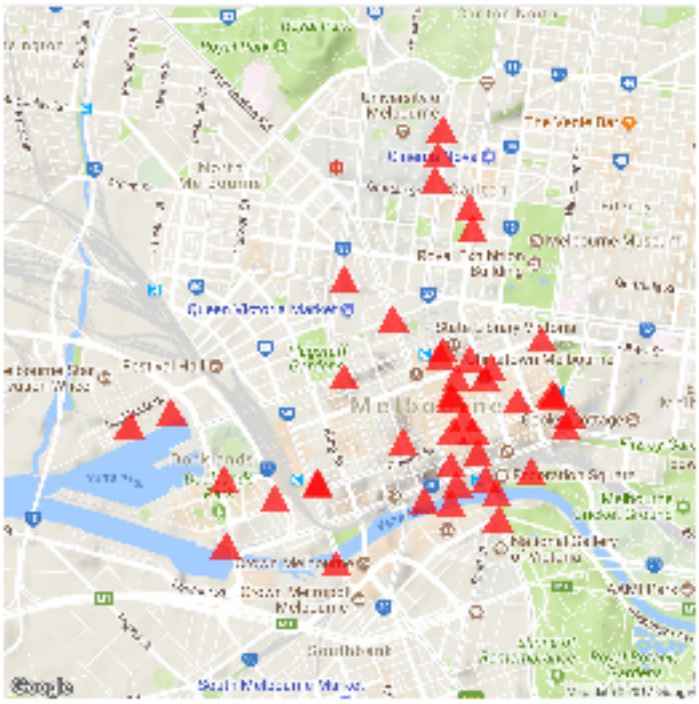

Figure 2.2: *Map of all the pedestrian count sensors currently installed around the Melbourne CBD. The proximity between sensors is very close at some locations. This indicates potential for imputation using nearby sensor counts as predictors.*

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). Roh et al. found that using a moving average worked the best out of all the heuristic methods used. The largest problem with heuristic methods, however, was 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 the highway traffic data, 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 major 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. Roh et al. proposed using non-parametric estimation methods

with k-Nearest Neighbours (kNN) for imputation, although this is unlikely to work well with the lack of data at certain sensor locations in the Melbourne CBD Pedestrian data. This is because kNN regression works best when there is a lot of historical data. The lack of historical data at newer sensor installations, where less than 12 months of training data is available, makes kNN unfeasible. Similarly, most other applications of imputation methods on traffic data used non-parametric regressions for traffic flow forecasting (Zhong, J., Ling, S. 2015) involving kNN.

Another important part of analysis which needed to be conducted when performing data imputation is assessment of robustness (Zhong, M. et al. 2005). Zhong et al. found that models using additional observations as input and more sophisticated prediction techniques were able to make better imputations on a consistent basis.

Chapter 3

Methodology: Imputation of Missing Values

3.0.1 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 quasipoisson error distribution, where specify the model as:

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

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

`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. Time of the day cannot be treated as an integer or numeric because it does not have a linear relationship with the pedestrian counts.

The quasipoisson error distribution is used 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 quasipoisson model estimates an additional parameter for overdispersion. This allows more flexibility in the model by allowing the

assumption that $E[Y] = Var[Y] = \mu$ to be relaxed. Instead, the quasipoisson distribution allows for $Var[Y] = \theta\mu$.

While this simple model works well with a small proportion of missing values, a large number of observations is required for estimation. Specifically, with this paramaterisation, $23 + 7 + (23 \times 7) + 1 = 192$ parameters need to be estimated. 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.

3.0.2 Improved imputation algorithm using 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, a GLM quasipossion regression with only time and date based variables as predictors is still an appropriate model to use. At the locations with a large proportion of missing values, however, neighbouring sensor counts are used for prediction.

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, it is not valid to simply classify all zero counts as `NA`, as true zero values are possible.

Table 3.1: *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 which is implemented is to look for long sequences of zero counts. In this case, any sequence of `Hourly_Counts = 0` running for longer than 6 hours can be considered `NA`. From the summary of lengths of repeated values (which are repeated at

least once), it is seen 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` will allow true zero counts to avoid being misclassified as `NA`. The length of 6 hours is selected as the limit 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 replacement of questionable zero values with the value `NA`, only then can the calculation of the proportion of missing values for each sensor location be performed. From these calculated proportion, classification of sensors as either having a "large" or "small" proportion of missing values is performed, where large is defined as $\texttt{NAprop}_{sensor} > \texttt{threshold}$ and `threshold = 0.1`.

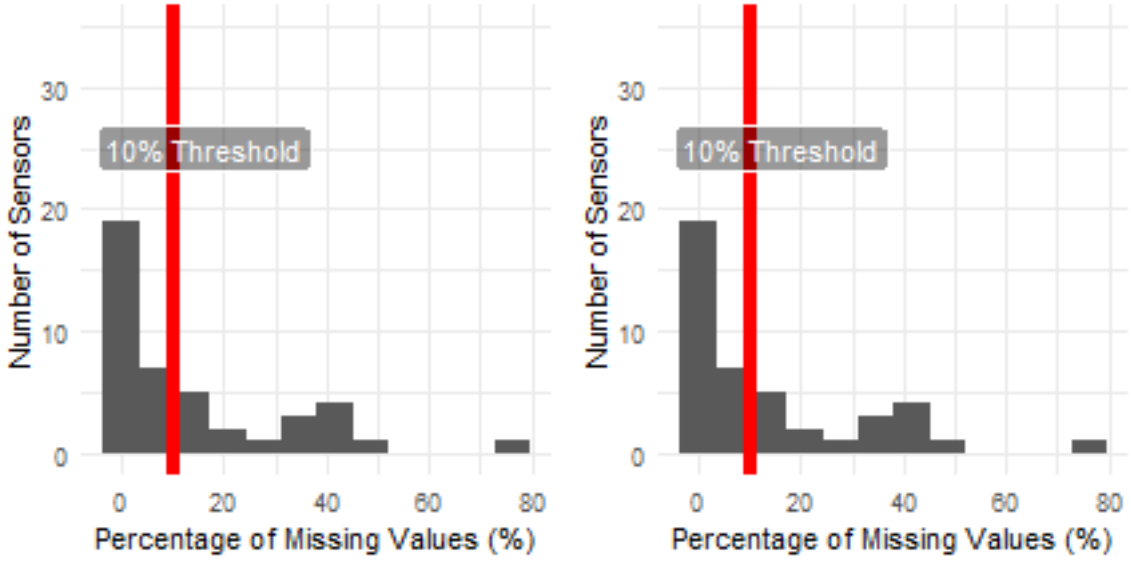

Figure 3.1: *Histograms of the proportions of missing values at each sensor during 2014 - 2016 when calculated before and after correction for false zero counts. The distributions of proportion of missing values at each sensor is significantly different after correcting for false zero counts in the data. This emphasises the necessity to correct for false zero counts as it will cause biased estimates.*

Because the models for the sensors with a large proportion of missing values rely on having neighbouring sensors having complete cases (no missing values), it is necessary to impute these values (counts at sensors with small proportion of missing values) first. This will ensure as much information is available for the models which use neighbouring sensors to to train on.

For sensors with a small proportion of missing values, a GLM quasipoisson model is estimated. To improve on the previous model used in the simple method, the model used is:

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

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

Instead of `HDay`, `DayType` is used which classifies Tuesday, Wednesday and Thursday as a single factor level, Midweek. This can be interpreted as the type of day. This can be done as the daily patterns on the Midweek weekdays are similar to each other. The intuition for the use of this variable is for the reduction in the number of parameters to estimate ($23 \times 2 = 46$ less parameters). Another consequence of this is more robust estimates for the midweek days, being less sensitive to outliers on a particular day. Because this model will be estimated for sensor locations with few missing values, enough data is available to add in month as an additive effect to help capture the annual seasonality. This is not possible at locations with a large proportion of missing values, particularly at newer installations where less than 12 months of data is available in the training period specified.

Using these estimated models, imputation of the missing values to produce complete data is performed 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%, there are 25 sensor locations to use as potential candidates to be used as neighbouring sensors for prediction.

For each of the sensor locations with high proportions of missings, the algorithm needs to decide which sensor to use for prediction. A simple method which is applied for finding the geographically closest neighbours is to take the haversine/great-circle distances between the sensor to be imputed and all the possible candidates.

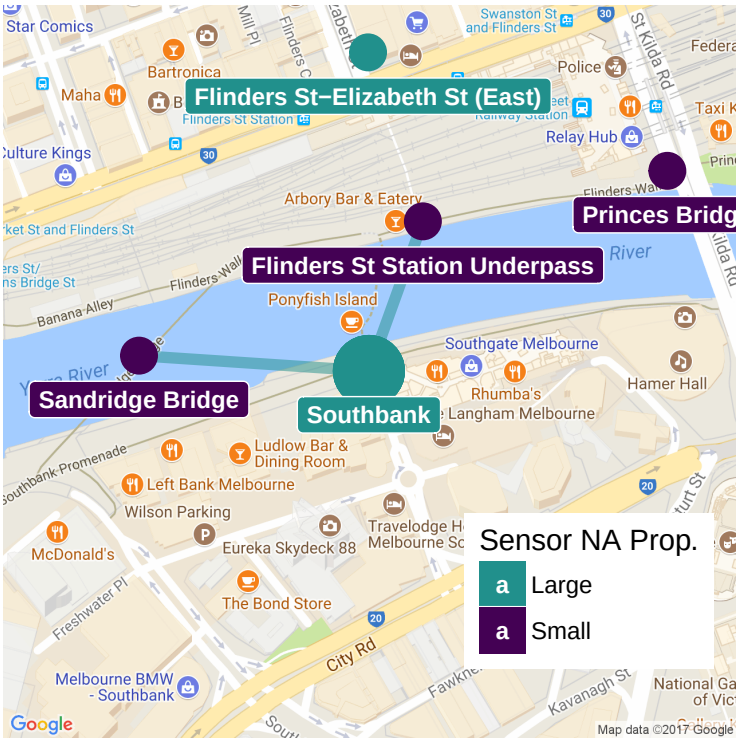

Figure 3.2: *Map showing the algorithm selecting Flinders St Station Underpass and Sandridge Bridge as neighbours by geographical distance to be used to predict the pedestrian counts at Southbank. The algorithm will not choose sensors with large proportions of missing values to be used as neighbours for prediction.*

Using the two geographically closest sensors (with complete cases) as defined by the haversine distance, the GLM specified for estimation is:

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

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

The model uses and interaction term between the standardised/scaled counts at the neighbouring sensors and the time of the day (hour factor).

The amount of people passing through the neighbouring sensors will be expected to be a strong predictor of the pedestrian counts as they are likely to also pass through. Using geographical neighbours, it is possible capture rare events effectively as the predictors are also random variables. Any large shocks to the counts at the neighbouring sensors will have an effect on the predicted counts. This method is a form of pattern matching, where

the sensor counts being imputed can be considered the study curve, and the neighbouring sensor counts are candidates for matching.

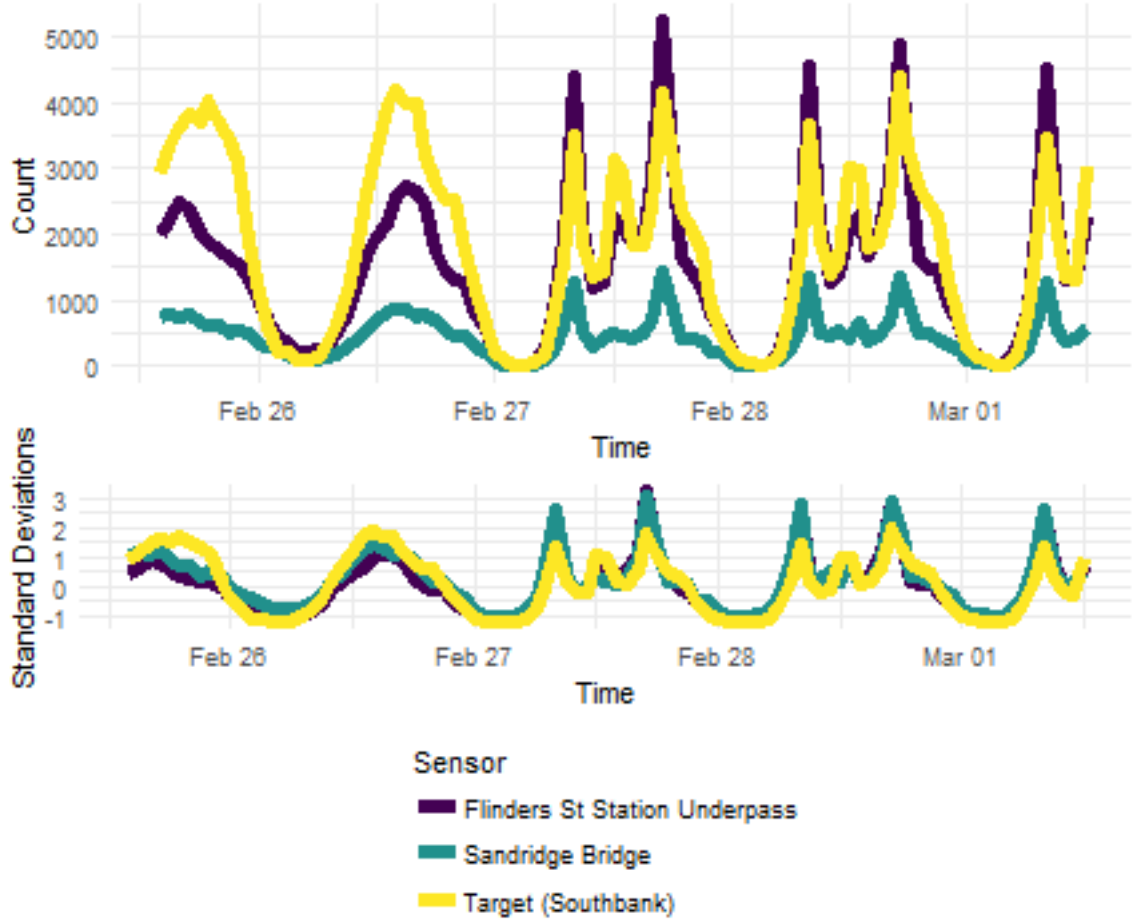

Figure 3.3: *Plot of the time series at the sensor with missing values, Southbank, and it's neighbouring sensors, as pedestrian counts and standardised. Because the counts are standardised to* `mean = 0` *and* `var = 1`*, the standardised counts are also standard deviations. The curves for the standardised counts at neighbouring sensors of Southbank closely match that of the standardised counts at Southbank. This demonstrates the importance of scaling counts at the neighbours when using them as predictors.*

In order to use a mixture of the patterns from two different sensors, the use of scaled counts at the neighbouring sensors is required to avoid the magnitude of the counts having an effect on the predicted counts. The counts are scaled by standardising to $\mu = 0, \sigma^2 = 1$ at each sensor (standardised within sensors). Because the covariance with neighbouring sensors may vary over time, an interaction term between the neighbouring sensor counts and the time of the day is included.

Chapter 4

Implementation of improved imputation algorithm

All data manipulation, exploration, visualisation and model estimation has been done in R using the RStudio IDE. A full list of R packages used is available in the Acknowledgements.

The exact code used for this paper can be found on http://github.com/gavinchin/honours2017

4.0.1 False zero-counts correction

The false zero-counts correction is performed in R using the run length encoding function, `rle()`. This function computes the length and values of runs of equal values in a vector. Unfortunately, this function does not recognise `NA` in the its computation, so all NA values are initially recoded to take the value 0 at each sensor. This results in all missing values and true zero counts to take the value zero when running `rle()`. With the returned run lengths, all sequences of repeated values longer than the defined length of 6 hours are replaced with `NA`. Any values which were also originally missing, but were not missing for longer than 6 hours (and hence are still coded as 0), are recoded to be `NA`. It must be noted that sequences of repeated values not equal to zero which run for longer that 6 hours would be incorrectly replaced with `NA`. This is highly unlikely, however, as 6 hours of the exact same pedestrian counts would be extremely rare.

4.0.2 Caculation of proportion of missing values and classification of sensors

In the working data, a wide format is used. This makes it easier to calculate missing value proportion at each sensor location, as it is the proportion of missing values in the column for each sensor. Using `is.na()` returns a logical vector indicating whether the sensor count is missing/`NA` or not. The sum of the instances where `is.na = TRUE` divided by the hours in the period of 2014-2016 is computed to get the proportion of missing values.

These proportions are saved for each sensor and then classified as either "large" or "small" based on a threshold value for the proportion. A list of sensors with large proportions of missing data and a list of sensors with small proportions of missing data is generated.

4.0.3 Estimation of GLM at sensor locations with a small proportion of missing values

A for loop is used to estimate the GLM specified:

```
glm(dfa[, i] ~ Month + DayType*Time, data = dfa,
    family = quasipoisson())
```

where `dfa` is the object name given to the data frame containing the training data after the false zero-count correction, and `i` is the i^{th} sensor in the list of sensors with a small proportion of missing values. These models are stored in a list, forming a list of models. An alternative approach to using a for loop in R is splitting the data into separate data frames for each sensor location (forming a list of data frames) and applying the `glm()` function on each data frame in the list using the function `purrr::map()`. However, when this was attempted, performance issues were encountered due to the large size of the estimated model objects in R. This was not an issue when a for loop was used. Additional code for removing unnecessary elements of the GLM model object before storing to the list of models, reducing the memory usage. Significant improvements in computing time is made with the use of parallel computing. Parallel computing is implemented with the use of `foreach()` and `doSNOW` rather than using the base R `for()` function. IT was found that estimation time was reduced by 50% when using a cluster size of two parallel nodes.

Additional nodes (more than two) did not result in a significantly reduced computing time, so only two were used.

4.0.4 Imputation at sensor locations with a small proportion of missing values

The predicted values are obtained using `predict.glm()`, using the option `type = "response"` to have predicted values returned predicted counts. The default option returns the predicted values of $\hat{\lambda}$ in the quasipoisson process, not the counts, $\exp(\hat{\lambda})$. Missing values are then replaced using the imputed values, generating a data frame of complete data as sensor locations for with a small proportion of missing values to be used for prediction at sensors with a large proportion of missing values.

4.0.5 Estimation procedure at sensor locations with a large proportion of missing values

The following steps are performed on each sensor with a large proportion of missing values inside a for loop.

Finding neighbouring sensors Using location data of the sensors provided by the City of Melbourne, a data frame containing the latitude and longitude coordinates of all the sensors which were classified as having a small proportion of missing values (and now have imputed, complete data), which are the candidates to be neighbours used for prediction. Using the Great Circle/Haversine distance equation, implemented in the function `gcd.slc()` from , an approximation of distance between each of the candidate sensors is calculated in and stored as a vector. Sorting by ascending order (using `sort()`) and taking the two smallest distances with `head(., 2)` returns the two geographically closest candidate neighbours.

Extracting neighbouring sensors counts and standardising Using the sensor location names, the neighbouring sensors counts are extracted from the imputed training data and each vector is standardised to $\mu = 0, \sigma^2 = 1$ using the base R function `scale()` then stored.

GLM estimation The GLM for sensor locations with a large proportion of missing values also uses `glm()` as follows:

```
glm(dfa[, i] ~ Time*close_dfsc + Time*close_dfsc2,
    data = dfa, family = quasipoisson())
```

where `dfa[, i]` is the vector of pedestrian counts for the sensor being estimated, `Time` is the factor level for the hour of the day and `close_dfsc` and `close_dfsc2` are the standardised pedestrian counts at two geographically closest sensors.
Like the GLM for sensor locations with a small proportion of missing values, the model object is slim down to remove unnecessary elements to reduce memory usage. It is also noteworthy that the computational burden of the procedure for imputation using neighbouring sensors is significantly less due to less parameters being estimated.

Replacing missings with imputed values Unlike the GLM used for imputation for sensors with a small proportion of missing values, the model for neighbouring sensors requires transformation of the counts at the neighbours for imputation. As a consequence, it is more efficient to use `predict.glm()` inside the for loop as the neighbouring sensor counts have already been standardised when used for training the GLM. Again, the option `type = "response"` is specified to return predicted counts.

4.1 Imputation Results

Firstly, the initial, simple method of imputation used is evaluated:

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

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, the counts from Southern Cross Station will be used to evaluated prediction accuracy. At this sensor, the proportion of missing values (adjusted) is 0.02.

Firstly, the goodness of fit of the predictions made within the training period (01/01/2014 to 31/12/2016) are evaluated.

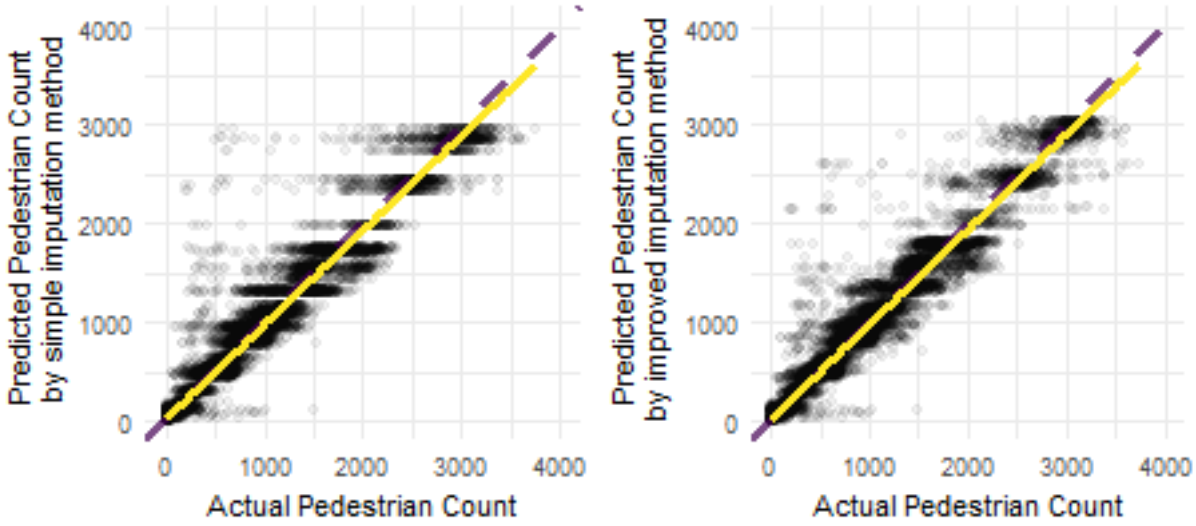

Figure 4.1: *A plot of the in-sample Actual vs Predicted pedestrian counts at Southern Cross Station (sensor location with small proportion of missing values) for the simple GLM model and the improved GLM model. The dashed line is $x = y$, representing a reference for perfect fit, while solid line is the linear fit of the points. The improvement is very minor at Southern Cross Station as the data at this sensor is close to complete. Inclusion of month as a predictor is a main source of improvement in the model in this case.*

Visually, it can be seen that the fitted values are good estimates of the pedestrian counts as the relationship between the actual values and the fitted values is very close to 1:1. This is best observed when performing a least squares estimate (linear fit) of $\widehat{Fitted}_t = \hat{\beta}_0 + \hat{\beta}_1 Actual_t$, where the estimated $\hat{\beta}_0$ is close to 0 and $\hat{\beta}_1$ is close to 1.

This provides evidence that the GLM model with only time and date based variables work well when there is only small periods of missing data.

However, using this same model specification at sensor locations with large proportions of missing values, the prediction accuracy is poor. The sensor at Australia on Collins, where the proportion of missing values is 41.96%, is used for the evaluation of goodness of fits.

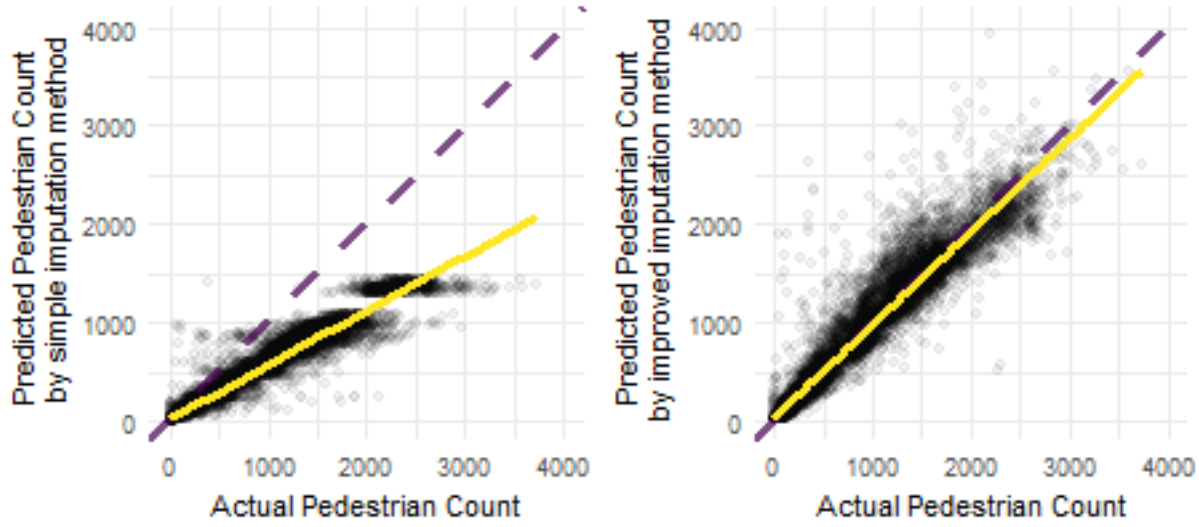

Figure 4.2: *A plot of the in-sample Actual vs Predicted pedestrian counts at Australia on Collins (sensor location with large proportion of missing values) for the simple GLM model and the improved GLM model. The dashed line is $x = y$, representing a reference for perfect fit, while solid line is the linear fit of the points. The improvement in the imputation model is significant, where the predicted values lie closer to the dashed line. The simple model shows bias caused by the false zero-counts, causing underestimation.*

It is observed that the simple model has biased estimates 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 used to measure goodness of fit is mean absolute relative error (MARE). It is calculated by:

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

Absolute errors are used instead of squared errors as penalising larger errors relative to smaller errors was not desirable. A relative measure is used so that comparisons can be made between sensors.

A separate MARE for predictions in-sample (in the training set) and the predictions out of sample (test set of 2017 data) is calculated. This is to help identify overfitting of the training data.

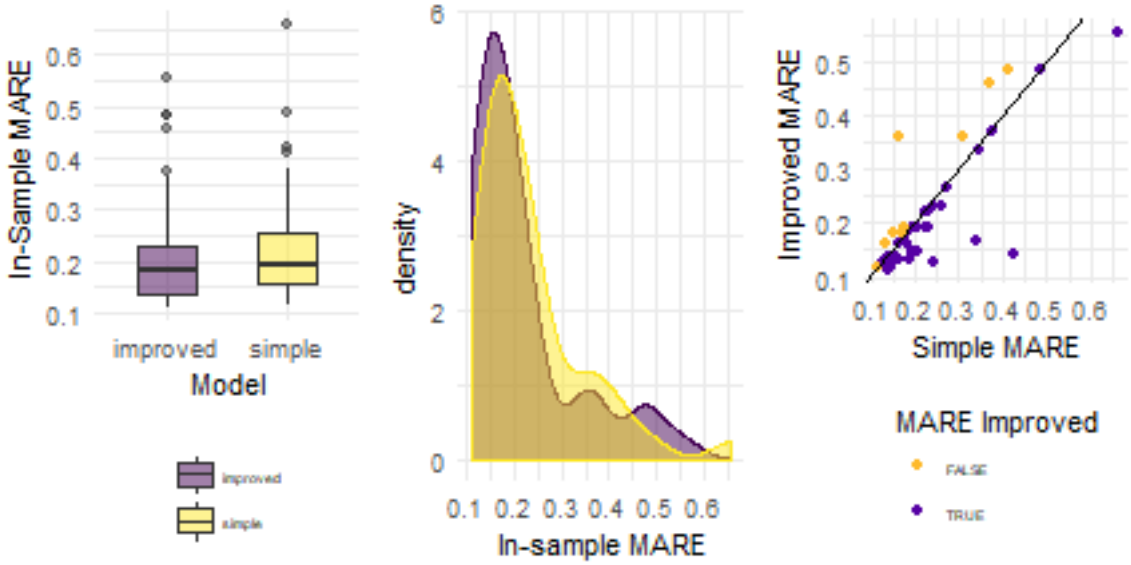

Figure 4.3: *Distribution of in-sample (2014-2016 data) MARE by imputation model. It is seen that the distribution of MARE in the improved model is generally lower, with 34 locations having better fit with the improved model*

While it is not very clear when comparing the distribution of in-sample MARE between imputation models, comparisons between the in-sample MARE at each location shows that the majority of locations had better in-sample fit with the improved model using the algorithm. It was observed that some locations did have worse in-sample fit, but overall, an improvement in model accuracy is seen with 34 locations 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, the replacement of missing values with imputed values is illustrated. It can be seen that the general pattern of pedestrian counts is retained, and thus should have minimal impact when using the imputed data for training predictive models.

More importantly, the performance of the model's predictions of the out of sample observations (2017 data) is evaluated as cross validation. Due to potential false zero counts in the test data sourced from `rwalkr::walk_melb()`, correction of false zero-counts are required to be able to properly evaluate the goodness of fit of out-of-sample observations.

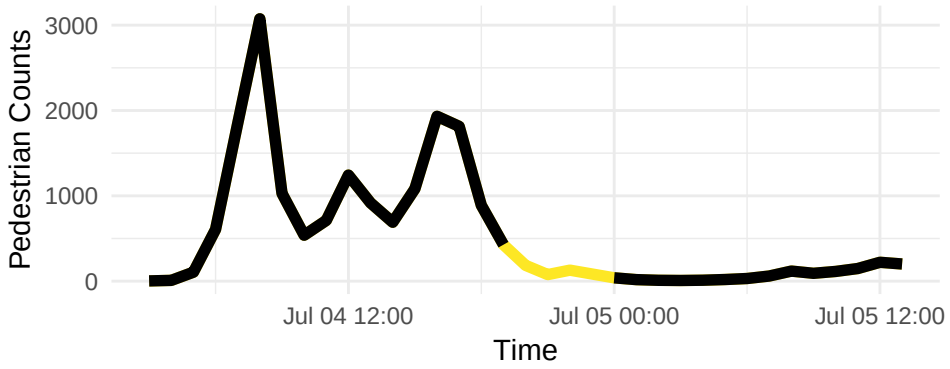

Figure 4.4: *A plot of the pedestrian counts at Southern Cross Station for 04/07/2014 - 05/07/2014. The black line is the observed counts, where the yellow line is the imputed counts. This illustrates the replacement of missing values with imputed values.*

The method used for this is identical to that used on the training set in the algorithm, with replacement of repeated value sequences of length 6 hours or longer with the value `NA`.

Again, comparing the fit at a sensor location with a small proportion of missing values, Melbourne Central shows the bias in the estimates of the simple model caused by false zero-counts. There is good improvement in the prediction accuracy using the improved method.

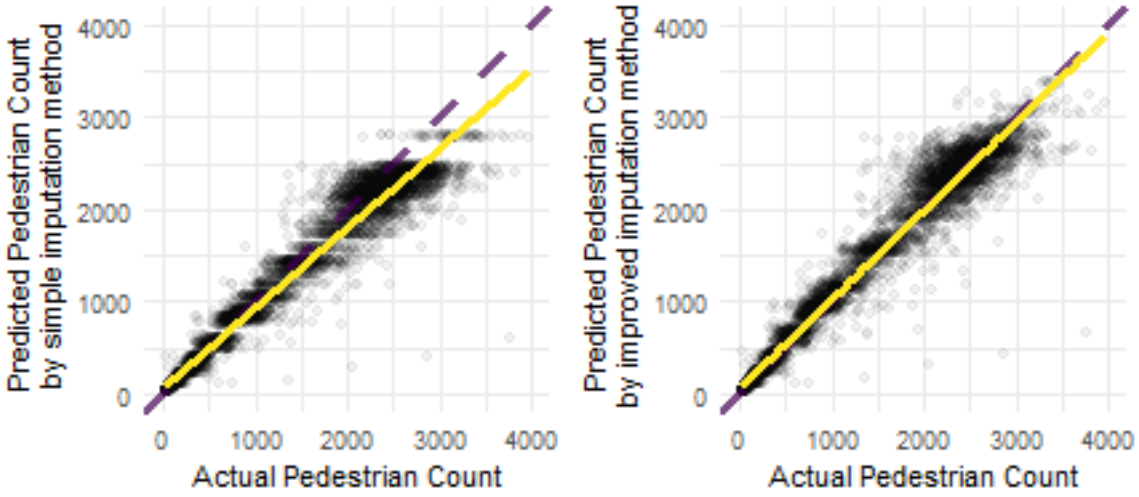

Figure 4.5: *A plot of the out-of-sample (2017) Actual vs Predicted pedestrian counts at Melbourne Central (sensor location with small proportion of missing values) for the simple GLM model and the improved GLM model. The dashed line is $x = y$, representing a reference for perfect fit, while solid line is the linear fit of the points. The improved method at Melbourne Central is seen to correct underestimation in the simple model by adding month of the year as a predictor in the model.*

When comparing the out-of-sample MARE between models at Melbourne Central, the improvement in prediction accuracy is seen by the MARE decreasing from 12.63% to 10.17% with the use of the improved model.

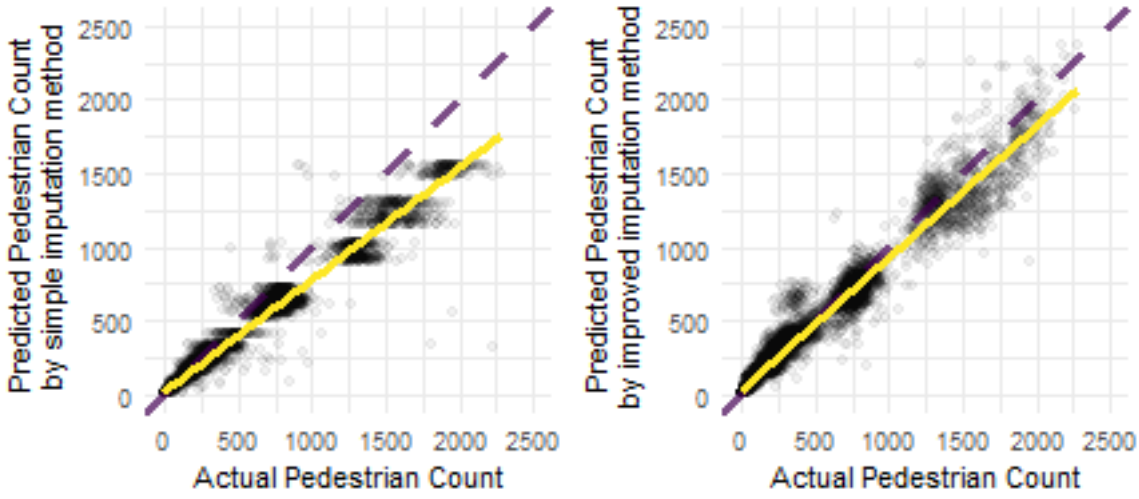

Figure 4.6: *A plot of the out-of-sample (2017) Actual vs Predicted pedestrian counts at Collins Place (South) (sensor location with large proportion of missing values) for the simple GLM model and the improved GLM model. The dashed line is $x = y$, representing a reference for perfect fit, while solid line is the linear fit of the points. Using neighbouring sensor counts as predictors, as well as correcting for the false zero-counts in the training data, are seen to reduce the degree of underestimation in the predicted values.*

At a sensor location with a large proportion of missing values, Collins Place (South), the tendency to underestimate counts with the simple imputation model is also observed. The improved imputation method reduces the underestimation of the predictions and provides better predictions as a result of the false-zero correction and using neighbouring sensor counts as predictors. The MARE is greatly improved, where it decreases from 21.4% to 14.21% with the use of the improved model.

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 2415.54%, 3017.97% and 23.87% 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

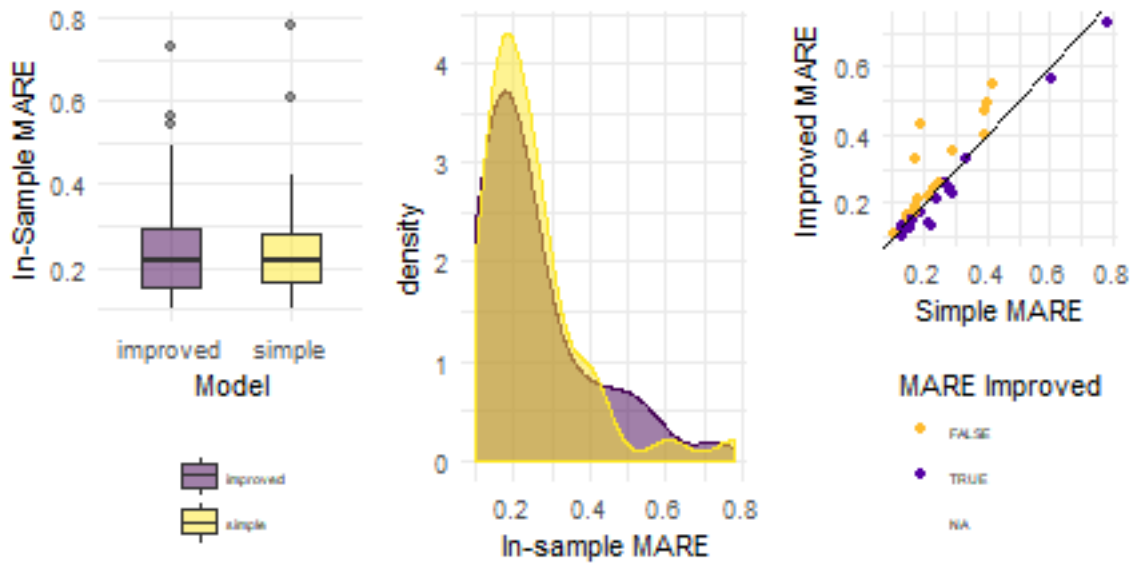

Figure 4.7: *Distribution of out-of-sample (2017 data) MARE by imputation model, excluding Spencer St-Collins St (North)/(South) and Flinders St-Swanston St (West) sensors. Spencer St-Collins St (North)/(South) and Flinders St-Swanston St (West) was omitted from the the plots above as the out-of-sample MARE is extremely high. Improvement in the MARE is seen at most sensors again, but some sensors did not have improved prediction accuracy due to their patterns varying from thier neighbours over time.*

neighbouring sensors has changed. This is because the in-sample MARE at these sensor locations were quite acceptable at 13.01%, 19.44% and 11.62% 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 neighbouring sensors. It was also seen 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 it is possible to add a variable for year to allow for year to year growth, the short period of training data doesn't make this appropriate at this stage.

Another method to perform cross-validation of the robustness of the model is to remove data at a sensor, and evaluate the predictions made. This will allow us to evaluate the imputation model's performance of out of sample but within training period observations, which is the primary concern with them imputation process.

The sensor selected with a small proportion of missing values, Bourke Street Mall (South), has 99.981% of data available. As observed in the data, missing data is present in different

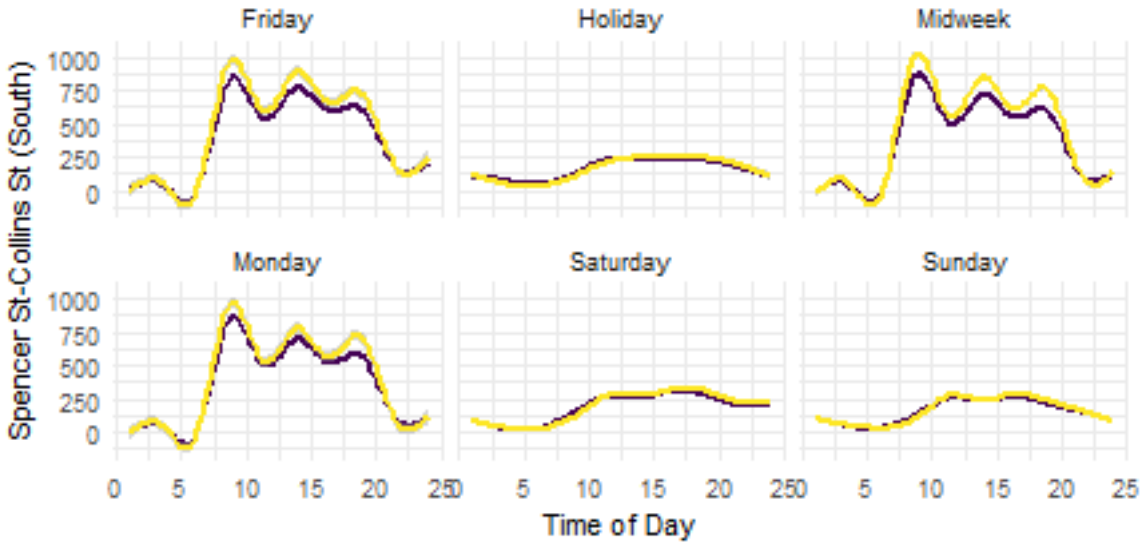

Figure 4.8: *Average pedestrian counts by time of day and type of day (DayType) at Spencer St-Collins St (South) for training data (dark line) and test data (light line). The pedestrian counts characteristics are seen to be different between the in-sample observations (2014-2016) and the out-of-sample observations (2017). This explains the much larger MARE in the out-of-sample predictions, as the model cannot account for a change in pattern over time.*

ways: random, short periods of missing data (no more that 6 hours) or long periods of missing data (multiple months). Simulation of both types of missingness at this sensor location is performed.

First, data is randomly removed such that 20% of the training data is missing, then the imputation model is estimated. Using the estimated model, the MARE within the training data is calculated (in-sample MARE).

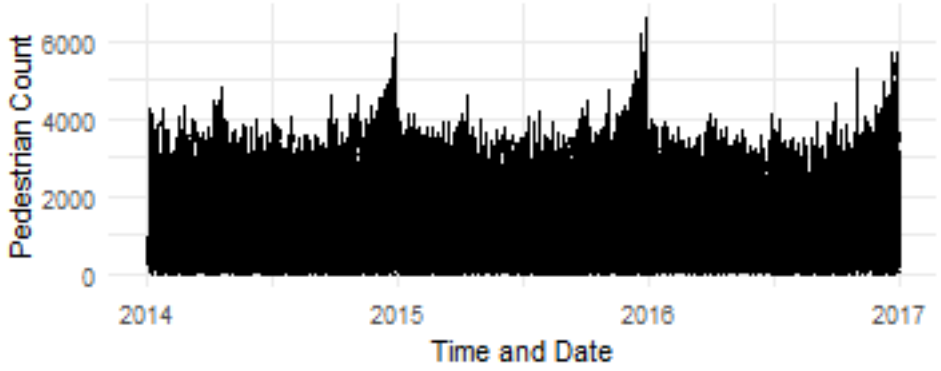

Figure 4.9: *Plot of simulated missingness method 1 at Bourke Street Mall (South). 20% of the observations are removed from the traning data for the imputation model at random.*

With random hours of missing data, the predictions made by using neighbouring sensors are estimated with good accuracy with a low MARE of 11.04%. Unfortunately, this type of missingness is not consistent with what is generally observed at most sensors with large proportions of missing data.

Instead, a more meaningful method is to remove 20% of the data in a single period to simulate what is typically observed. A random date is taken and then 2629 hours of data is removed before and after this date (10% before, 10% after). The imputation model is estimated again in order to calculate a MARE.

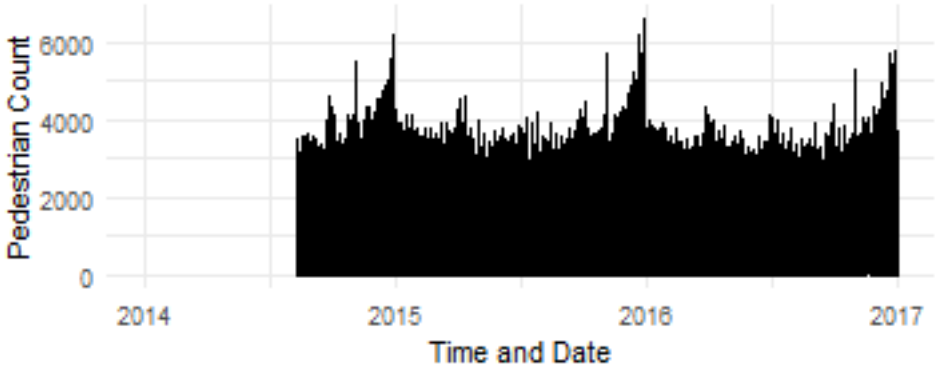

Figure 4.10: *Plot of simulated missingness method 2 at Bourke Street Mall (South) with 20% of the training data removed as a sequence. This method of simulating missing data is a better representation of the missing data observed at sensors with a large proportion of missing data compared to randomly removing hourly observations from the training data.*

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 performs quite well when 20% of the observations from the training data is missing/removed.

Removing 50% of the data using the same method, it is also found that the in-sample MARE is not affected significantly by the larger proportion of missing training data where it is only 11.26%. As such, it can be concluded that, at least for this sensor location, the imputation using neighbouring sensor counts as predictors is robust to high proportions of missing data.

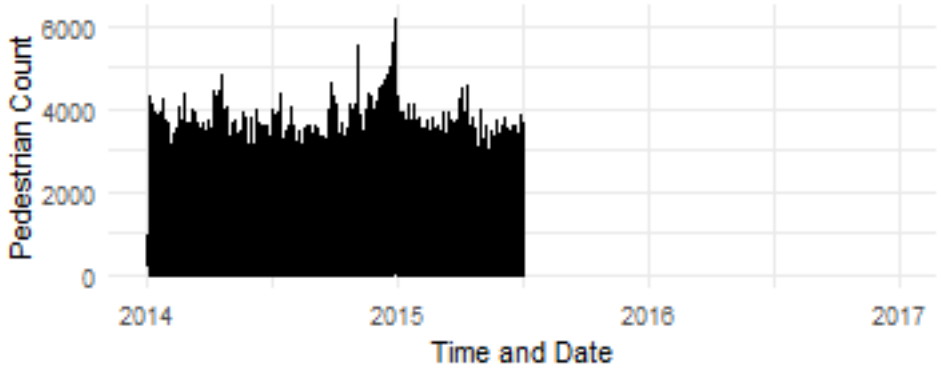

Figure 4.11: *Plot of simulated missingness method 2 at Bourke Street Mall (South) with 50% of the training data removed as a sequence. A large proportion of the training data is removed to test the robustness of the imputation method used at sensors with extremely large proportions of missing data during the training period defined.*

Chapter 5

Predictive Model

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

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

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

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

Using a GLM predicts well compared to estimating with other models such as AR(p) or other time series models when considering the trade off between model complexity and prediction accuracy. Due to multiple seasonalities, AR(p) models would need to be considered on a hour of the week basis at each sensor location. Prediction with time series models would also require of historical data. For the objective of this predictive model,

the additional inputs required by end users to predict pedestrian counts when making out of sample predictions is not desirable.

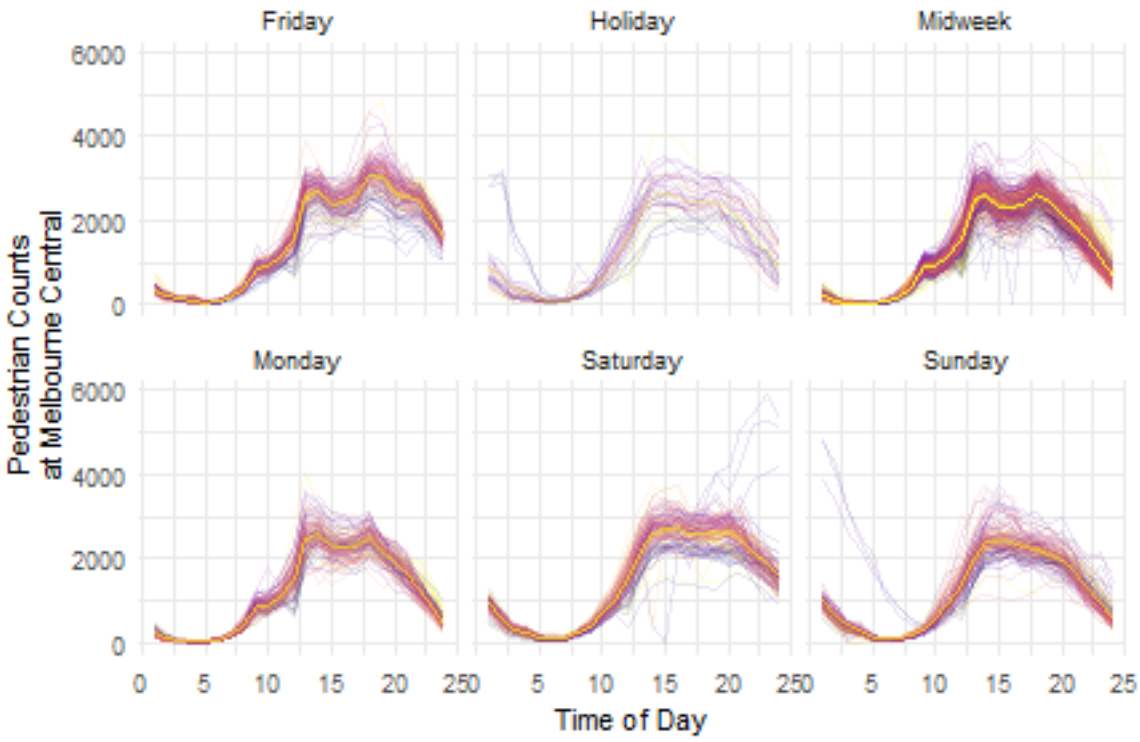

Figure 5.1: *Plot of daily pedestrian counts series at Melbourne Central by DayType and Month. Month of the year is mapped to colour to help identify annual seasonal patterns, where month has an additive effect on the pedestrian counts. At Melbourne Central, different months can be seen to have different mean counts for a given type of day and hour of the day.*

A plot of the daily pedestrian counts series at Melbourne Central by type of day and month show that time and date variables as deterministic predictors adequately explains most variation in the pedestrian counts. It can be seen that month has an additive effect as the pattern of the hourly counts for a given date are not affected by month. The six different factor levels that `DayType`, the type of day, is seen to be a sufficiently explain six different patterns. The type of day, Midweek, shows evidence that no additional information is gained from identifying the day to be Tuesday, Wednesday or Thursday.

When estimating the GLM at each sensor location with R, only the necessary elements in the model object which are required in order to retain the functionality of R's `predict.glm()` function are saved to file. 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 the purposes of the predictive model: 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, a new function is written 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.

Using these function as a basis, visualisations of pedestrian counts can be easily generated in R:

```
ped_predict_day("2018-05-21") %>%
  ggplot() + geom_line(aes(x = Date_Time, y = Southbank))
```

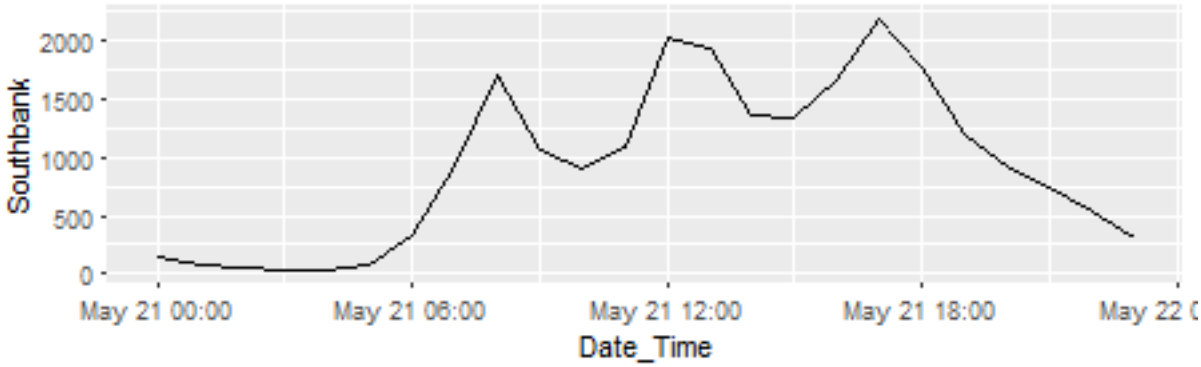

Figure 5.2: *Prediction for 21/05/2018 in R using* `ped_predict_day()`*, then plotting the time series using* `ggplot2`*. This shows the user friendly nature of the function, where the function returns predictions in a tidy, wide format. No data manipulation is required in order to plot the predicted counts.*

Chapter 6

Conclusion

Analysis of the Melbourne CBD pedestrian data made publicly available by the City of Melbourne has revealed issues with the dataset with relation to missing values in particular. A major issue which was found when using data sourced from the City of Melbourne's official pedestrian data visualisation (also accessible using `rwalkr::walk_melb()`) is potential for false zero-counts where missing values were given the value of zero. As outliers have major implications on analysis, a correction for the false counts in the data was proposed, by checking for lengths of repeated value sequences.
An imputation method for the pedestrian data was developed with a theory of sensor locations with a large proportion of missing values require modelling with non-deterministic predictors due to the lack of data. This method involved using neighbouring sensor counts data for prediction. Sensors with a small proportion of missing values used imputation models with only time and date variables as predictors. The proposed algorithm was evaluated in comparison to a single model specification of a generalised linear model with only time and date based variables at each sensor location and uncorrected training data (false zero counts not corrected). This method was found to have improved prediction accuracy, as well as being robust at high levels of missingness.

Implementation of the imputation algorithm was made such that it is efficient, reproducible and general enough to allow for changes to model parameterisation and new data with from new sensor installations.

Using the imputed data, a predictive model to be used for point estimates for a given time was built and packaged in a function in R, `ped_predict()`, to facilitate predictions in an easy and tidy format. The model has been saved so that predictions can be made without the end user needing to estimate the model prior to prediction.

Further research to improve the imputation algorithm can be made, such as more alternative models for different proportions of missing values. Cross validation of different missing proportion threshold values could also improve the prediction accuracy of the imputation models.

References

City of Melbourne. (2017) City of Melbourne - Open Data Portal, data.melbourne.vic.gov.au

City of Melbourne. (2017) City Of Melbourne: Pedestrian Counting System, pedestrian.melbourne.vic.gov.au

David Kahle and Hadley Wickham. (2016) ggmap: Spatial visualization with ggplot2

Simon Garnier. (2017) The viridis color palettes

Lionel Henry and Hadley Wickham. (2017) purrr: Functional programming tools

Kirill Müller and Hadley Wickham. (2017) tibble: Simple data frames

Hyuk-Jae Roh, Satish Sharma, and Prasanta K. Sahu. (2016) Imputation of missing classified traffic data during winter season

Vitalie Spinu, Garrett Grolemund, and Hadley Wickham. (2016) lubridate: Make dealing with dates a little easier

Mario Pineda-Krch. (2010) Great-circle distance calculations in R, R-bloggers

Earo Wang. (2017) rwalkr: API to Melbourne Pedestrian Data

Hadley Wickham and Winston Chang. (2016) ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics

Hadley Wickham and Romain Francois and Lionel Henry and Kirill Müller. (2017) dplyr: A grammar of data manipulation

Jing-ting Zhong and Shuai Ling. (2015) Key factors of k-nearest neighbours nonparametric regression in short-time traffic flow forecasting, Proceedings of the 21st International Conference on Industrial Engineering and Engineering Management 2014, 9–12

Ming Zhong, Satish Sharma, and Zhaobin Liu. (2005) Assessing robustness of imputation models based on data from different jurisdictions: Examples of Alberta and Saskatchewan, Canada, Transportation Research Record: Journal of the Transportation Research Board 1917, 116–126

Nina Zumel. (2014) Trimming the fat from glm() models in R, Win-Vector LLC

Appendices

6.1 Appendix I: Specification of imputation models:

Model at sensors with small proportion of missing values

$$E[HourlyCounts_{sensor,t}|.] = \exp\{\beta_{0,sensor} + \sum_{j=1}^{11}(\beta_{1,j,sensor}Month_t^j) + \sum_{k=1}^{23}(\beta_{2,k,sensor}Time_t)$$

$$+ \sum_{l=1}^{5}(\beta_{3,l,sensor}DayType_t) + \sum_{k=1}^{23}\sum_{l=1}^{5}(\beta_{4,k,l,sensor}Time_t DayType_t)\} \qquad \forall \texttt{NA}_{prop}(sensor) < \texttt{threshold}$$

Model at sensors with large proportion of missing values

$$E[HourlyCounts_{sensor,t}|.] = \exp\{\beta_{1,sensor}HourlyCounts^{SC}_{neighbour_1,t} + \beta_{2,sensor}HourlyCounts^{SC}_{neighbour_2,t}$$

$$+ \sum_{k=1}^{23}(\beta_{3,k,sensor}Time_t) + \sum_{k=1}^{23}\left(\beta_{4,k,sensor}HourlyCounts^{SC}_{neighbour_1,t}Time_t\right)$$

$$+ \sum_{k=1}^{23}\left(\beta_{5,k,sensor}HourlyCounts^{SC}_{neighbour_2,t}Time_t\right)\}$$

where $Month_t, DayType_t, Time_t$ are broken into dummy variables.