COM3018 RubixOnRails PBA Report

Analysing periodic Santander bike usage in London to predict future demand

Abstract

Climate change and global warming is an ever-expanding issue across the globe. Eco-friendly transport options were introduced to combat high pollution levels in cities such as London. Santander Bikes was one such measure introduced. However, as Santander Bikes are unfortunately finite in resources, it would be beneficial to create a model that can accurately predict trends and usages of the Santander Bikes to plan accordingly. We expected to see overall trends such as an increase in bike usage in warmer weather and seasonal repeating patterns such as increased bike usage during rush hour periods. We will also find which variables have the biggest impact on bike usage. The purpose of this report was to see how different variables impact the usage of Santander bikes, such as time and weather, and find repeating trends and patterns using the different modelling algorithms: Regression, Time Series, and Neural Networks. While regression was not able to highlight any clear relationships; time series was able to show seasonal repetitions with daily patterns, displaying clear spikes during weekdays and rush hour. We also managed to develop a highly accurate model using Deep Neural Networks, that was able to determine the most important variables in determining bike usage were time and whether it was weekends.

Table of Contents

Table of Figures

1. Introduction

Climate change and global warming is an ever-expanding issue across the globe. Eco-friendly transport options were introduced to combat high pollution levels in cities such as London.

For instance, Santander bikes were introduced into London in 2010 not just to improve transport within the city but reduce pollution. In 2018 it was projected that over 10 million bikes were rented across the year in London alone.

We wish to investigate how different variables can affect the number of bikes that are going to be rented on a day; this can range from weather controls such as it being cloudy to whether it is on the weekend or during rush hour. This report will contain the procedure behind preparing, pre-processing and integrating the chosen dataset. We're then going to investigate a wide range of modelling methods to find out which one is the best fit for our hypothesis.

The main objective of the project is to accurately predict the number of bikes that are going to be rented the following day to help streamline the business process of distributing bikes to set locations.

Hypothesis: During times of holidays and weekends, we expect the bike usage to go up just by virtue of having a higher population in the city. Peak times are expected to play a large role in hourly bike usage. However, we expect the weather to have a more acute effect on the number of bikes used each day.

2. Project Definition

2.1 Project Overview

The objective for this project is to analyse the chosen dataset, which gives the number of rental bikes hired in London between 2015 and 2018, on an hourly basis. By doing this, we expect to write the R code to predict the expected number of rental bikes hired and identify trends within the data. The dataset used in this project is from Kaggle (kaggle.com). This dataset was noticeably clean with no missing values, and the data types were consistent to their respective column.

2.2 Data Dictionary

Below is the data dictionary for the chosen dataset used in the project, showing the column names, column description and data type for the values.

Column Name	**Column Description**	**Data Type**
timestamp	Timestamp field to group the data into hourly entries (e.g. 2015-01-07 11:00:00)	char
cnt	The count of new bike hired during the hour	int
t1	The real temperature in C	double
t2	The temperature that it feels like in C	double
hum	Humidity in percentage	double
wind_speed	Wind speed in km/h	double
weather_code	**Weather code category description:** 1 = Clear; mostly clear but have some values with haze/fog/patches of fog 2 = Scattered clouds/few clouds 3 = Broken clouds 4 = Cloudy 7 = Rain/light Rainshower/light rain 10 = Rain with thunderstorm 26 = Snowfall 94 = Freezing Fog	int
is_holiday	Boolean field, 1 = holiday / 0 = non-holiday	boolean
is_weekend	Boolean field, weekend / 0 = non-weekend	boolean
season	**Meteorological seasons represented as:** 0-spring ; 1-summer; 2-fall; 3-winter.	int

2.3 Project Objectives

The rental bike company must move, add and remove bikes to each bike rack every morning. Essentially, they would want the correct number of bikes so that the bike racks have enough bikes to support the day's demand but are not at full capacity preventing bikes being dropped off at that location. It is complicated to predict the number of bikes needed on a day as there are weather and temperature implications in addition to the season and whether it is a weekend or holiday.

The project's main goal is to accurately predict the number of rental bikes used on the following day. This will improve the effectiveness of allocating the correct amount of rental bike on a given day. This amount will be estimated by knowing whether it is a weekday, weekend, or holiday alongside the predicted weather conditions and temperature for that day.

There is a fuel cost of transporting a bike to any given rack throughout the city. Through this project, we intend to develop methods to advise the number of bikes to deploy and therefore save and optimise the business's fuel costs.

The business analytic techniques that will be investigated during this project are; regression, time series and neural networking.

Linear regression should be useful for identifying trends in the data. These methods will be used as a basis to create simple relationships (e.g. does the temperature and wind speed affect the number of rental bikes hired on a given day) to identify correlations within the data. Afterwards, we plan to use time series to separate the data into periods to further facilitate the identification of trends on a given day, week or month. Finally, the implementation of a neural network allows relationships in sequence to be learnt and therefore, predictions to be made.

Primary Objective
Create a model that will provide accurate predictions of bike usage count for any given hour provided the required data.

Secondary Objective #1
Glean conclusive information regarding the relationships between predictors from the dataset.

Secondary Objective #2
Validate our hypothesis about the expected ebb and flow of usage depending on external factors from the data such as "Rush Hour". Identify trends within the dataset.

2.4 Workload Distribution

The workload was distributed evenly among each group member. As a group, we allowed each member to perform to their strengths, breaking down the design, implementation and report writing tasks.

All members were active in each role.

We divided the group into two subgroups, each group worked on a different classifier, allowing three members to work on time series and regression, where the other two would work on the neural network. Moreover, all members of the group contributed to the report writing section of the project.

3. Data Preparation

3.1 Data Quality

It is important to ensure that the data quality is of a high enough standard to warrant use in our further training and testing of models. Therefore, we need to evaluate it against a set of criteria. The dataset is provided courtesy of TFL Open Data (Find license at the end of the report), and as a government organisation, one would hope that they provide good quality data.

- Precision
    - When viewed as a plot, the data is of a cyclic pattern which is to be expected of data that is separated by the hour as we will expect periods of low and high uses depending on the time of day.
- Accuracy
    - As we do not have the true values at hand, we cannot confirm the accuracy; however, the fact that we get exact statistics for each hour period (not averaged into 100s or 1000s) is an indicator that the data is accurate.
- Bias
    - As the data is collected and aggregated from the Greater London area, there should be no bias in the number of bikes rented - and weather data itself cannot be biased.
- Consistency
    - The weather data is consistent as all measures follow the same enumeration pattern for weather codes, and we can see that on average, each data point follows the hypothesised cyclic flow.
- Completeness
    - Aside from listing “Freezing Fog” as a potential weather condition while it is not present in the dataset, all data points and measures are present and accounted for.

3.2 Regression

The first modelling method we investigated was regression. Regression is a useful modelling method to visualise any clear trends between variables, in the case of our report we will be looking to find any trends between the input variables within our dataset and the bike count.

Preparing our data for regression modelling meant considering how different variables were measured in different units. Furthermore, we wanted to consider whether we wanted to disregard outliers. The main reason to disregard outliers is if there are a lot of outliers that will significantly change the association, or it is due to incorrectly measured data.

1. Determine variables to use
   1.1 Discard weather_code as there is very little correlation between count and weather_code (refer to figure 1). This is most likely because whilst certain weather conditions it is to be expected for the count to be lower (i.e. for thunderstorm), for most weather conditions the count, as high variation

   1.2 Disregard is_season as only four variables are making it hard to find a correlation, as well as that values, show high variation in bike count

2. Convert datestamp to a timestamp column - we don't expect the exact time and date to be reappearing, assigning each row value from 1-24 representing the corresponding hour.

3. Determine the type of each field
   3.1. Numeric
   3.2. Symbolic
   3.3. See below

Timestamp	t1	t2	windspeed	Is_weekend	Is_holiday	cnt
Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric

4. Numeric vs Binary Fields

Timestamp	t1	t2	windspeed	Is_weekend	Is_holiday	cnt
Numeric	Numeric	Numeric	Numeric	Binary	Binary	Numeric

5. Numeric Fields
   5.1. Standardise all values, as the different variables are measured in different units

6. Separate dataset by weekends and weekdays, as we expect different values for each

7. Find the number of outliers (outliers we consider if they are either 0.95% higher or lower than the mean value for a variable) – if outliers cause a significant change in the assumption or it is clear there is a mistake in the recording of the data

7.1. Include outliers – few outliers found relative to the number of entries, will not impact assumptions. Furthermore, no reason to assume there's any miss-entered data, so no reason to discard outliers

3.3 Time Series

Time Series was used in this project for analysing time-series data to extract meaningful patterns and trends in the data. It made sense to use this method during the project as the dataset included a timestamp, facilitating the grouping of data to investigate potential correlations further. The main objective of pre-processing time-series is to make sure we do not create a model sensitive to an exact date.

1. Determine variables to use:
    - 1.1. Will only use all variables except for datestamp, due to not wanting our model to take into account exact dates, and season as we believe seasons are too long to obtain any meaningful patterns and trends that can be used to predict the bike count
    - 1.2. Will keep is_weekend and is_holiday as we want to differentiate time series based on this data

2. Determine whether or not time fields are classical or point processes
    - 2.1. The dataset we used is an example of a classical data (a sequence of values taken at successive equally spaced points in time. Data is taken at hourly periods

3. Extract a yearlong worth of data – this is to speed up the processing time
    - 3.1. Dataset did not start at the 1st of the year so must make sure to extract 365 days' worth of data

4. Implementation and use of bins as a pre-processing method
    - 4.1. Using old dataset, create a new copy however group each entry into bins of 3-hours (this allows us to capture rush hours easier)
    - 4.2. Derive new variable "timestamp" – with values 1-8 representing the bin it belongs to. This variable will be cyclical.
    - 4.3. For each new binned entry, use of Mode (most occurring weather code) in Weather_code to represent a weather condition for each bin.
        - 4.3.1. In cases which three different weather code appears we decided to choose the weather code which appears in the second hour (middle) of each period as it is more likely to be the prevalent type of weather for the segment.
    - 4.4. For the temperature, how the temperature feels, humidity and wind speed, find the mean value throughout the three-hour period.

5. Determine the type of each field
    3.1. Numeric
    3.2. Symbolic
    3.3. See below

cnt	t1	t2	hum	windspeed	weathercode	isholiday	isweekend	timestamp
Numeric	Numeric	Numeric	Numeric	Numeric	Symbolic	Numeric	Numeric	Numeric

6. When running the model, must select the frequency – this represents the number of seasons you expect to see with the data given

3.4 Machine Learning

DNN's seemed like the right choice for our dataset as it will produce a list of variables ranked by their importance when determining the output and therefore, we can verify which factors are the most impactful when determining the bike count for the next day. We prepared the data in line with the following plan to make it amenable to deep learning,

1. Discard the date field. Not relevant to the training as we'll only need to look per hour.

2. Convert the timestamp field into a pair of sin/cos to make it amenable to network input.
    2.1. Currently, our data has a row for each hour of the day, and the NN has no way of interpreting what time is, so it is important that we encode it with some continuous value
    2.2. Time data is cyclic, use a function to get it between 0 and 1, that is when it is 00:00 sin = 0, cos = 1 and vice versa.
        2.2.1. Firstly, determine the number of seconds in a day. Let Max_Seconds equal this.
        2.2.2. For the x-axis, use sin(2*pi/Max_Seconds)
        2.2.3. For the y-axis, use cos(2*pi/Max_Seconds)

3. Determine the type of each of the field
    3.1. Numeric
    3.2. Symbolic
    3.3. See below

cnt	t1	t2	hum	windspeed	weathercode	isholiday	isweekend	season
Numeric	Numeric	Numeric	Numeric	Numeric	Symbolic	Numeric	Numeric	Numeric

4. Numeric Fields
    4.1. Determine whether they are Ordinal or Discrete
        4.1.1. See below

cnt	t1	t2	hum	windspeed	weathercode	isholiday	isweekend	season
Ordinal	Ordinal	Ordinal	Ordinal	Ordinal	Symbolic	Discreet	Discreet	Discreet

    4.2. Separate the ordinals into their own dataset
        4.2.1. See below

cnt	t1	t2	hum	windspeed
Ordinal	Ordinal	Ordinal	Ordinal	Ordinal

        4.2.2. Merge the converted timestamp field back into the ordinal dataset.

    4.3. Transform this dataset using z-scaling
        4.3.1. (Datatechnotes.com, 2018).

4.3.2. It calculates how far away a value is away from its mean and places it between a sigma range (based on the variance of data).
4.3.3. It can be calculated using **z = (x - μ) / σ**
4.3.4. **μ** is the mean of the column, **σ** is the standard deviation.
4.3.5. Essentially it produces a normal distribution graph
4.4. Transform z-scored data frame between 0 & 1
4.5. Ensure both ordinal and discrete values are between 0 and 1

5. Symbolic Fields/Discreet
5.1. These fields are transformed using a 1-hot encoding. This means that each literal will have its own column and will be between 0 and 1
5.2. Weather code consists of 7 different literals (“Clear”, “Broken_clouds”, “Few_Clouds”, “Broken_Clouds”, “Cloudy”, “Light_rain”, “Thunderstorm”, ”Snowfall”)
5.2.1. After applying this transformation, each of these will have their own column

6. Combining the datasets into one
6.1. Now all the fields are in the correct format for processing we need to concatenate them into one dataset.

7. Splitting data
7.1. Determine the split factor:
7.1.1. 70/30: Training/Test
7.2. Training set consists of the first 70% of the dataset
7.3. The testing set consists of the last 30% of the dataset excluding the count column as that is the output column.
7.4. Separate the output column into a new expected dataset.
7.5. Once the neural network has finished, we will compare the predicted count values against the expected dataset.

8. Training data is then randomised to remove any patterns that may already exist.

9. From the z-scored dataset determine the Mean, Max and Min for the count column to revert z-scoring and 0 to 1 bounding from the neural network results.

4. Model Development and Evaluation

When deciding which models to move forward with, it was important to analyse the pros and cons of each method. As a collective, we investigated the following modelling methodologies:

- Regression
    - Advantages:
        - Quick at providing a simple benchmark,
        - Easy to identify outliers or anomalies
        - Provides the ability to look at how more than one predictor value influences your expected outcome
    - Disadvantage:
        - Doesn't work well with large datasets
        - Incomplete data can lead to false conclusions
- Decision Tree
    - Advantages:
        - Works with small and large datasets
        - Easy to interpret results
        - Results are displayed as rules
        - Can handle both numerical and categorical data
    - Disadvantage:
        - Too simplistic
        - Tends to over-fit data
        - May miss relationships due to splitting
        - Restricted to one output attribute
- Deep Neural Networks
    - Advantages:
        - Works well with discrete values
        - Works better with larger datasets
        - Provides a list of variables ranked by their importance
        - Improved generalisation
    - Disadvantage:
        - Models can be complex and difficult to interpret
        - Can overfit data
        - Heavily Biased by initial weights
        - Requires significant computing power for business solutions
- Time Series
    - Advantages:
        - Enables greater understanding of past behaviours (with the inclusion of trend, seasonal and random)
        - Helpful for future predictions
        - Works well with a timestamped dataset
        - Important for trending type data
    - Disadvantage:
        - Doesn't always give a perfect conclusion
        - If there are many factors, forecasting may become unreliable
        - Observations are not mutually independent

        - Only useful for timestamp data
- Recurrent Neural Networks
    - Advantages:
        - Reliable and converges well
        - Have “memory.”
        - Allows the ability to look back through data for sequences
        - Requires large amounts of data
    - Disadvantages
        - Difficulties learning long-term dependencies
        - Gradients tend to vanish, which leads to weight becoming less impactful
        - Limited to looking back only a few steps
        - Slow to train

After carefully analysing the pros and cons of each model and comparing them with our data, we decided to move forward with the following models:

- Regression
    - As the data consists of many predictors, we can use Multi-Linear Regression to look at how more than one predictor value can predict the output. Can also use non-linear regression to try and emulate the data is also not too large and is not missing any values reducing the chance of a miscalculation.
- Time Series Analysis
    - Considering the dataset is timestamped, this method was useful to identify direct trends in the data and help formulate future predictions.
- Deep Neural Networks
    - Since the data contains lots of predictors, it would be interesting to find out which of them is the most significant when it comes to predicting the number of bikes to be released the next day. There is no issue with computation power and overfitting can be prevented.

4.1 Regression

4.1.1 Development

Regression is a simple supervised learning algorithm, whch allows us tp examine the strength of the relationship between an independent variable and dependent variable, the dependent variable, in this case, is the number of bikes (or count). Regression was chosen as it is a good starting point to view any relationships with the number of bikes rented and other variables. We believe there should be some correlation (relationships) between count and other variables within our dataset. While we do not expect any linear correlations to exist with count and any one indiviual variable as there is multicollinearity between predictors (for different predictors to have relationships with one another, such as t1 and t2), there's a chance for there to be a non-linear regression with certain variables as well as multi-linear correlation when combining two variables. To summarise, the main objective for using regression is to use multi-linear and non-linear regression to bring up any clear and obvious relationships between count and the other variables in our dataset.

Multi-Linear regression development

To train the data in multi-linear regression, we initially started using 70% of the dataset. We had to be aware that a training dataset too large could potentially risk overfitting the algorithm, whilst a dataset too small will not give us enough values to create an accurate predictor. The most important process for multi-linear regression was selecting the appropriate combination of variables to try and predict count. We chose only to have two independent variables as we felt the main benefit of regression is it gives a clear visualisation of trends, and when we input too many variables, then regression would not be the best algorithm for this.

We also were curious as to how the variable "weekend" impacted the regression model. Because of this, we decided to create models using only weekend and weekdays respectively, as well as a model using all days.

Non-linear regression

For nonlinear regression, we would take extra care in selecting a polynomial that would give us the best results while not overfitting the data. We would train a regression classifier to predict count using different input variables and by using the different performance evaluation methods MAE, RMSE and R^2, we could see how reliable the classifier is and thus infer which variables have the strongest relationship with the count of bikes.

We chose to investigate non-linear regression as we felt that certain variables such as time might display a non-linear relationship with the count. For example, we expect there to be peak hours during certain points in the day, which is not a linear relationship. We tested how training dataset size impacted results. We noted that when we set training data to a large size of 90%, the prediction line was far above the test dataset (displayed in figure 4). When decreasing the training dataset size to 50% of the official dataset size, we noticed a very poor MAE and RMSE value, showing that this is not enough data to get an accurate fit. Because of this, we found that 70% was an appropriate dataset size.

Like multi-linear regression, we also split up our regression models into weekends and weekdays. We did not run tests for both, however, as this provided no new information. Rather is just gave results in-between what we got for weekday and weekends.

When deciding polynomial values, we found that for timestamp, as we expect there to be multiple peaks during the day, we needed a relatively high polynomial count of 9 for weekends and weekdays. This implies we see a change in peak approximately every 4 hours. We found increasing it to 11 might very little difference in results. Thus, to avoid overfitting, we kept it as low as possible. We also found that decreasing the polynomial to 6 almost halved the R^2 value (please refer to figure 3). For other predictors, we didn't expect as many peaks; thus, we did not need as high of a polynomial value. After looking into it we found that a polynomial order of 4 was all that was needed to get the best prediction for these results, any higher made little difference, whilst any lower made a major difference in R^2 values.

4.1.2 Evaluating Results

When it comes to evaluation results, the process was identical with multi-linear regression and non-linear regression. Once the code has finished processing the training set and has created a model, we will plot the model on the graph, in non-linear regression, this will be a curved line while with multi-linear this will be a plane on a 3d graph.

In either case, the Y-axis will represent the output variable "count". We will then plot the data points of the testing dataset against the model. We can then compare the data points, with what the model "predicted".

Using the difference between the model and the testing data, we can then calculate MAE, RMSE and R^2 (Refer to appendix D for more information on these values). We can judge that predictor has a stronger relationship if MAE and RMSE are lower while R^2 is higher. MAE and RMSE help measure the variable of interest, RMSE being the "standard deviation of the unexplained variance". MAE gives lower weight to large errors when compared to RMSE. R^2, on the other hand, is the measure of fit of the training dataset. To measure MAE and RMSE, we will use a proportion of our dataset that was not used to train the formula and compare the difference between the given count values and the predicted values.

R^2, on the other hand, is a "goodness of fit" measure, not necessarily a "goodness of prediction". This means it tells you how well the regression graph matches the values it was used to train. This means to calculate R^2; we use the training data as opposed to the testing data. Because of this, it is useful to find R^2 as well as values for MAE and RMSE.

4.2 Time Series

4.2.1 Time Series Development

Time series was mainly applied to see trends and patterns within the dataset, as we expect to see some reoccurring patterns such as two peaks on weekdays representing rush hours traffic. During the development process, we initially started with deciding on how to encode the time data, as we did not expect our time series to be reliant on a specific date, more the time of the day. We considered encoding the time of by transforming our timestamp into angles, which would work as our data is cyclic. However, we opted to bin into periods as we felt any trends would be made clearer when binning into periods, as we could capture rush hours periods (where we should see the greatest change) into single bins.

Other developments that occurred while testing was separating out between weekends and weekdays, as we found that the trends being seen were a fair bit different when comparing weekends, holidays and weekdays. On weekends we noticed there would only one peak that would occur during mid-day, as opposed to the two peaks being witnessed on weekdays.

The main parameter when deciding time series was the frequency. Frequency is meant to represent how many entries do we expect to see a repeating pattern in. It represents the number of seasons. I.e. if the frequency is set to 8, then we expect to see a new season every 8 values. Thus, if the dataset is 160, we expect to see 20 seasons. We initially started looking at daily seasonal patterns, starting with two weeks' worth of data, before moving to a month's worth of data, to a year. This was to see if we saw much change if the dataset size increased – we did not expect this to happen especially for weekdays as we expect the time series trends to remain consistent, two peaks during rush hours. There might be an overall increase however in the count. Unfortunately, a year contained too much data; thus, it was nearly impossible to see any pattern as it was too condensed. We did the above with only weekends and weekdays. Following this, we checked to see if we could find weekly seasonal patterns, with weekends and weekdays included, to see if we could find weekly patterns.

4.2.2 Time Series Evaluation

The purpose of time series was not to necessarily retrieve scores, but more to find trends within data. During the evaluating phase of time series, we made use of the time series plotting function used in the lab. This allowed us to view the actual data alongside the trend in data, seasonal periods and data noise. By using this function, we were able to identify patterns in the data.

The expected output would be a model, with the observed values of count, below would be the underlying trend of the values. The model will also show the seasonal repeating patterns based on the fixed period. The final part of the model will display the residuals, which will show us the observed values once we've removed the seasonal and trend data, indicating the essence of this structure. We can then observe trends and compare with our hypothesis.

To evaluate the data, we would use the same seasonal patterns (i.e. daily) however, with a larger initial dataset, meaning we would see more seasonal patterns. If the model is

accurate, then the seasonal patterns should remain consistent between models. This is a good opportunity to see general patterns in bike count usage over the day, as well as perhaps seasonal trends. Finally, we can also look at the trends in bike count and see how similar it is to trends for other variables, this information could be useful is seeing what variable impacts the yearlong trend of bike count the most.

4.3 Neural Networking

4.3.1 Network Development

Following a considerably low r^2 result for our regression work, we decided to use a deep neural network.

We did not want to use any input dropout as that could potentially lead to our even distribution of data along each hour interval being upset. The feedforward nature of the deep neural network was attractive as we hoped that it would learn the cyclic pattern of the original data to make more accurate predictions relative to the previous time (as we do not want extremely large shifts in results, instead of a more natural flow.) We implemented the Regression variant as it is amenable to time-series-esque data, and as our data is separated by the hour - it seemed like a straightforward choice.

We initially opted to use the tanh activation function, as we felt its unique characters of the zero inputs tending to be mapped near to zero would help it better adapt to the circular nature of our data. However, we later decided to use a ReLU (Rectifier) activation function as its linear activation function better suited the periodicity of the data.

We did not include any dropout for ReLU, despite its reputation of potentially failing to continue to learn earlier than usual. However, we did not encounter such scenarios and found that adding dropout instead greatly reduced the performance of the network.

However, recognising that ReLU *does* have its issues and the typical solution – Leaky ReLU – does not help, we instead select the final activation function “Maxout”. This is a compromise between the two ReLU functions and allows us to edge out a bit more accuracy with the model. Maxout selects the maximum of the inputs provided to the hidden neuron. We further add controls both the adaptive learning rate time decay factor and adaptive learning rate time smoothing factor to mitigate overfitting.

We have added some further controls for the adaptive rate learning of Maxout through rho and epsilon to help mitigate any overfitting and improve learning.

We recognise that relative to the data, our number of neurons is relatively low, but we attempt to avoid overfitting by adding a second hidden layer. Furthermore, we did not find much improvement in both r^2 values of our training and testing values when running with a higher set of hidden neurons (which even reduced performance).

The number of epochs selected was arbitrary, set to approximately 300 after trial and error following extreme diminishing returns. However, in most runs, the moderate limit for no training improvement to cut off model training kicked in as the learning plateaued at 150.

4.3.2 Evaluating Results

When retrieving the count results from the neural network, they are returned in the same form of the count data that we trained the neural network with - z-scored and bound between 0 to 1.

Therefore, the first steps we must take to evaluate the results must be to first revert the 0 to 1 bounding and then remove the z-score from the data.

To revert the 0 to 1 scaling we simply take the local minima and maxima of the pre-scaled z-scored dataset and invert the scaling using the local maxima and minima of it (((maxv-minv)*x)+minv).

To remove the zscoring , we take the original z-scored matrix and extract two features from the count column, scaled: scale and scaled: centre which were produced with the original z-scoring operation. We can then retrieve the un-zscored, un-0to1 results with the equation “real = un0to1_results * scaled: scale + scaled: centre”. We applied the same method to our expected data during experimentation to ensure that our approach returned the correct values.

Once we are in possession of the real results from the neural network, we can compare them against our raw expected results.

As an exploratory measure, we wanted to plot all of our predicted values against our expected values to ensure that we saw the expected cyclic pattern that is present in our non-randomised data. However, due to the large size of our dataset and therefore, the number of returned values, we struggled to plot them successfully in R Studio due to memory issues. To mitigate this, we instead wrote our large graph comparing our expected to predicted values to disk.

To evaluate the neural network, we generated three sets of metrics: RMSE, MAE and r^2.

We recognise that the r^2 statistic is not the be-all and end-all measure of how well a model predicts, so we wanted to explore other measures of ranking our model’s performance.

We investigated using the F-statistic to provide us with further insight into our results. It can be calculated for any data set and can be used to explain variance. However, we found that this is only applicable for linear models that meet all of the assumptions made in the hypothesis stage. Since none of our models produce linear models, it will not be used in the following investigation.

Instead of using the F-statistic, we briefly caused the other measures not yet evaluated and compare them against the other approaches the group developed to determine relative performance.

5. Results / Conclusion

5.1 Regression

5.1.1 Multi-linear regression

Below is a table of results for our multi-linear regression tests, specifying the two predictors we used, whether we included weekends, and the MAE, RMSE, R^2, MSE values.

(Note Y in weekend column means we only used weekends, N means we only used weekdays, and B means we used both)

Predictor1	Predictor2	Weekends Y / N /B	RMSE	MAE	R^2	MSE	Figure
T1	T2	Y	0.28	0.34	0.36	0.0784	5
T1	T2	N	0.56	0.44	0.13	0.3136	6
T1	T2	B	0.5	0.41	0.18	0.25	7
T1	Wind Speed	Y	0.93	0.76	0.36	0.8649	8
T2	Wind Speed	Y	0.95	0.78	0.33	0.9025	9
T1	Timestamp	Y	1.07	0.92	0.37	1.1449	10
T2	Timestamp	Y	1.06	0.9	0.33	1.1236	11
T1	Wind speed	N	0.83	0.67	0.13	0.6889	12
T2	Wind speed	N	0.82	0.67	0.12	0.6724	13
T1	Timestamp	N	1	0.87	0.13	1	14
T2	Timestamp	N	1	0.86	0.12	1	15
T1	Wind speed	B	0.85	0.68	0.17	0.7225	16
T2	Wind speed	B	0.85	0.69	0.16	0.7225	17
T1	Timestamp	B	1.02	0.88	0.18	1.0404	18
T2	Timestamp	B	1.01	0.87	0.16	1.0201	19
Timestamp	Wind speed	Y	0.84	0.7	0.05	0.7056	20
Timestamp	Wind speed	N	0.9	0.75	0.02	0.89	21
Timestamp	Wind speed	B	0.89	0.74	0.02	0.7921	22

As you can see from the results above, the two input values with the highest R^2 and lowest RMSE and MAE are T1 and T2 on weekends only. This implies that the two values combined values that impact the count of bikes is actual temperature and real feel temperature on weekends. Judging from the graph in figure 5, it seems as when both T1 and T2 are higher, and the bike count usage is expected to be higher. Also, when comparing weekday values, T1 and T2 seem to have the strongest relationship with count. A possible explanation for why temperature has less impact on weekdays is because people may use the bike's as a main form of commute; thus, they are less likely to be put off by poor weather conditions. While on weekends, people are more likely to use Santander bikes when the weather is warmer. With that in mind, when the weather is too hot, people may find riding bikes too intensive, which will decrease the bike count and maybe why the results aren't as linear as we'd expect. Interestingly, whilst temperature has the biggest impact on weekends, this is the only time where the weekend results values for MAE and RMSE are lower than the weekday results. Even when we combine the temperature predictor with another predictor such as wind speed, the results are still stronger on weekdays (weekend values for T1 and wind_speed are MAE: 0.76 RMSE: 0.93 while the weekday values are 0.83 and 0.67 respectively), while R^2 remain consistently higher on weekends. The likely reason for this is because R^2 does not take into bias, and the data on weekends could be more bias, meaning weekend values could very well be consistently higher or lower depending on

unexplained variance. Perhaps on days where daylight is shorter, it's common to get values lower than expected during early and later hours while daylight hours are higher. Whilst on weekdays, again people often use bikes as a main source of commute, so it's unlikely to be affected by the variance as much. Thus, MAE and RMSE tend to be more consistent.

Other observations are that the real feel and observed temperature makes little difference for the predictor.

To sum up this section, based on the data found, we couldn't find any clear periodic bike usage counts based on multi-linear regression. The closest we could find was how, when the observed real feel temperature is higher, the bike count usage is expected to be higher. It appears MAE, RMSE and R^2 is always higher on weekend predictions except for when predicting only T1 and T2, implying there is more bias during the weekends.

5.1.2 Non-linear regression

When looking at non-linear regression, as mentioned in the data processing section, we chose to if we could predict count using T1, T2, wind_speed and timestamp, separated again by weekends, weekdays. We chose not to look at both as often the result is a combination of both. A table of results is below.

Predictor	Weekends Y / N	CPoly	MAE	RMSE	R^2	MSE	Figure
T1	Y	4	0.49	0.64	0.45	0.4096	23
T2	Y	4	0.49	0.64	0.44	0.4096	24
Wind speed	Y	4	0.57	0.71	0.04	0.5041	25
Timestamp	Y	9	0.73	0.88	0.07	0.7744	26
T1	N	4	0.71	0.99	0.14	0.9801	27
T2	N	4	0.71	0.99	0.14	0.9801	28
Wind_speed	N	4	0.76	1.01	0.02	1.0201	29
Timestamp	N	9	0.88	1.13	0.12	1.2769	30

The results above seemingly agree with what was found with multi-linear regression in that, while no one variable has a deciding impact on count, temperature on weekend has the strongest relationship. With this data here however whilst T1 and T2 have slightly higher MAE and RMSE than they were in multi-linear when combined, they also have slightly higher R^2. This again could likely be due to bias, however in general neither are too useful predictors. It makes a lot of sense for these two variables to have similar results as their multi-linear counterpart as T1 and T2 have a very linear relationship with one another (please refer to figure 2). Moving further ahead, very few of these predictors gave good scores.

In general, this seemingly implies no single predictor has a strong relationship with count (similar to multi-linear regression), and thus no single predictor can be used for non-linear regression to find a trend in count. The only real knowledge we can gain from these two tests was that temperature seems to have the biggest relationship with count, as well as that weekends differ to weekdays in prediction models. A possible reason for this is due to the dataset being too large, and regression, both multi-linear and non-linear regression are known to struggle with large datasets. One potential way to have overcome this in future experiments would be to have used k-fold cross-validation. This would require resampling the data by shuffling the dataset then split into k datasets, and then perform a regression

analysis between each dataset. Unfortunately, due to time constraints, we lacked time to reach this.

5.2 Time Series

Due to timestamp playing a key factor in our dataset, we have shown that this model can be quite good at presenting common trends when used effectively. However, due to the many factors responsible for the increase in the number of bikes used, we concluded that this might not be the best method.

Taking in consideration factors such as time of the day especially for rush-hour we can consistently see a pattern in numbers of Santander bikes increasing at two points of the day (figure 49, 50), which mostly likely coincides when people commute to and from work, in comparison to the rest of the day. The fact that we got the same seasonal repeats over different periods helps consolidate this idea that bike count is impacted by the time of the day.

We can further analyse other factors throughout the year, such as weather (temperature, wind, & etc.) and holidays. However, we occasionally came to an inconsistent set of results due to multiple factors affecting count within the same time period. A table listing the different factors and a figure numbers to view a comparison is shown below.

Variable	**Figure**	**Brief comparison of trend vs count trend**
T1	Figure 55	Trends for both count and t1 follow a very similar pattern, implying t1 does have an impact in bike count
Hum	Figure 56	Peak in trend for bike count coincides with a low point in trend of humidity, this seems the only real visible correlation
Wind_speed	Figure 57	Little similarities in trend in wind_speed and count, other then the highest point in trend of wind_speed coincides with a low point in bike count.
T2	Figure 58	Trends for both count and t2 follow a very similar pattern, implying t2 does have an impact in bike count

We were unable to find any other observations with the remaining variables.

Weekends and holidays were shown to be much more difficult to predict, although it seems as though there's a more gradual increase in weekends before hitting one peak (figure 50). Furthermore, figure 52 shows the weekly repeating pattern, which as expected shows 5 days with two spikes each day, representing weekdays, while having two more days with one peak each. With figure 52 showing the same seasonal repeating pattern as figures 49 and 50 however at a lower frequency, this helps validate the assumption that you will see peaks twice a day on weekdays, and once a day on the weekend.

A clear time-series graph was created using a whole year worth of data and analysing the overall trend in count across this period. We also looked at how other variables affected trend over the year, temperature was the only one where we saw a trend that seemed to follow that of bike count. As expected, we can see a gradual increase starting from march as we get closer to summer (increase in temperature) a significant number of count increase, whereas during the winter, people tend to use these bikes less (figure 54). Although this can be a vague assumption to make without considering other factors simultaneously, it is expected a decrease in number of cyclists in general during colder time periods.

However, this is perhaps not enough information when determining trends with bike count. Whilst timeseries was good at displaying overall trends and seasonal repeating patterns, for actual predictions on values, there are most likely better methods.

5.3 Deep Neural Network

Our deep neural network produced the most consistent results of our collection of models. This is likely due to the added complexity of the network and training patterns which far better models the data then any traditional approach tried previously.

Despite the model reaching a strong r^2 value, we do note that it struggles to accurately model peak time usage (though still predicting a higher than usual spike of activity).
As seen in the table below we find that our most important values are the time of day which would largely reflect the ebb and flow of usage in London and coincides with our hypothesis. Apart from time, whether it is a weekend and how humid or hot it is are the greatest secondary predictors for our model. While weather is lower than expected on the variable importance table, we hypothesise that this is due to the close relationship between temperature and weather.

Rank by importance	Variable	Relative_importance	%
1	x(time)	1.000	0.116
2	y(time)	0.647	0.075
3	isweekend	0.544	0.063
4	hum	0.472	0.055
5	t1	0.466	0.054
6	season_0	0.451	0.052
7	weathercode_clear	0.436	0.050
8	season_1	0.431	0.050
9	season_3	0.407	0.047
10	weathercode_thunderstorm	0.403	0.047
11	windspeed	0.402	0.047
12	weathercode_fewclouds	0.401	0.047
13	isholiday	0.393	0.046
14	weathercode_cloudy	0.382	0.044
15	weathercode_snowfall	0.379	0.044
16	weathercode_brokenclouds	0.368	0.043
17	weathercode_lightrain	0.358	0.042
18	t2	0.351	0.041

19	season_2	0.336	0.039

We further note that a peculiar ceiling / floor line appears to trend across the plot of expected v predicted as a result of switching to ReLU activation (Figure 32 compared to Figure 36). This is interesting as it appears that these would be errors despite being on the mean and our r^2 is higher than the previous activation function. Once we switched to the Maxout activation function (Figure 39) we no longer saw those trends and therefore we know that the model is no longer overfitting to the training data.

When comparing the rate of training versus validation the graph across the three activation functions (Figure 31, Figure 35, Figure 38) it holds the same curve between the offset which indicates that our training is performing as expected across any of the activation functions.

In conclusion, this model is the best predictor we have managed to produce (using the Maxout function) throughout the course of this project. We can predict the number of bikes used per hours with accuracy outside of peak times. Inside of peak times, we still predict large spikes in bike usage however struggle to match high outliers in our testing validation dataset. (Figure 34)

For business applications, given a weather forecast for the next day, we can inform with relative confidence the number of bikes to be in use per hour, allowing for more fuel-efficient morning deliveries.

However, we must consider the model's error in determining if it is fit for purpose. Looking at our residuals (Figure 39 and Figure 40), we can see that we fit the graph with great accuracy for most occasions. Considering our relative RMSE and MSE we approach zero with some degree of closeness and consider these deviations to be well within the acceptable error for a business situation.

We should also consider that due to the wide range of results that we expect the network to produce that we should considering the visual data of the residuals when judging it instead of our RMSE and MSE statistics. Most deviation from the expected results occurs during the highly variable peak time usage, and as such they are likely the source of our error values.

Apart from these, the model appears fit for purpose.

Appendix

Appendix A – Regression Models

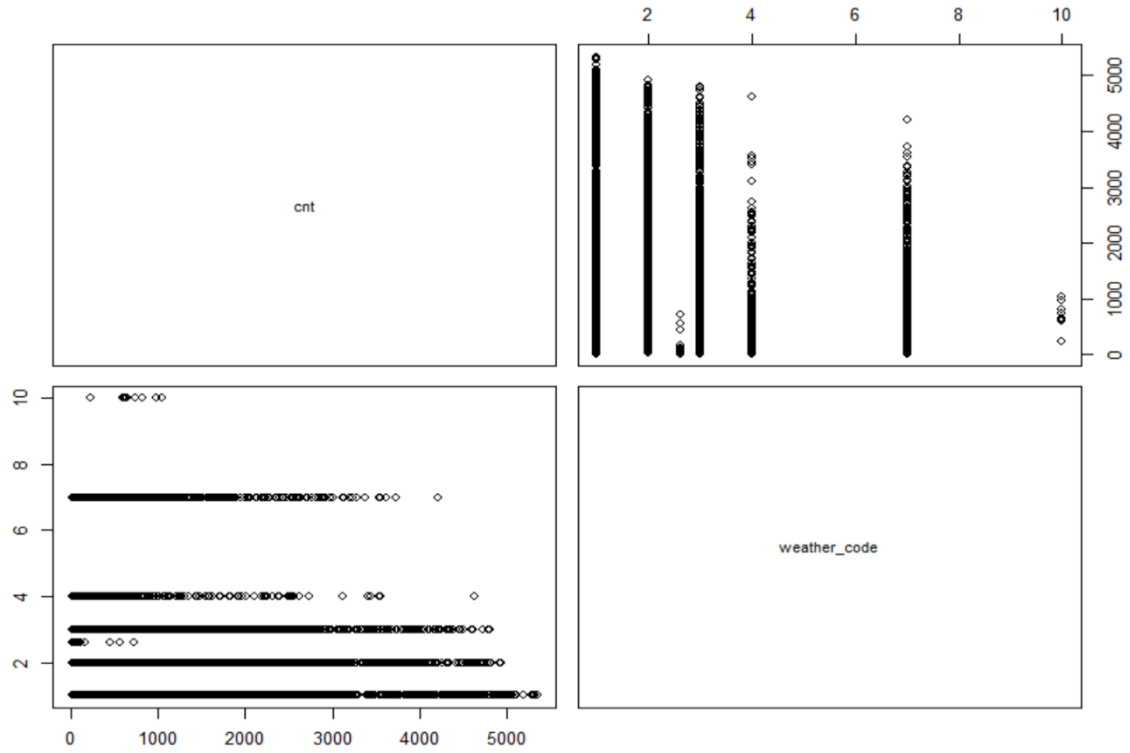


Figure 1: Correlation between count and weather_code:

Using the pairs() method within R, this displays two tables mapping count and weather_code as you can see there is very little relationship displayed. As you can see from this image there is a lot of variation between count depending on the weather_code making it hard create a regression model.

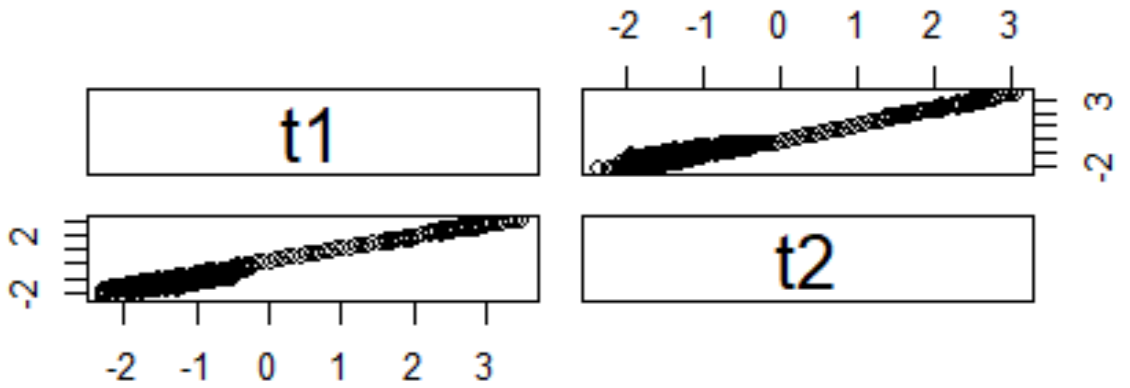


Figure 2: Correlation between t1 and t2:

Using the pairs() method within R, this displays two tables mapping count and t1_code as you can see there is very linear relationship displayed

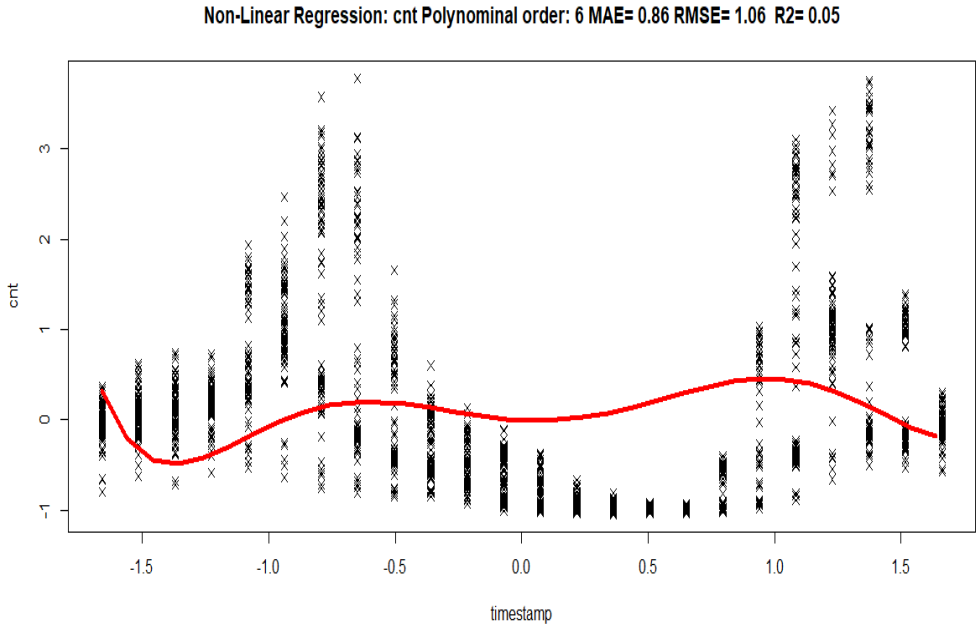


Figure 3: Example of overtrained non-linear regression model

Shows an attempt to create a non-linear relationship between timestamp and count, with polynomial order of 6. As you can see from this the error metrics is significantly worse than when the polynomial order is at 9, hence why we found 9 to be a safe polynomial order for timestamp.

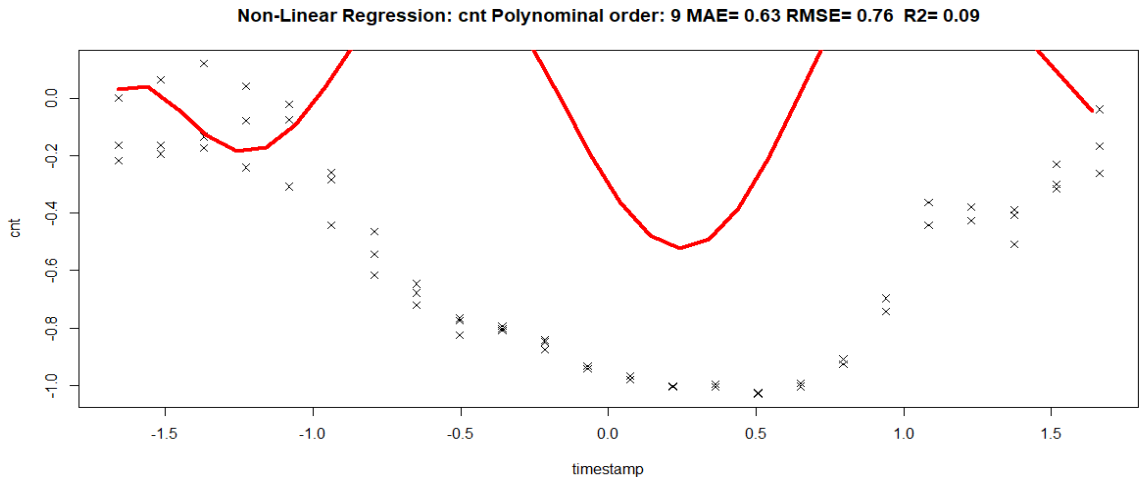


Figure 4: Poor polynomial optimisation example

Shows a non-linear regression model, with a training set, as you can see majority of predictions are way higher than expected showing an overfitted model

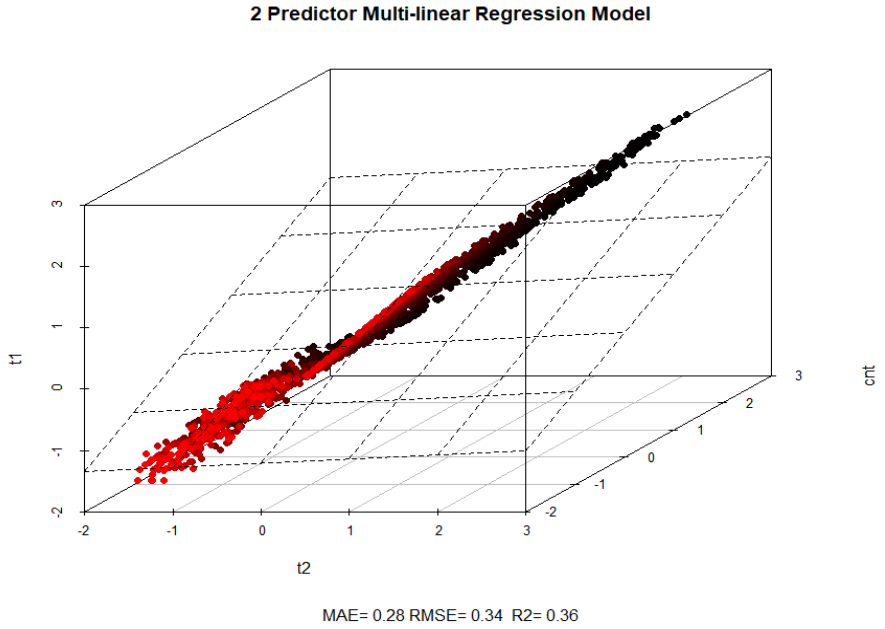


Figure 5: Multi-Linear Regression Model T1 and T2 Weekends Only

A multi-linear regression model modelling T1 and T2 against count with only weekends used for training data. Graph suggests that when T1 and T2 are higher, count should be higher as well.

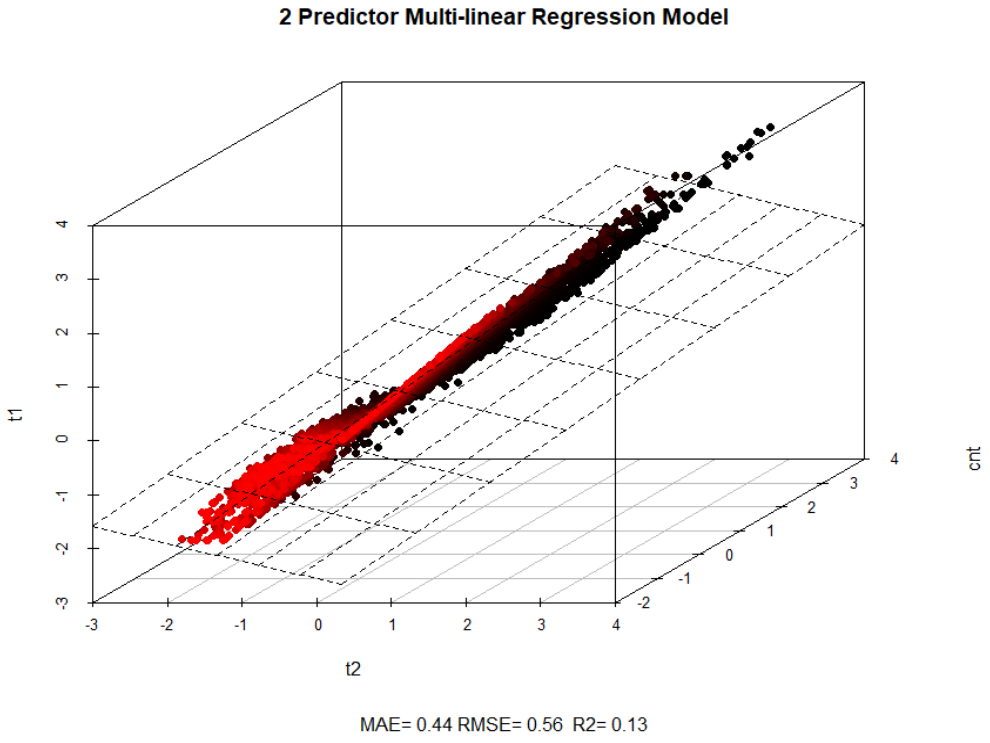


Figure 6: Multi-linear Regression Model Predictors t1 and t2 Weekdays

A multi-linear regression model modelling T1 and T2 against count with only weekdays used for training data. Graph suggests that when T1 and T2 are higher, count should be higher as well.

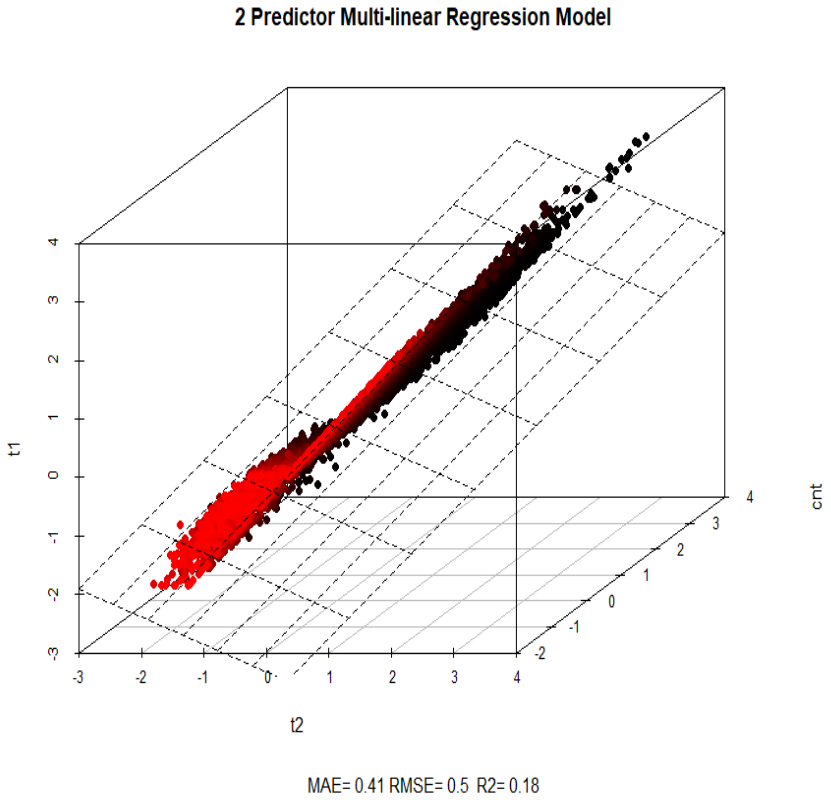


Figure 7:Multi-Linear Regression Model Predictor T1 and T2 all days

A multi-linear regression model modelling T1 and T2 against count using all types of days considered. Graph suggests that when T1 and T2 are higher, count should be higher as well.

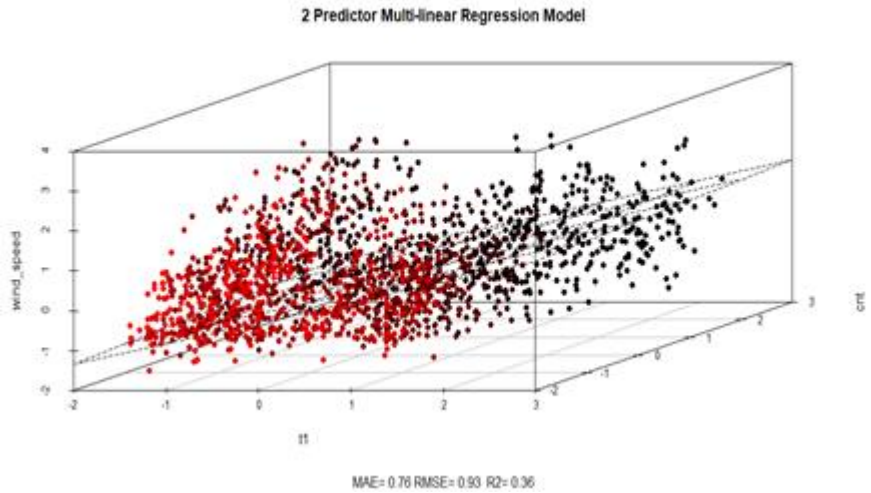


Figure 8: Multi-Linear Regression Model Predictor Wind_Speed and T1 weekends only

A multi-linear regression model modelling T1 and wind_speed against count with only weekends considered. Graph shows very scattered plots implying that there is little relationship between the three values on weekends.

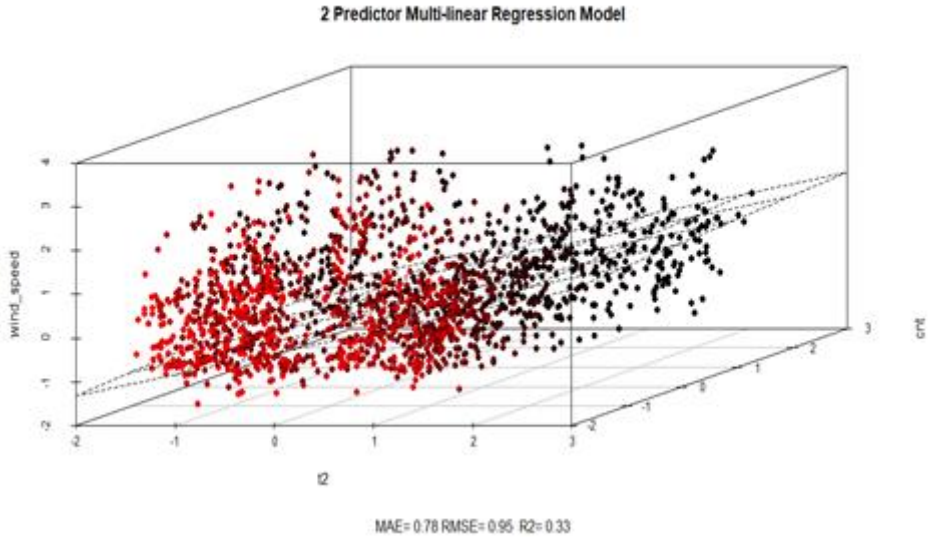


Figure 9: Multi-Linear Regression Model Predictor T2 and wind_speed weekends

A multi-linear regression model modelling wind_speed and T2 against count with only weekends considered. Graph shows very scattered plots implying that there is little relationship between the three values on weekends.

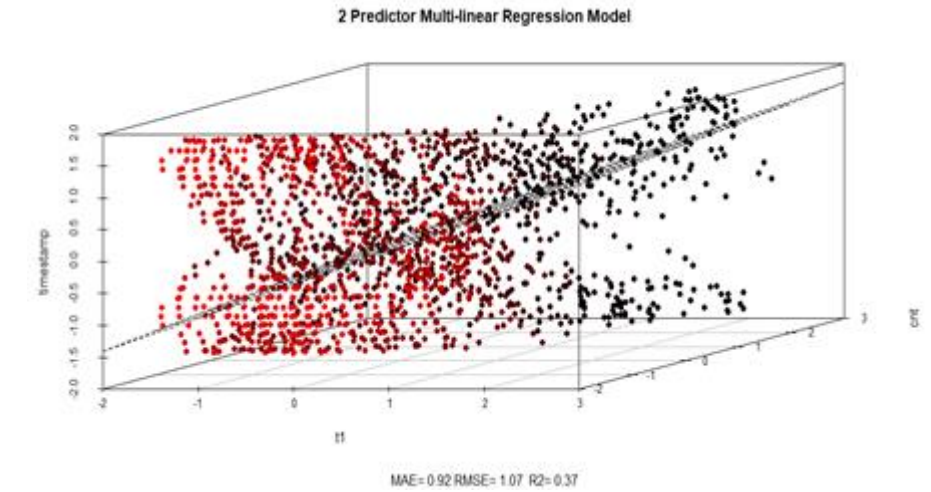


Figure 10: Multi-Linear Regression Model Predictor T1 and timestamp weekend

A multi-linear regression model modelling T1 and timestamp against count with only weekends considered. Graph shows very scattered plots implying that there is little relationship between the three values on weekends.

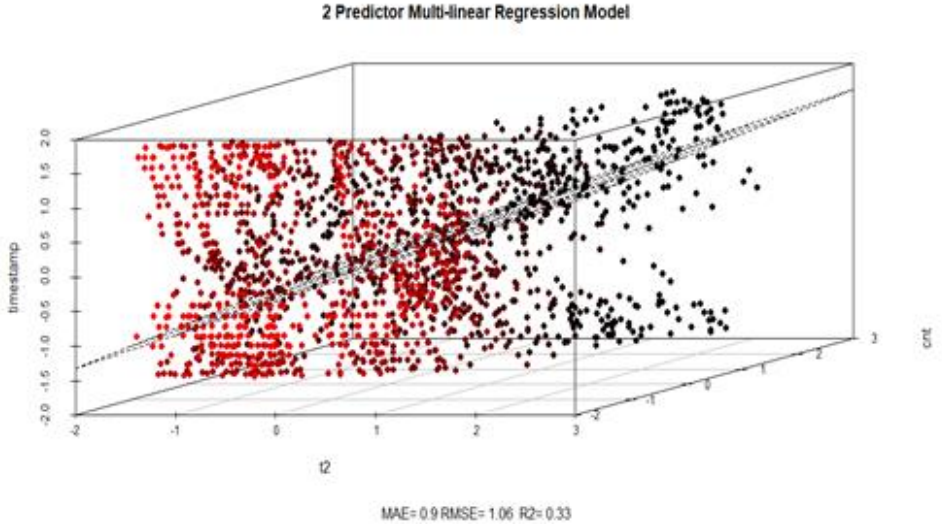


Figure 11: Multi-Linear Regression Model Predictor T2 and timestamp weekday

A multi-linear regression model, modelling timestamp and T2 against count with only weekends considered. Graph shows very scattered plots implying that there is little relationship between the three values on weekends.

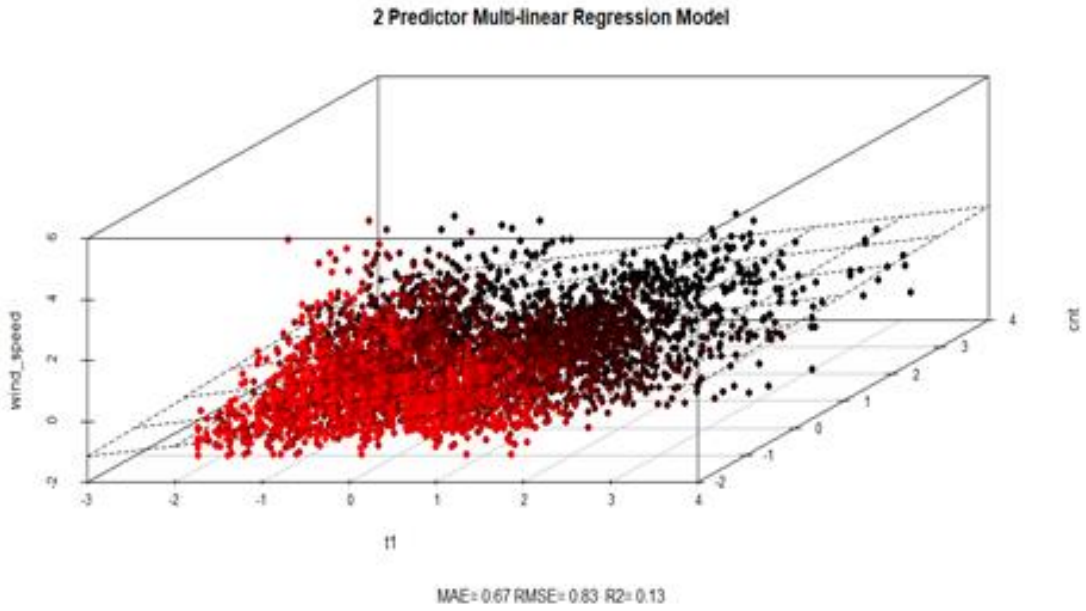


Figure 12: Multi-Linear Regression Model Predictor T1 and wind_speed weekday

A multi-linear regression model modelling T1 and wind_speed against count with only weekdays considered. Graph shows very scattered plots implying that there is little relationship between the three values on weekdays.

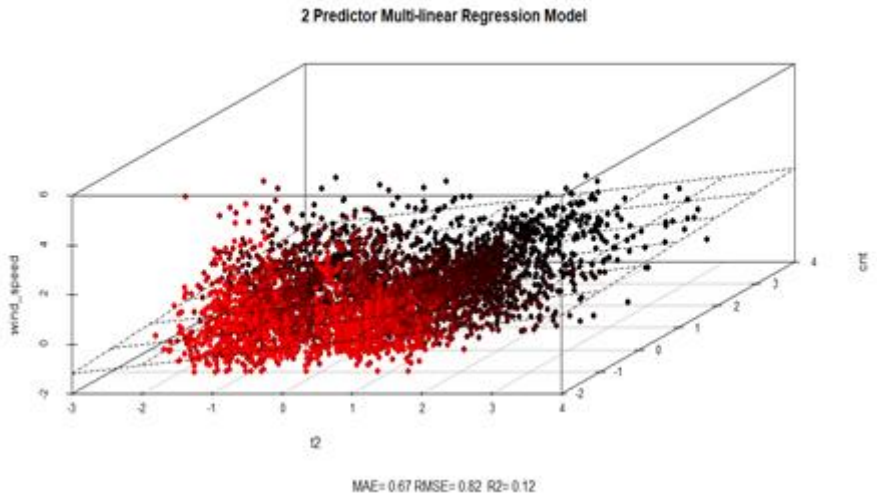


Figure 13: Multi-Linear Regression Model Predictor T2 and wind_speed weekday

A multi-linear regression model modelling T2 and wind_speed against count with only weekdays considered Graph shows very scattered plots implying that there is little relationship between the three values on weekdays.

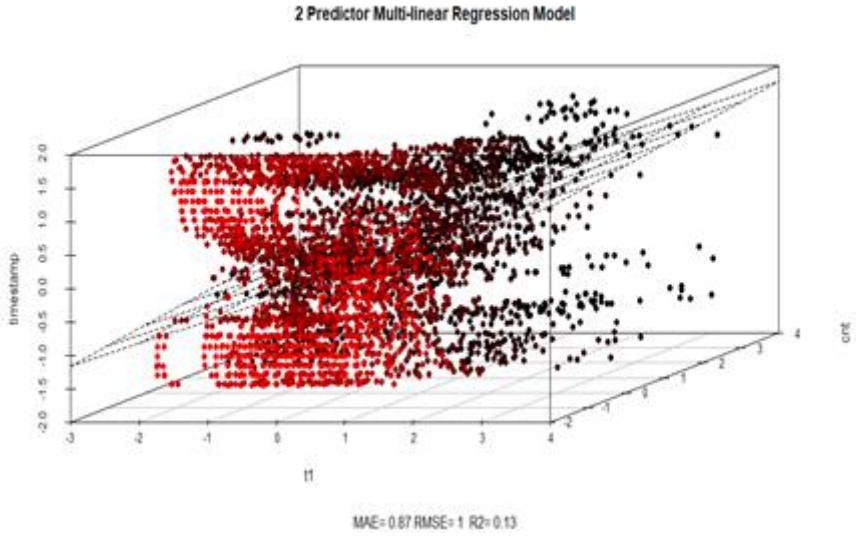


Figure 14: Multi-Linear Regression Model Predictor T1 and timestamp weekday

A multi-linear regression model modelling T1 and timestamp against count with only weekdays considered Graph shows very scattered plots implying that there is little relationship between the three values on weekdays.

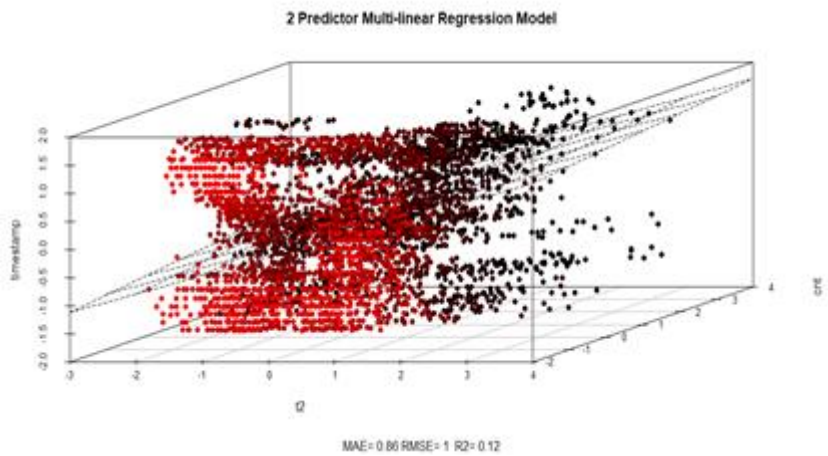


Figure 15: Multi-Linear Regression Model Predictor T2 and timestamp weekday

A multi-linear regression model modelling T2 and timestamp against count with only weekdays considered Graph shows very scattered plots implying that there is little relationship between the three values on weekdays.

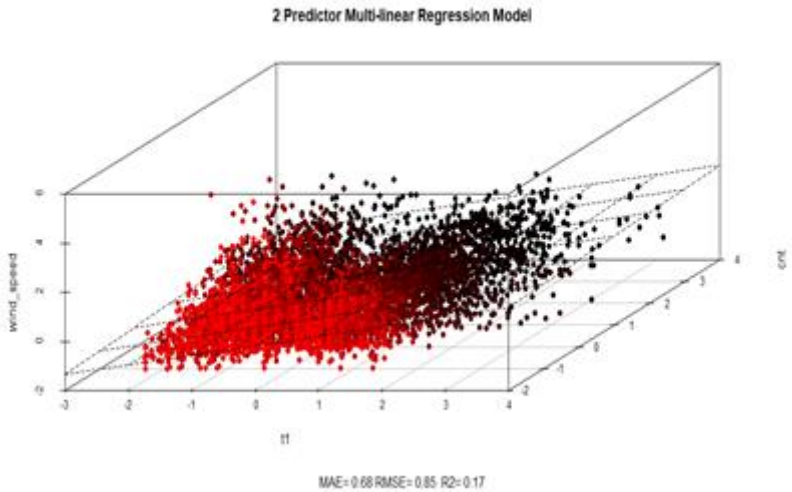


Figure 16: Multi-Linear Regression Model Predictor T1 and wind_speed both

A multi-linear regression model modelling T1 and wind_speed against count with all days considered. Graph shows very scattered plots implying that there is little relationship between the three values on any given day.

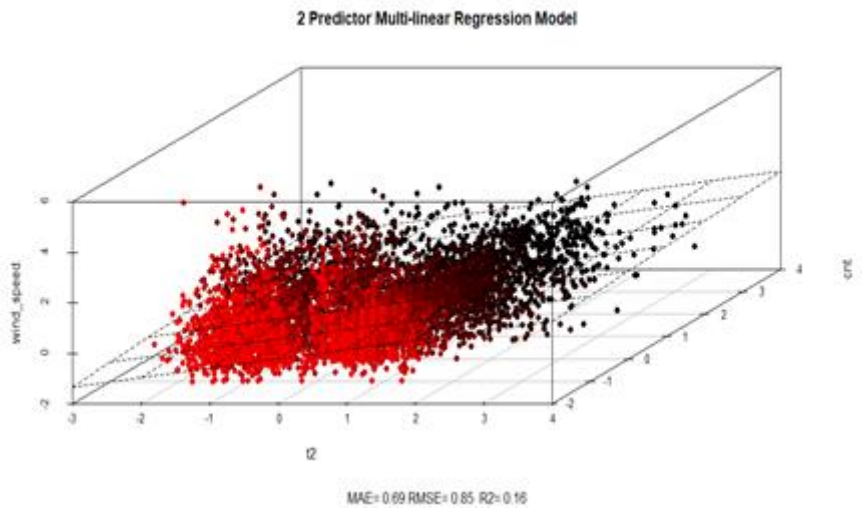


Figure 17: Multi-Linear Regression Model Predictor T1 and wind_speed both

A multi-linear regression model modelling T2 and wind_speed against count with all days considered. Graph shows very scattered plots implying that there is little relationship between the three values on any given day.

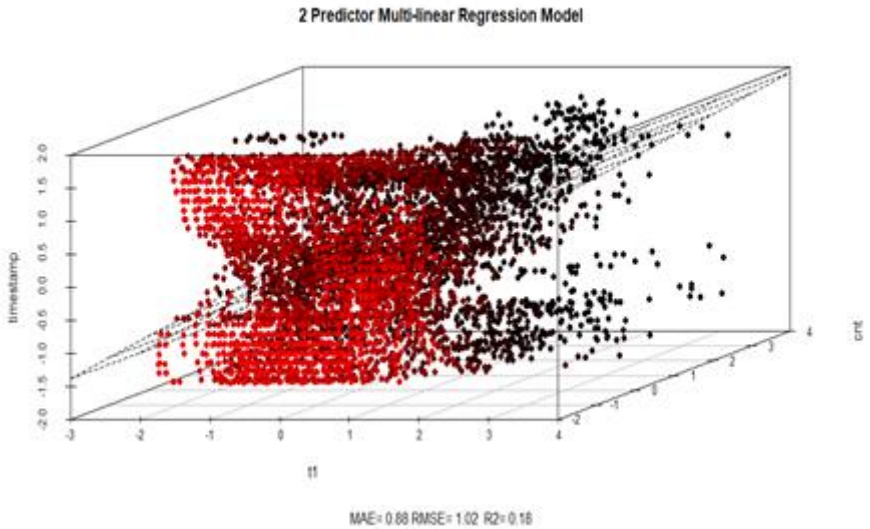


Figure 18: Multi-Linear Regression Model Predictor T1 and timestamp both

A multi-linear regression model modelling T1 and timestamp against count with all days considered. Graph shows very scattered plots implying that there is little relationship between the three values on any given day.

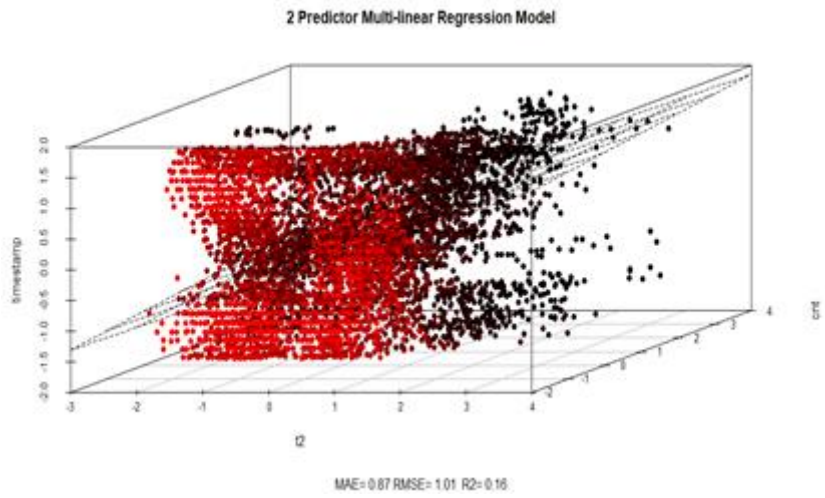


Figure 19: Multi-Linear Regression Model Predictor T2 and timestamp both

A multi-linear regression model modelling T2 and timestamp against count with all days considered. Graph shows very scattered plots implying that there is little relationship between the three values on any given day.

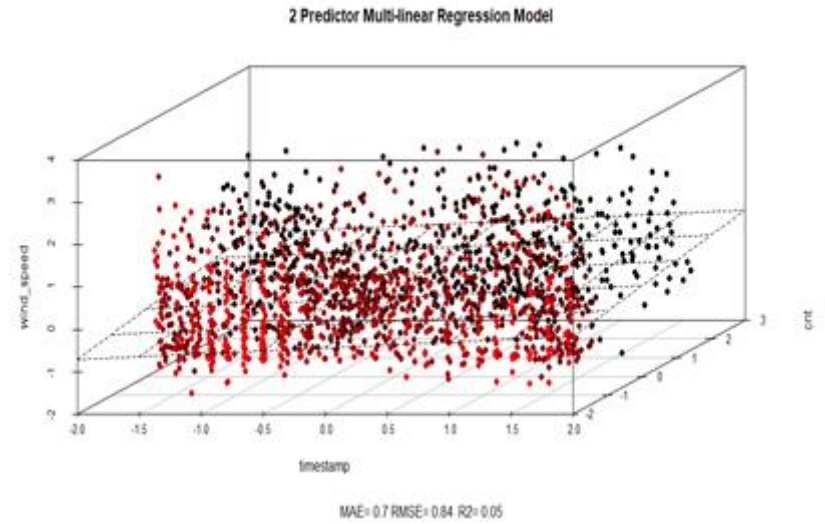


Figure 20:Multi-Linear Regression Model Predictor Windspeed and timestamp weekend

A multi-linear regression model modelling timestamp and wind_speed against count with only weekends considered. Graph shows very scattered plots implying that there is little relationship between the three values on weekends.

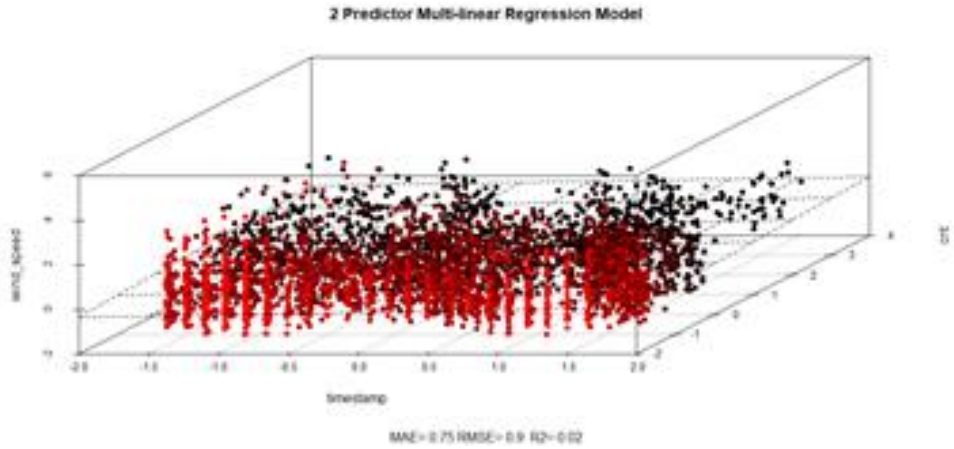


Figure 21: Multi-Linear Regression Model Predictor windspeed and timestamp weekdays only

A multi-linear regression model modelling timestamp and wind_speed against count with only weekdays considered. Graph shows very scattered plots implying that there is little relationship between the three values on weekdays.

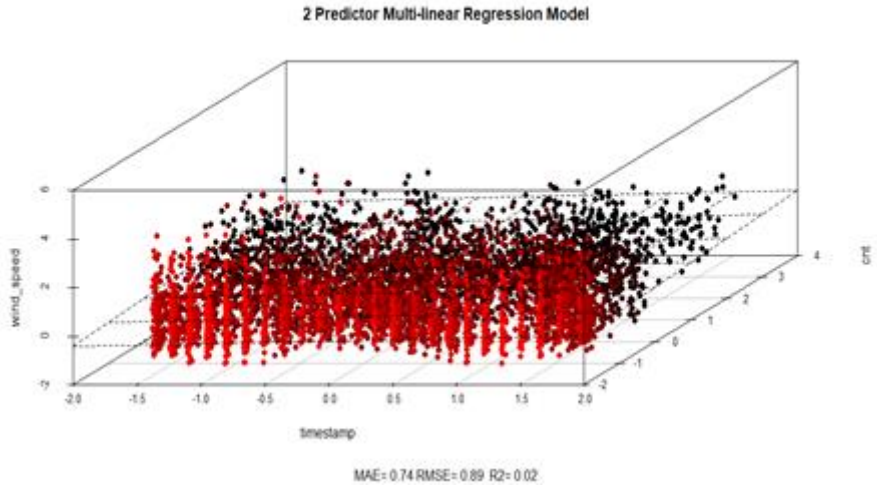


Figure 22: Multi-Linear Regression Model Predictor windspeed and timestamp all days

A multi-linear regression model modelling timestamp and wind_speed against count with all days considered. Graph shows very scattered plots implying that there is little relationship between the three values on any given day.

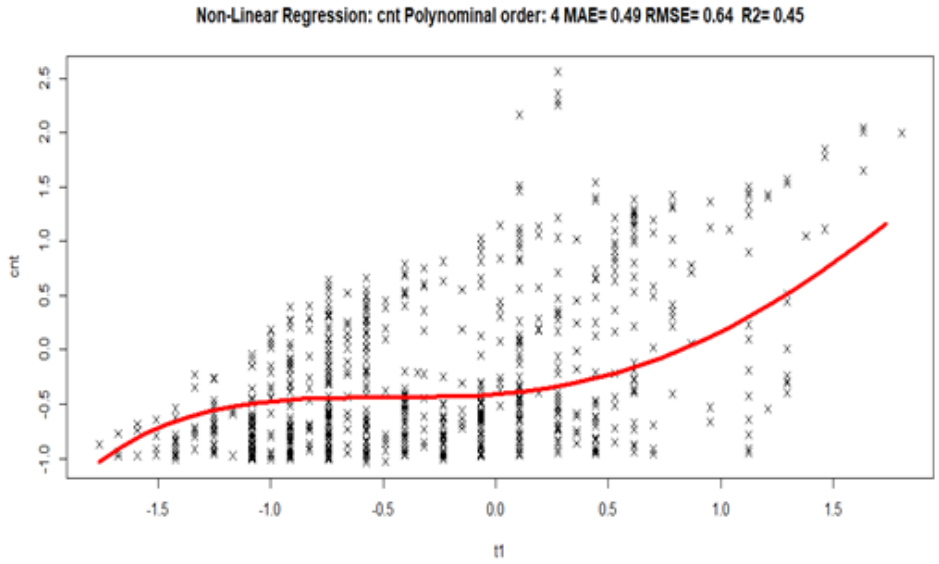


Figure 23: Non-Linear Regression Model Predictor t1 weekend only

A non-linear regression model modelling T1 against count with only weekends considered, with polynomial count equal to 4. Prediction shows a lot of values not very close to line, showing a lot of variance.

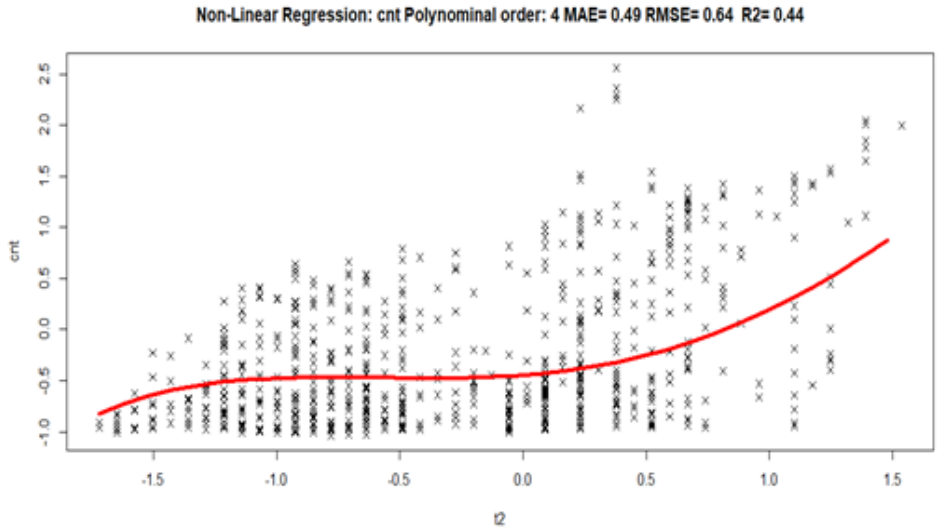


Figure 24: Non-Linear Regression Model Predictor t2 weekends only

A non-linear regression model modelling T2 against count with only weekends considered, with polynomial count equal to 4. Prediction shows a lot of values not very close to line, showing a lot of variance.

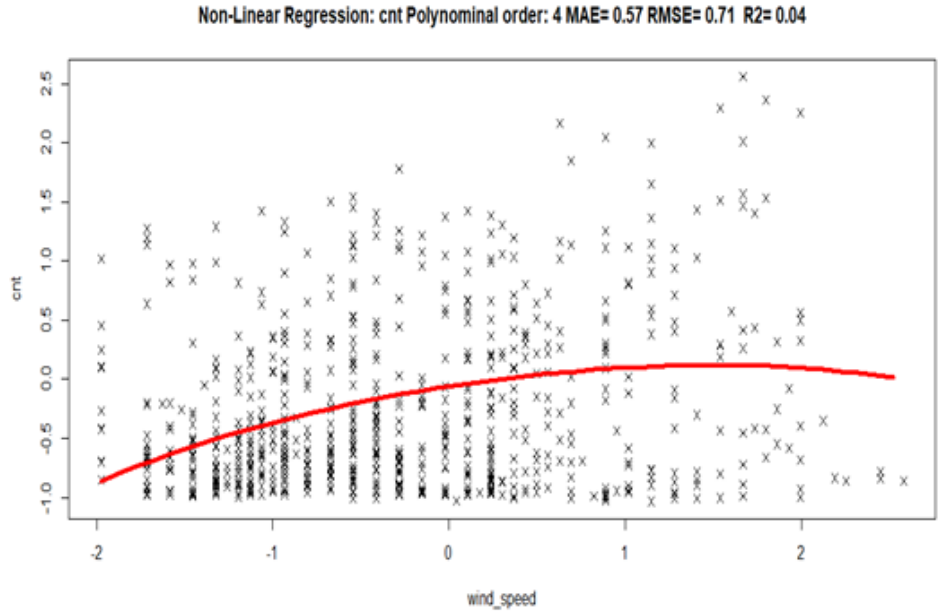


Figure 25: Non-Linear Regression Model Predictor windspeed weekends only

A non-linear regression model modelling wind_speed against count with only weekends considered, with polynomial count equal to 4. Prediction shows a lot of values not very close to line, showing a lot of variance.

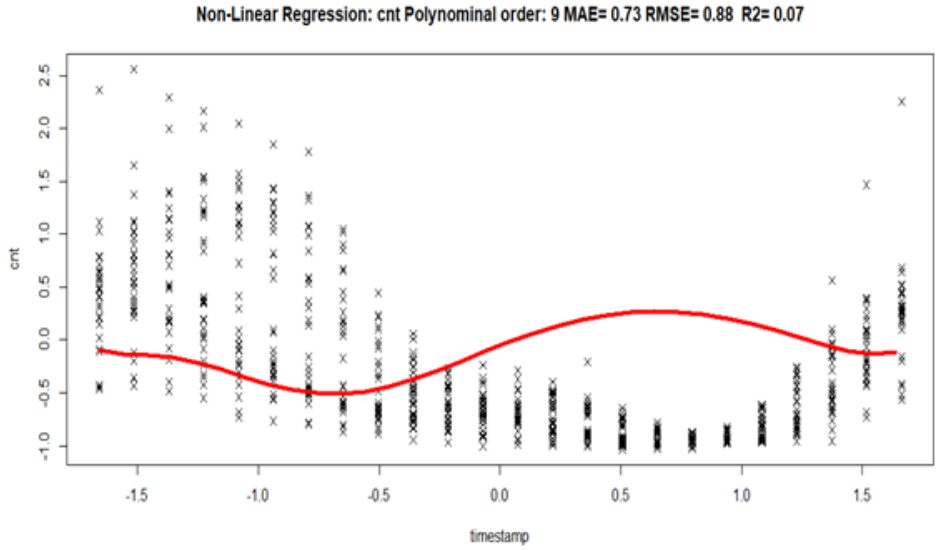


Figure 26: Non-Linear Regression Model Predictor timestamp weekends only

A non-linear regression model modelling timestamp against count with only weekends considered, with polynomial count equal to 9. Prediction shows a lot of values not very close to line, showing a lot of variance.

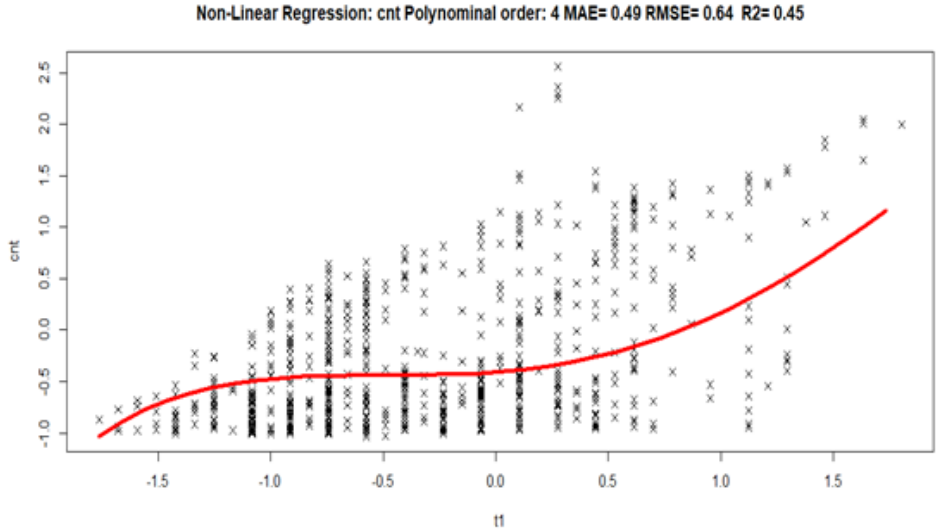


Figure 27: Non-Linear Regression Model Predictor t1 weekdays only

A non-linear regression model modelling t1 against count with only weekdays only considered, with polynomial count equal to 4. Prediction shows a lot of values not very close to line, showing a lot of variance.

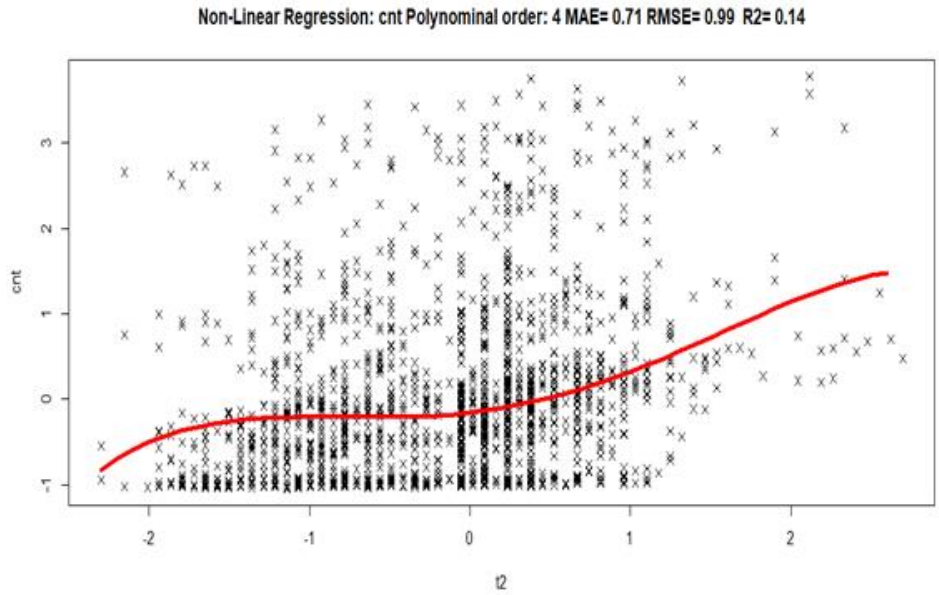


Figure 28: Non-Linear Regression Model Predictor t2 weekdays only

A non-linear regression model modelling T2 against count with only weekdays considered, with polynomial count equal to 4. Prediction shows a lot of values not very close to line, showing a lot of variance.

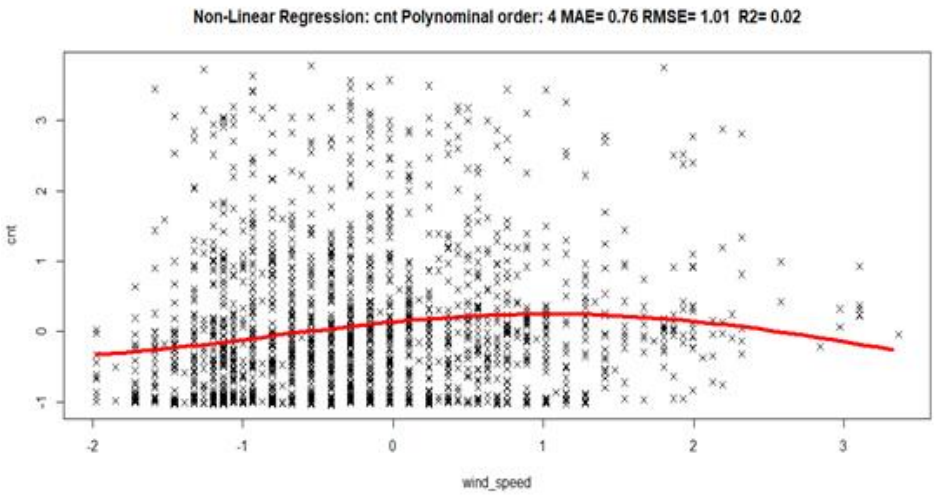


Figure 29: Multi-Linear Regression Model Predictor wind_speed weekdays only

A non-linear regression model modelling wind_speed against count with only weekdays considered, with polynomial count equal to 4. Prediction shows a lot of values not very close to line, showing a lot of variance.

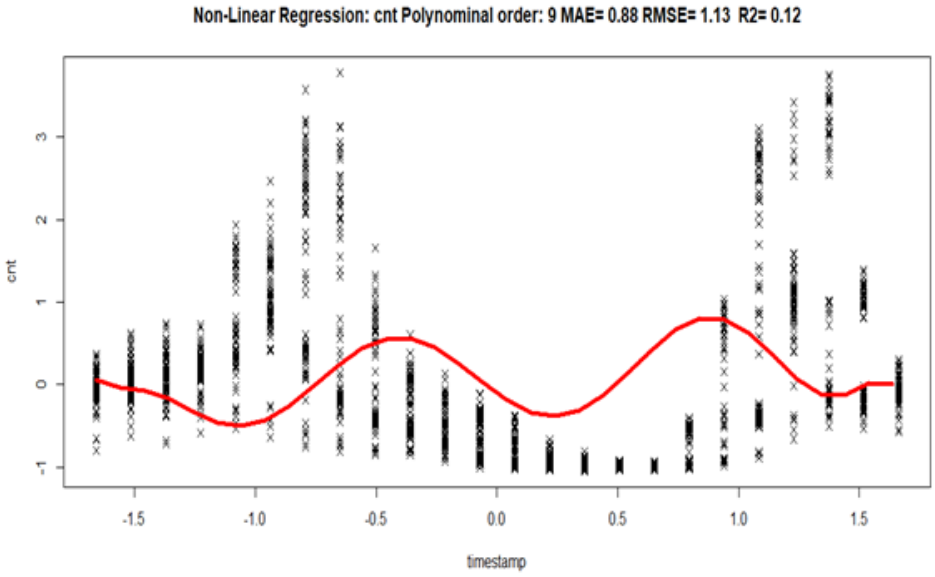
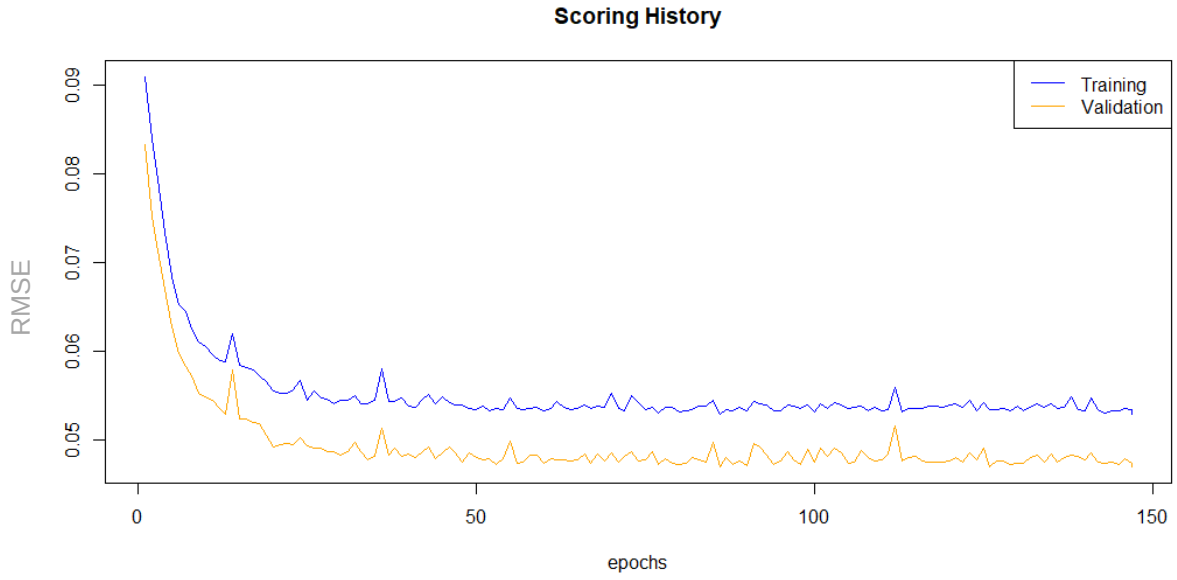


Figure 30: Non-Linear Regression Model Predictor timestamp weekdays only

A non-linear regression model modelling timestamp against count with only weekdays considered, with polynomial count equal to 9. Prediction shows a lot of values not very close to line, showing a lot of variance.

Appendix B – Deep Neural Network Models

Non-Linear Function (Tanh)



Figure

31: Training vs Validation with number of epochs

Figure 31 compares the root mean square error of the training data vs the validation data over a period of epochs. As the number of epochs increases the rmse steadily decreases with both datasets following a similar pattern.

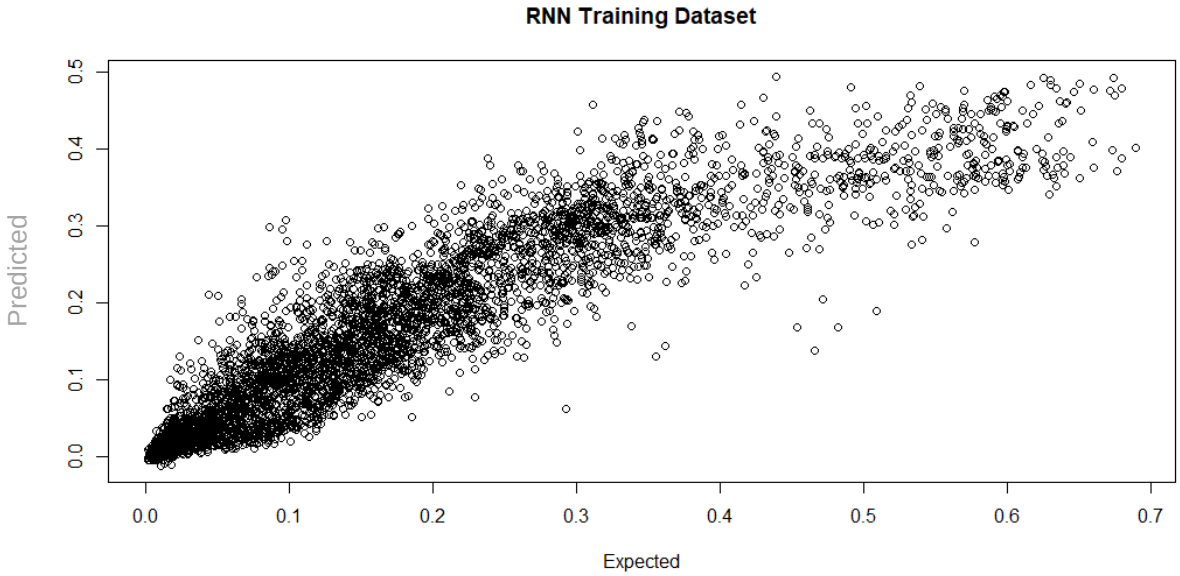


Figure 32: Expected vs Predicted

This table plots the predicted bike count value against the expected count value between 0 and 1. The graph forms a somewhat straight-line where there is match between the results. Where the area is darker this indicates more overlap between the training data and the validated data which illustrates the neural network is efficiently learning.

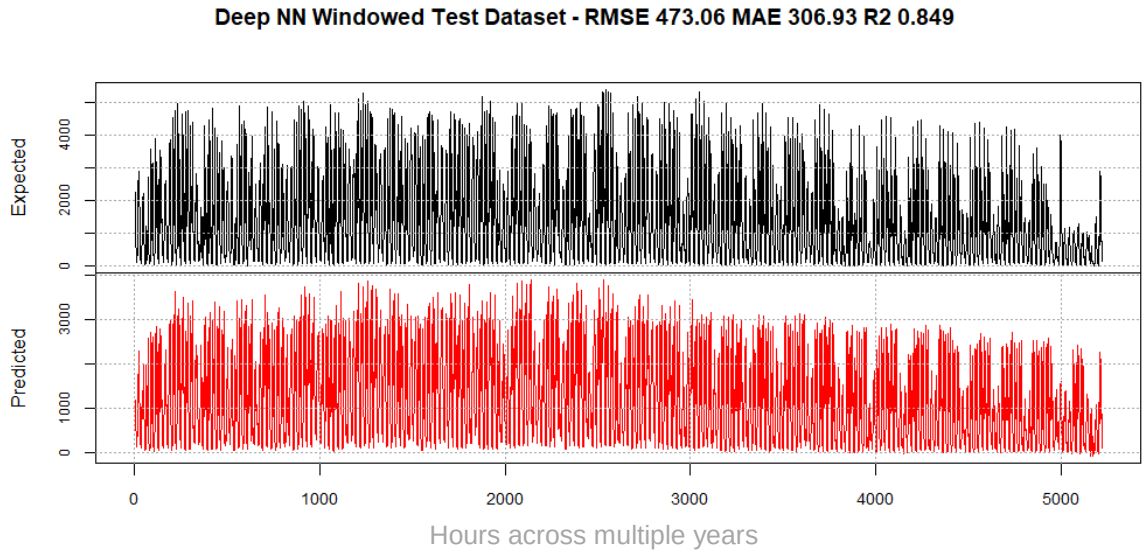


Figure 33: Comparison of results per unit time (RMSE MAE and R^2 scaled up to real values)

Figure 33 plots the predicted count against the count the we were expecting across different hours of the day. The RMSE and MAE have been scaled so that they correspond the actual number of bikes. The r^2 value fairly close to 1 which indicates a strong relationship between the input variables and the output “count”.

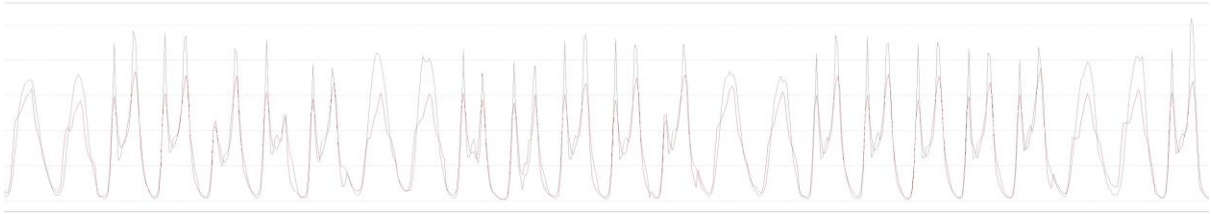


Figure 34: Snippet of overlay between predicted and expected count (Too big to title or legend. Use colour legend from previous figure.)

Figure 34 is another way of viewing figure 33. The two graphs have been overlaid and stretched so that we can clearly see the patterns, it still has issues predicting the outliers.

Linear Function (ReLU / Rectifier)

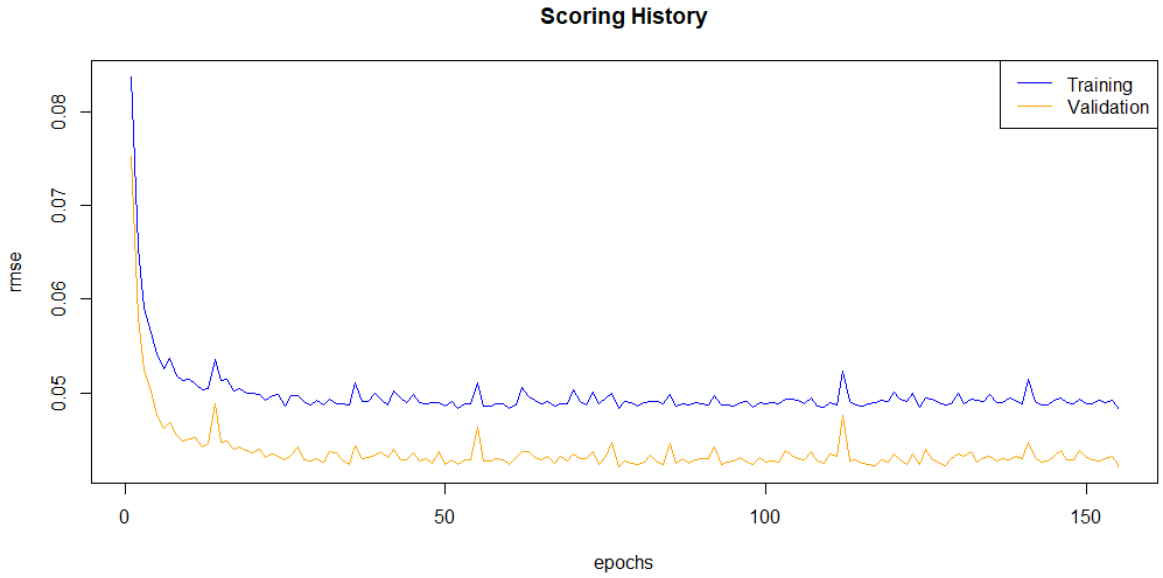


Figure 35: Training vs Validation with number of epochs

This graph displays the same data as Figure 31, however, the rmse has decreased slightly and there as less peaks across the epochs.

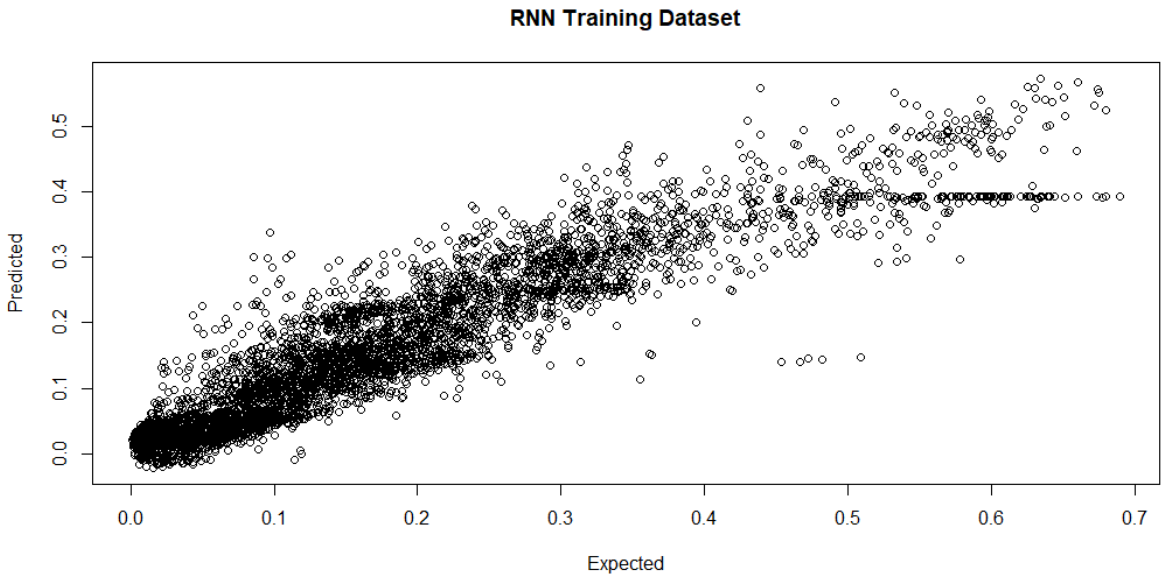


Figure 36: Expected vs Predicted

This table plots the predicted bike count value against the expected count value between 0 and 1. Note how it appears to bottom out for predicted values as our expected gets higher. This is a problem that we want to fix, clearly ReLU is having some problems (r^2 is higher but that's not a definitive measurement).

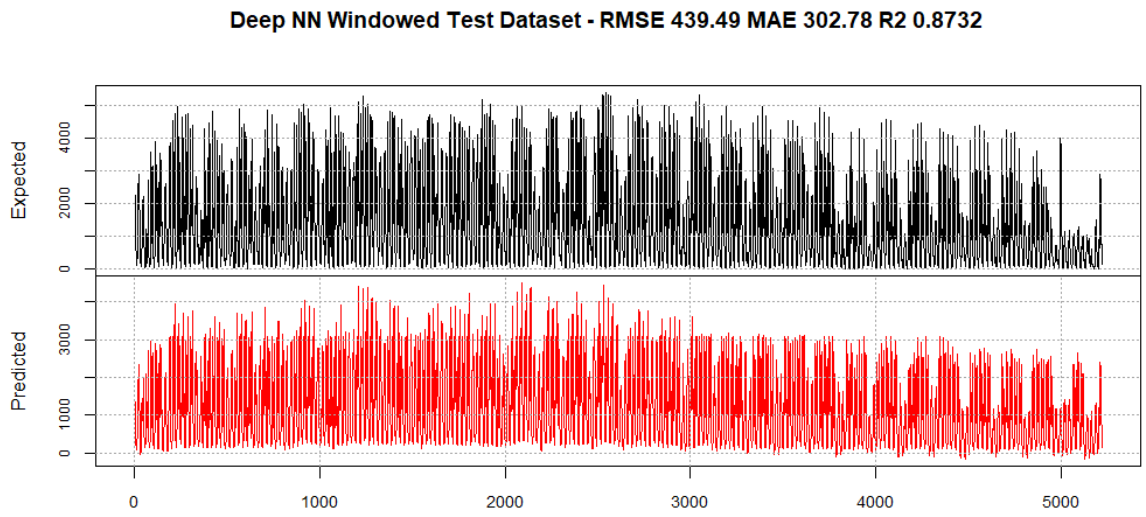


Figure 37: Comparison of results per unit time (RMSE MAE and R^2 scaled up to real values)

Figure 37 plots the predicted count against the count the we were expecting across different hours of the day. The RMSE and MAE have been scaled so that they correspond the actual number of bikes. The r^2 value close to 1 which indicates a strong relationship between the input variables the output “count”. This variant features a higher r^2 then the previous activation function but we can visually identify some issues in the previous figure.

Piecewise Linear Function (Maxout)

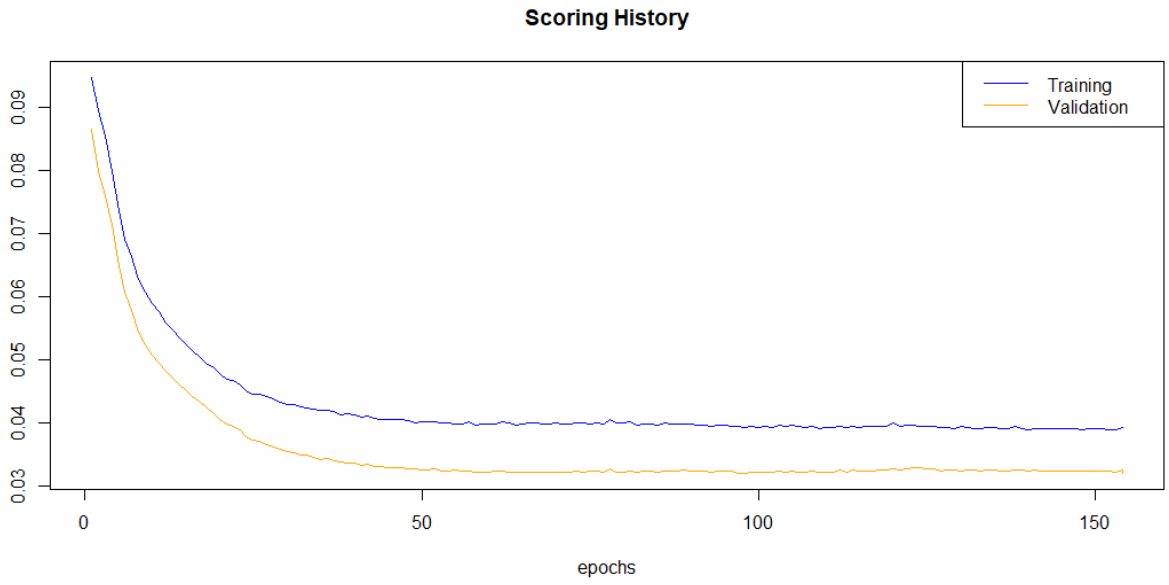


Figure 38: Training vs Validation with number of epochs

This graph displays the same data as Figure 31, however, Maxout has almost completely eliminated all peaks in the curve throughout our epochs.

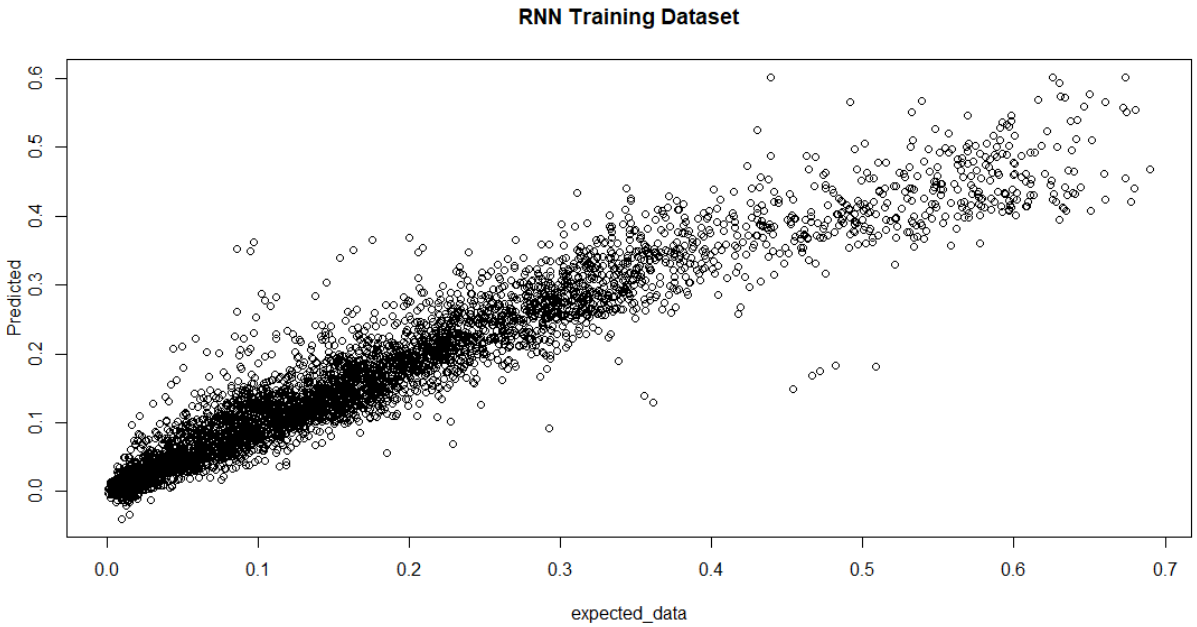


Figure 39: Expected vs Predicted

This table plots the predicted bike count value against the expected count value between 0 and 1. Maxout variant. We note that we no longer have a straight line forming at high expected values as in ReLU, and our data is bunching up nicely along the diagonal.

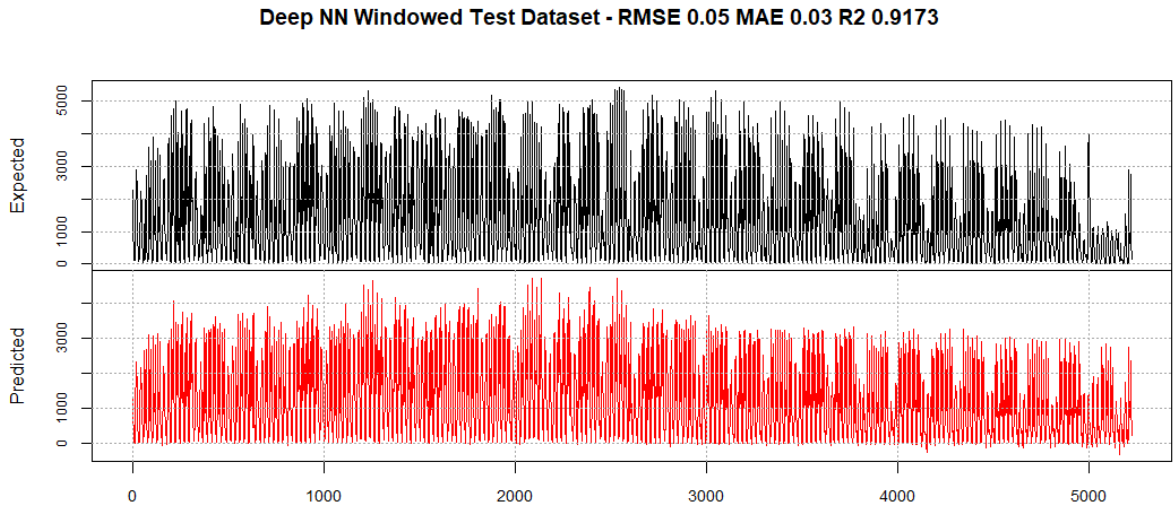


Figure 40: Comparison of results per unit time (RMSE MAE and R^2 non-scaled between 0 to 1)

Figure 40 plots the predicted count against the count the we were expecting across different hours of the day. We no longer scale the RMSE and MAE as actually drawing any conclusions from that data was proving very difficult for the purposes of a conclusion. The r^2 value is as close to 1 as our data permits without heavily overfitting to our training / testing dataset.

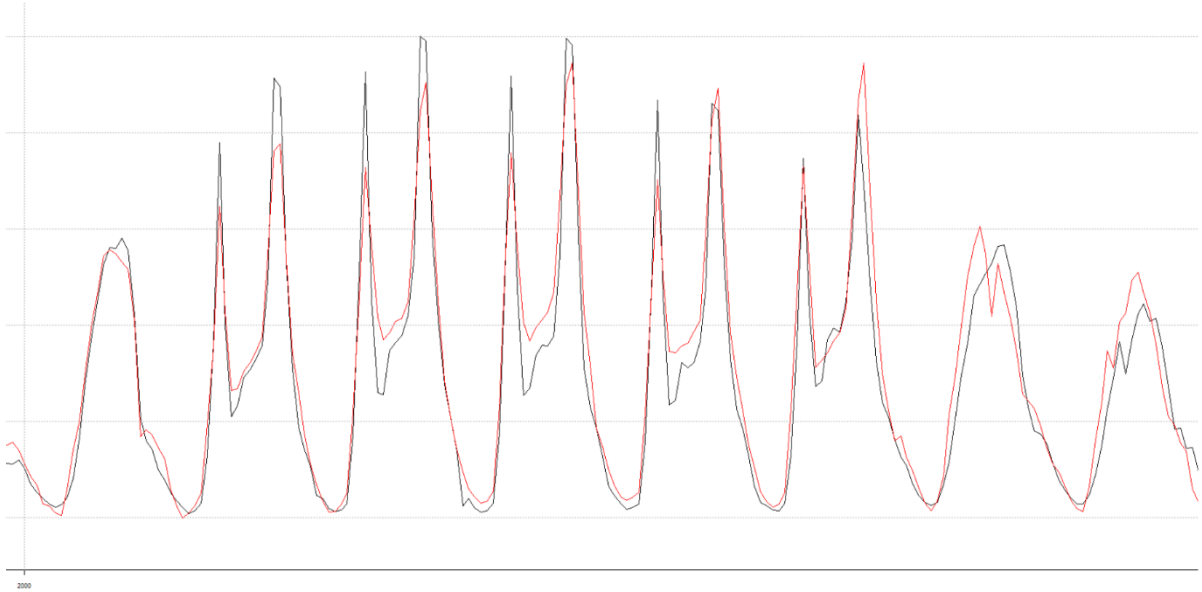


Figure 41: Residuals. Snippet of full training vs expected results. Too big to title or legend. Use colour legend from previous figure.

Figure 41 is another way of viewing figure 40. The two graphs have been overlaid and stretched so that we can clearly see the patterns, it still has issues predicting the outliers. However, we now look at a much more zoomed in a few on the span of a little over a week's worth of data, seeing the highs and lows of typical days.

Appendix C – Time Series Models

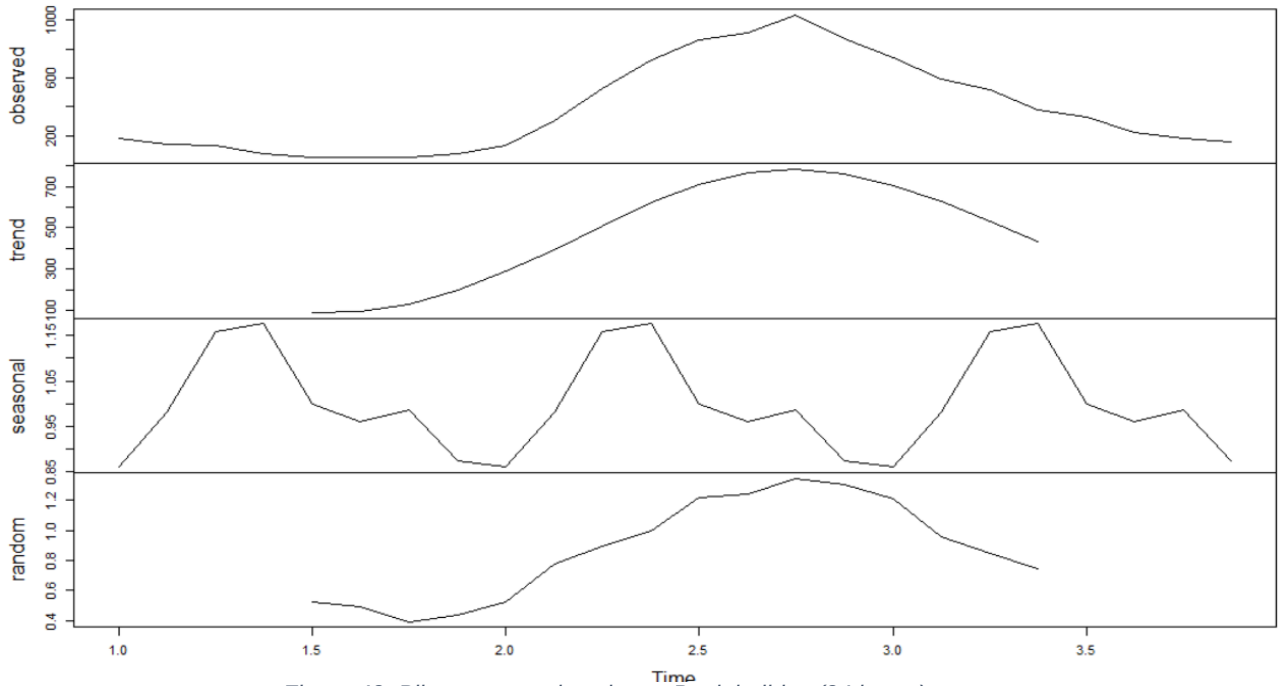


Figure 42: Bike count analysed on a Bank holiday (24 hours)

A single peak in trend here shows us the highest number of bikes being used at just past midday.

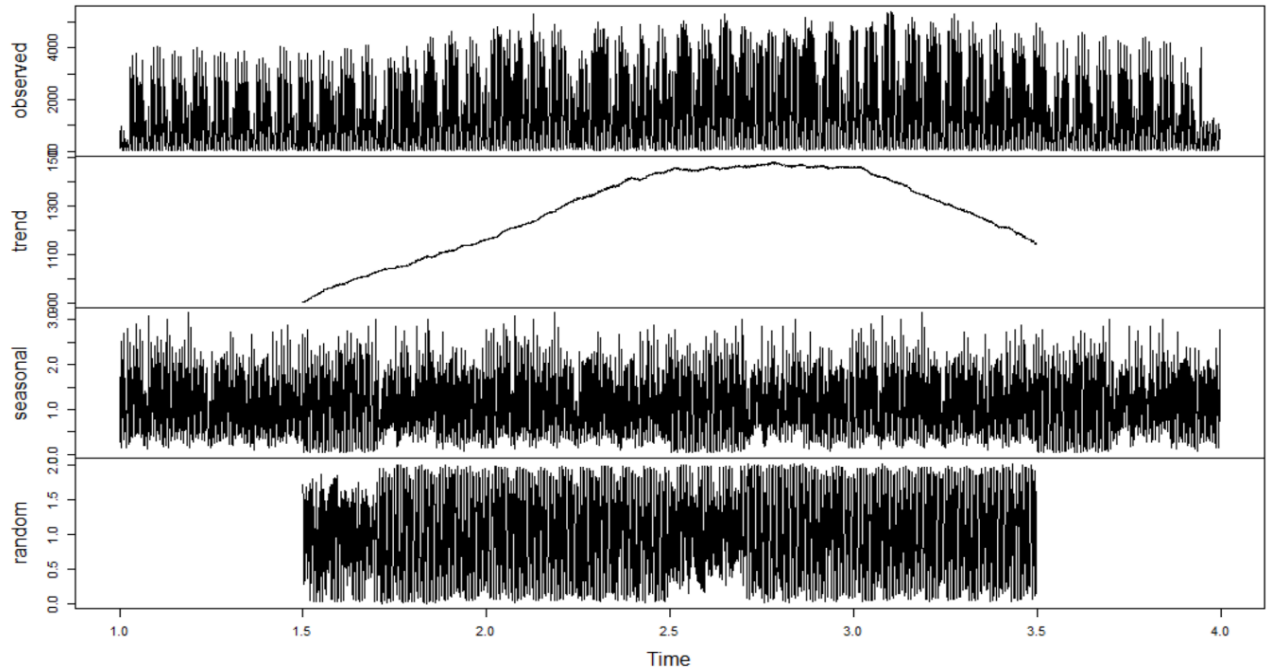


Figure 43: Bike count analysed over 2016 data (starting from January)

We can see a clear trend in count increasing during the summer period (Factors such as higher temperature, School holidays & etc) and a significant decrease during the colder period. A lot of seasonal repeats makes it hard to see the pattern here.

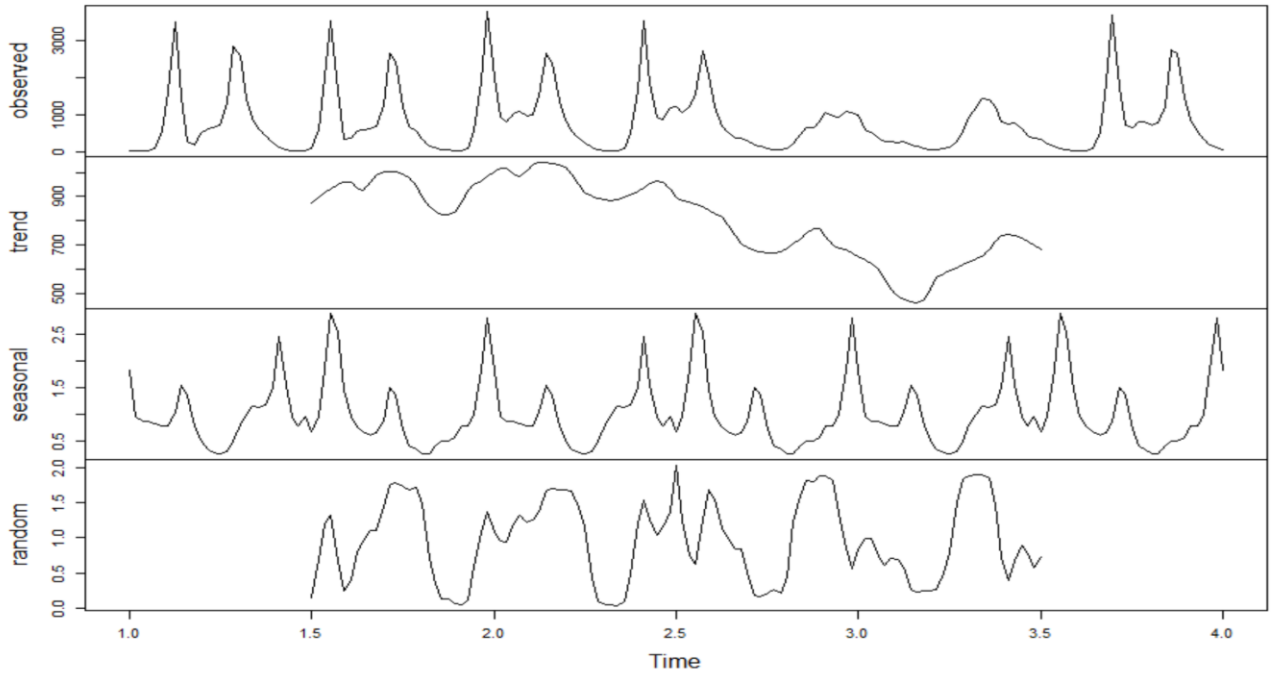


Figure 44: Trends analysed over a week in March starting from Tuesday

We can identify the weekdays from the double spikes within the observed graph which represents the rush hours of each day. Saturday and Sunday are standing out with a much less count in general as well as a much gradual (but not significant) increase in count.

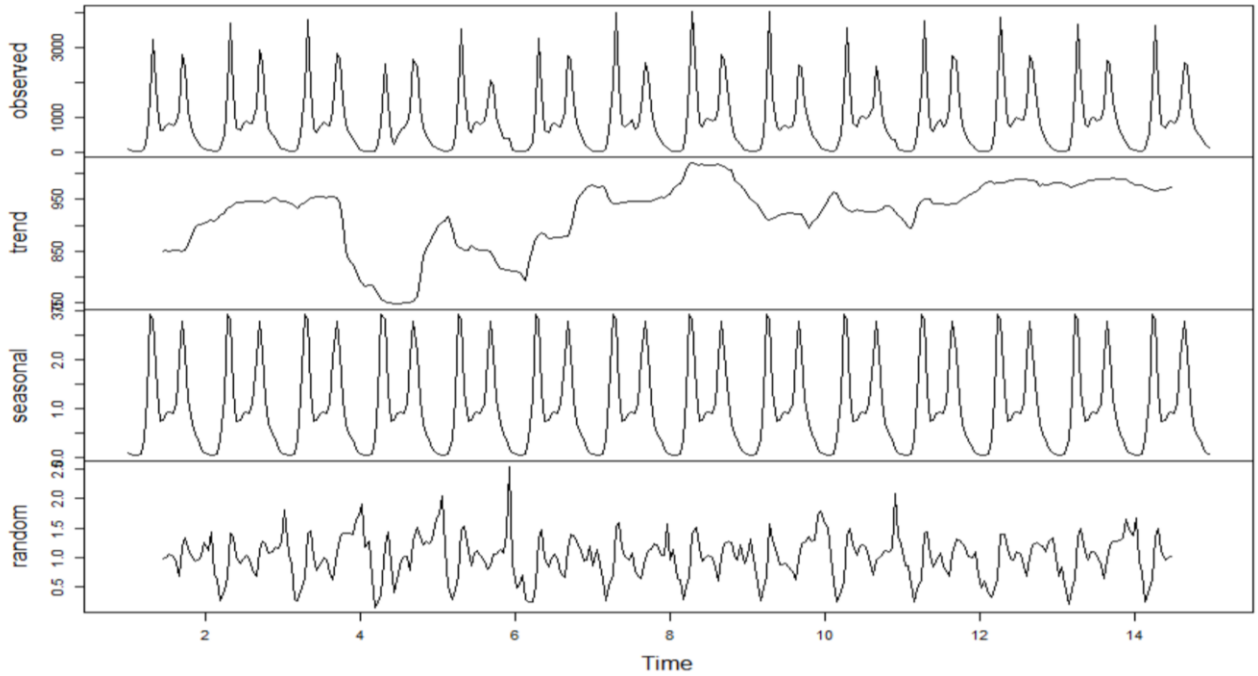


Figure 45: Bike count analysed over 14 days in March excluding Weekends & Holidays

Data shows average pattern for weekdays is that there will be two substantial peaks – these seem to coincide with rush hour times.

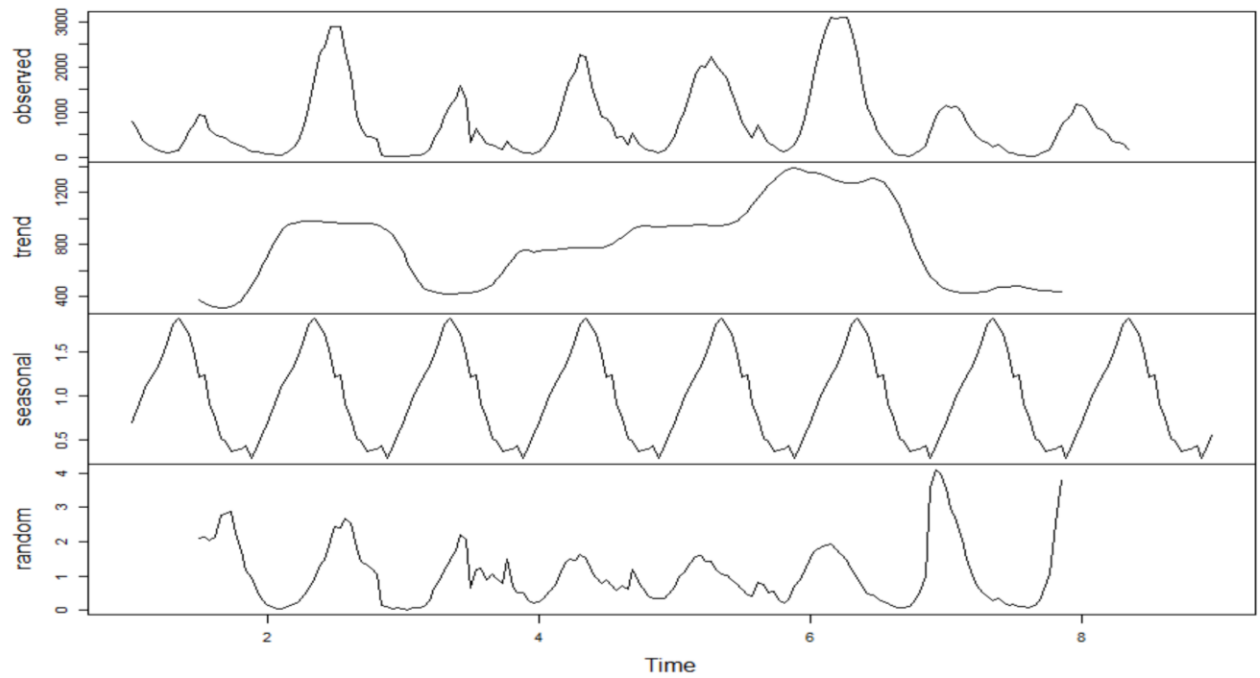


Figure 46: Bike count analysed over seven weekends in March excluding weekdays

Not much correlation as weekends vary depending on the temperature and time of the year (considering number of tourists). We can also see inconsistency in trend throughout the weekends compared to weekdays, this is due to a significant less constant number of commuters who rely on the bikes to get to work during the week.

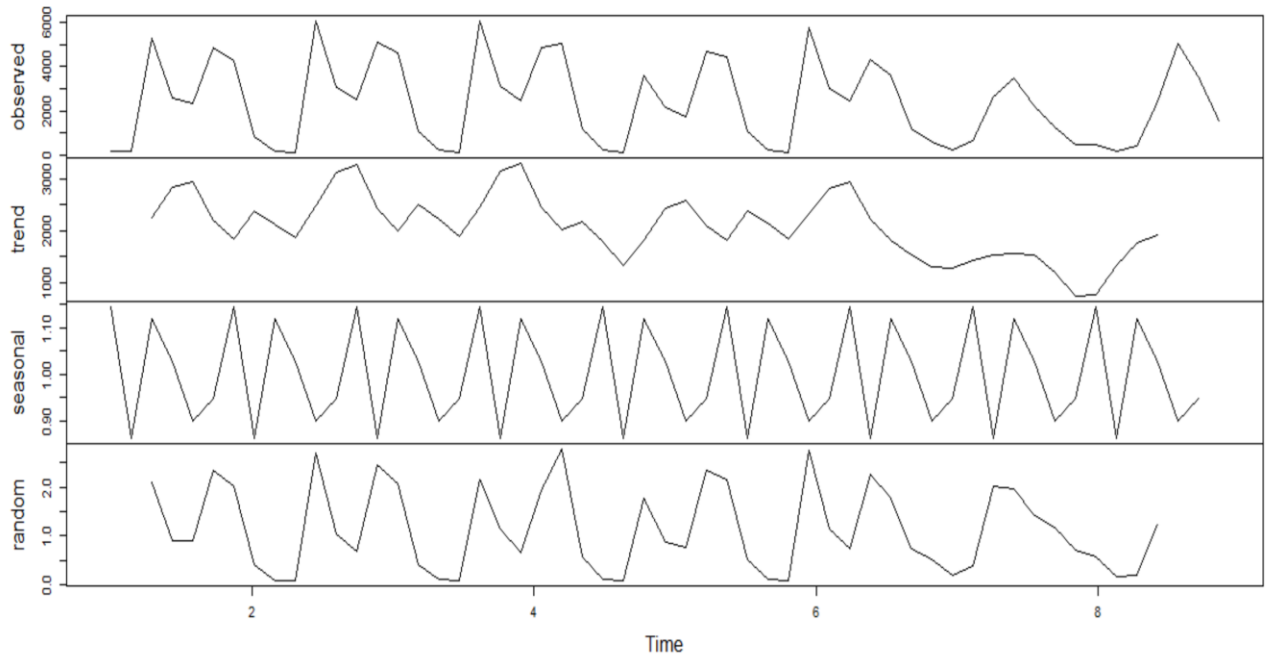


Figure 47: Bike count analysed over a week in January (With 8 bins of 3-hour periods)

We can identify a gradual decline in trend when comparing weekdays to the weekend. We can see a level of consistency throughout the weekdays (with 2 spikes occurring during rush hour).However, when it comes to Saturday and Sunday, we have unpredictable observation due to many factors affecting the trend at those times such as weather.

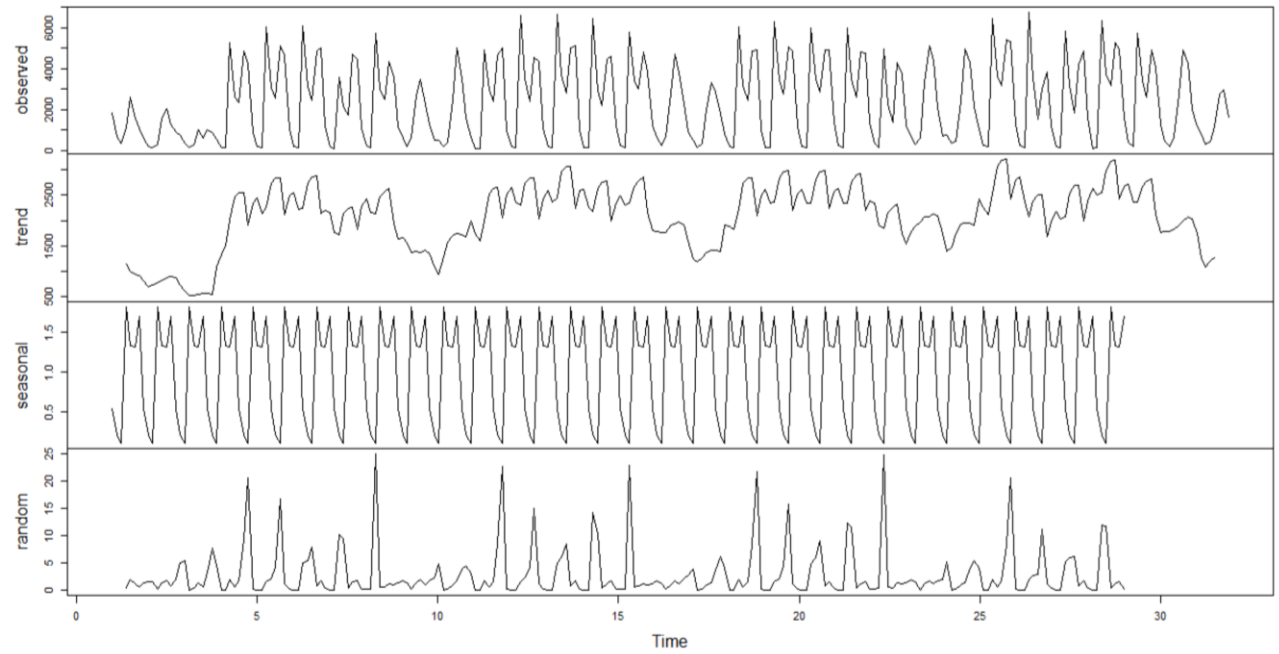


Figure 48: Bike count analysed over January including weekends (With 8 bins of 3-hour periods)

Once again, we can identify a pattern in trend where highest values correlate to weekdays and a significant decrease in usage of bikes on the weekends.

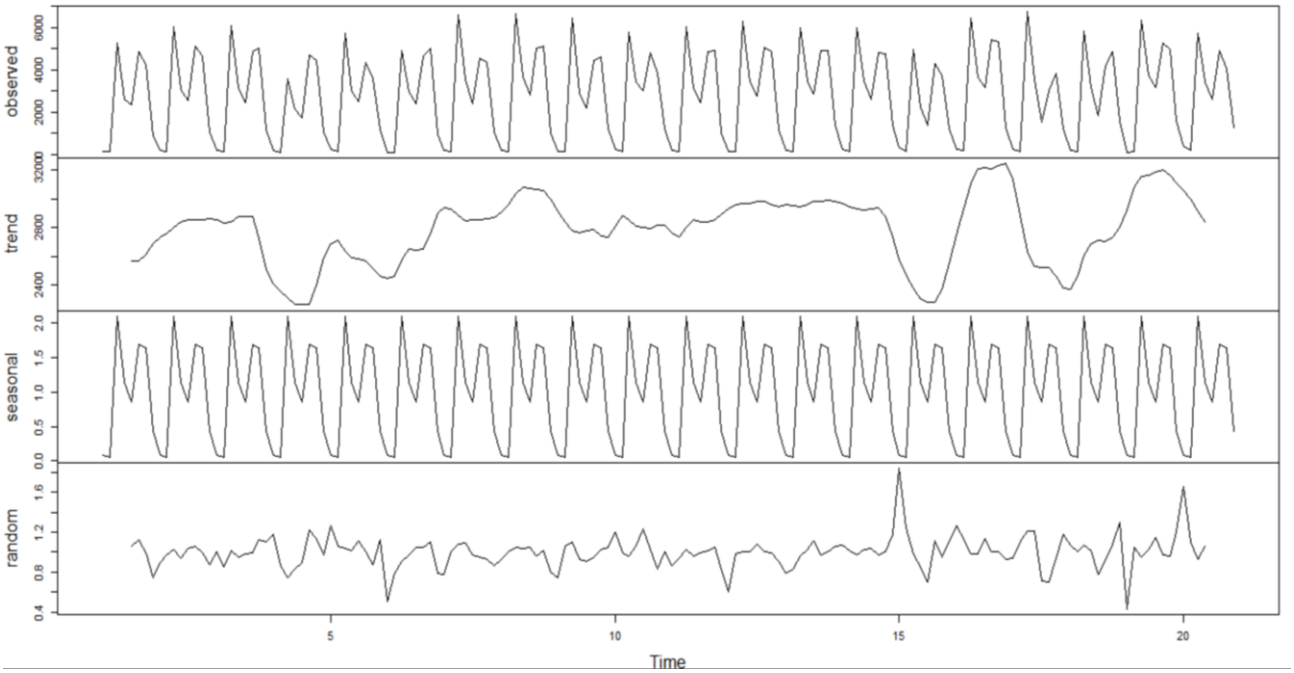


Figure 49: Bike count analysed over January excluding weekends & holidays (With 8 bins of 3-hours)

We can see the seasonal repeating patterns throughout weekdays in this month, this is mainly due to regular people using these bikes as a form of transport to commute to work and do so on a regular basis.

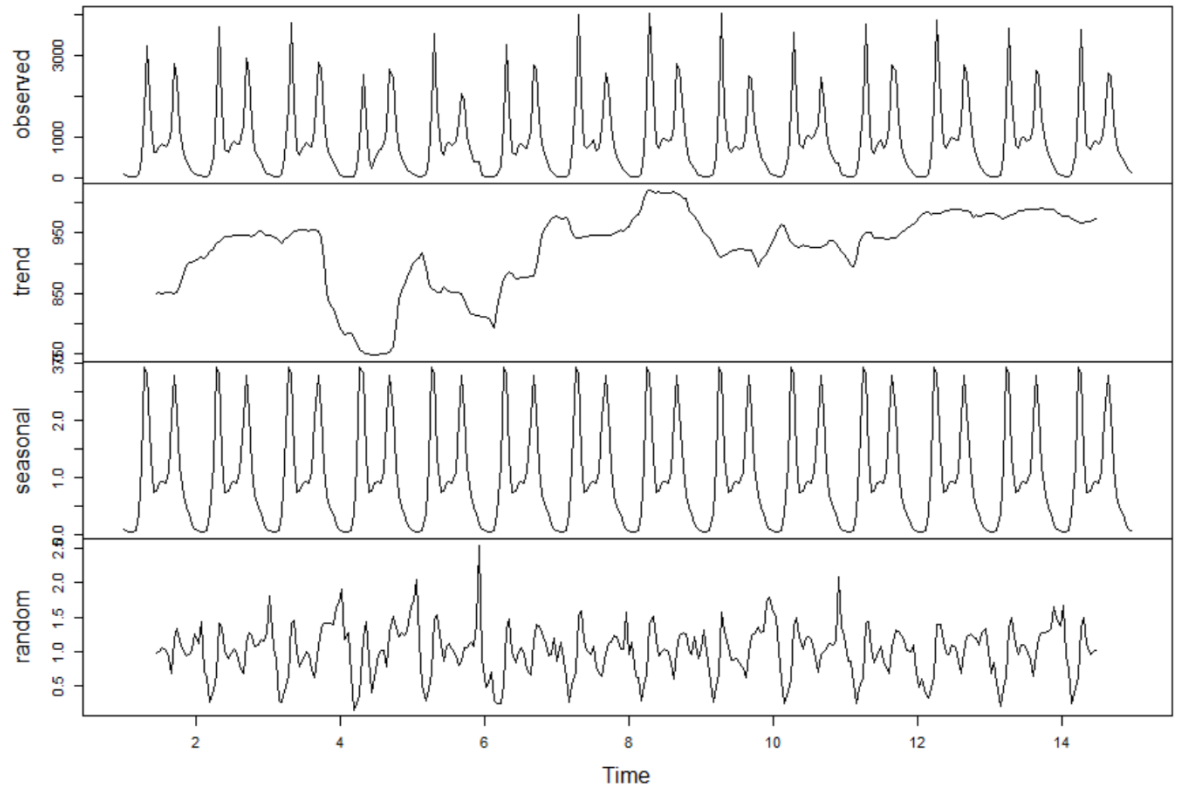


Figure 50: Bike count analysed over 14 days weekdays only

We can see the seasonal repeating patterns throughout weekdays in this period, this is mainly due to regular people using these bikes as a form of transport to commute to work and do so on a regular basis.

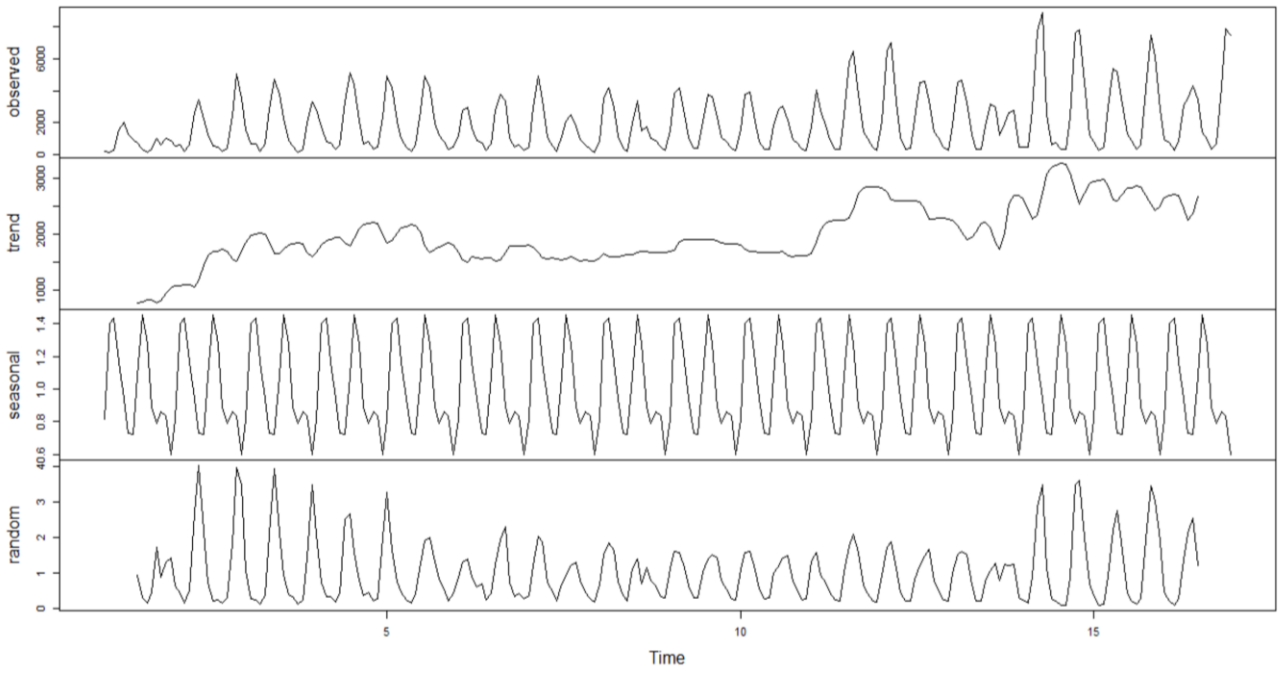


Figure 51: Bike count analysed across 16 weekends starting from January (With 8 bins of 3-hours)

We can see the 2 spikes in seasonal trends representing Saturdays & Sundays across the 16-week period. Towards the end of march as temperature increases, we can see a trend in bike counts going up, however it is possible that other factors cause the rise in this trend.

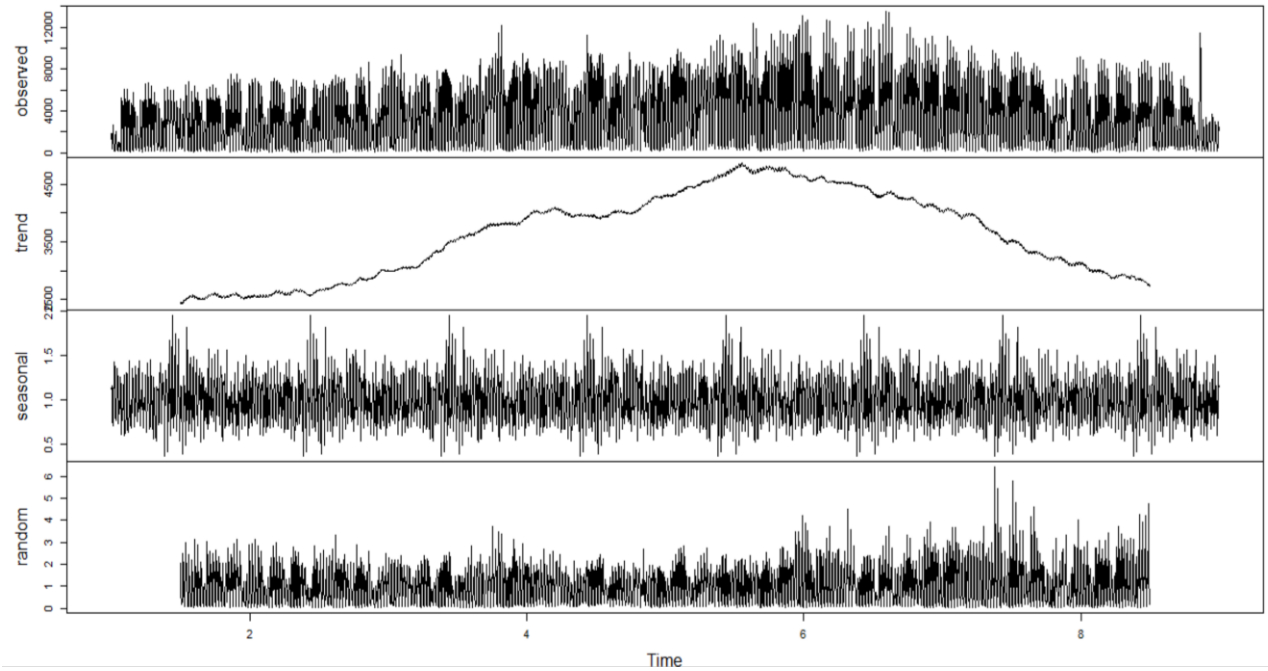


Figure 52: Bike count analysed over 2016 data starting from January (8 bins of 3-hours)

We can see a clear trend in count increasing during the summer period (Factors such as higher temperature, School holidays & etc) and a significant decrease during the colder period. In comparison to Figure 39 we can see that count has increased (x3) and a more precise trend in terms of visualization due to using Bins.

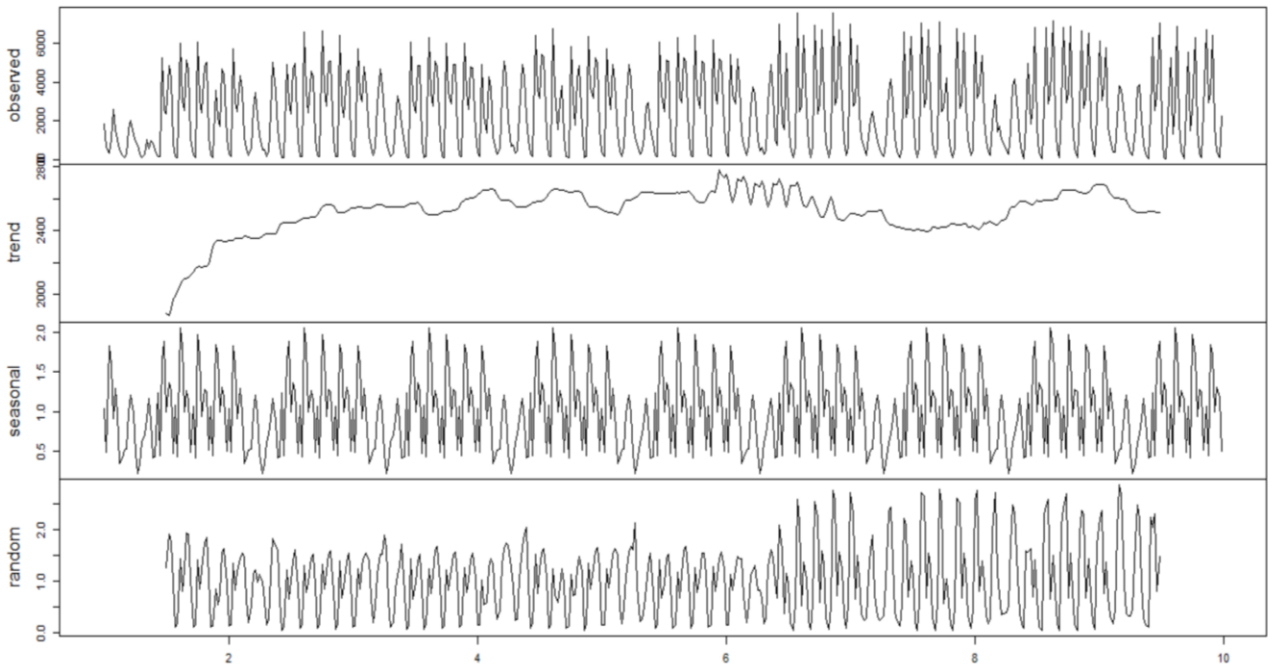


Figure 53: Bike count analysed over 10 weeks from January including weekends (8 bins of 3-hours)

We can see during a colder period of the year the trend is consistently low, however, due to constant number of commuters we still have a fluctuation when weekdays and weekend is compared.

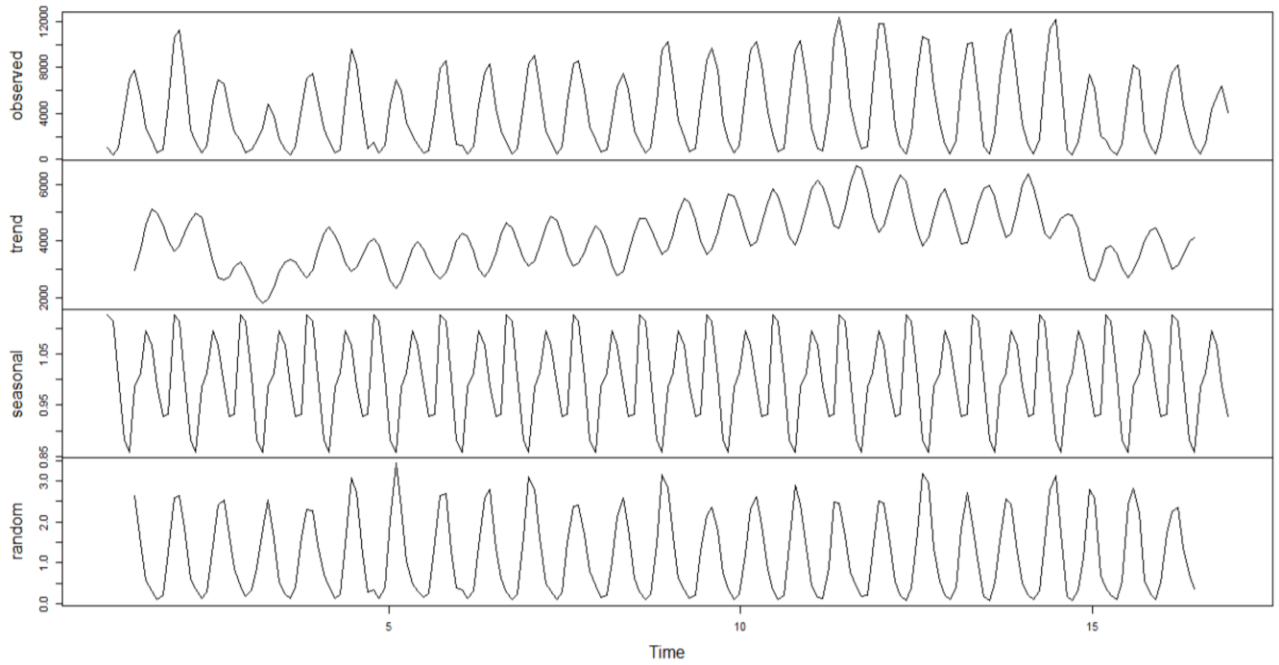


Figure 54: Bike count analysed over summer 2016 excluding weekdays

Although we do not consider weekdays (where we get the most consistent results due to commuters) during the summer period on weekends we have a combination of warm weather, school holidays, tourists & people off from work which are all reasons for increasing number of bikes being used in London. However due to many factors affecting count at once we cannot identify in detail which reason affects bike count from just looking at this trend.

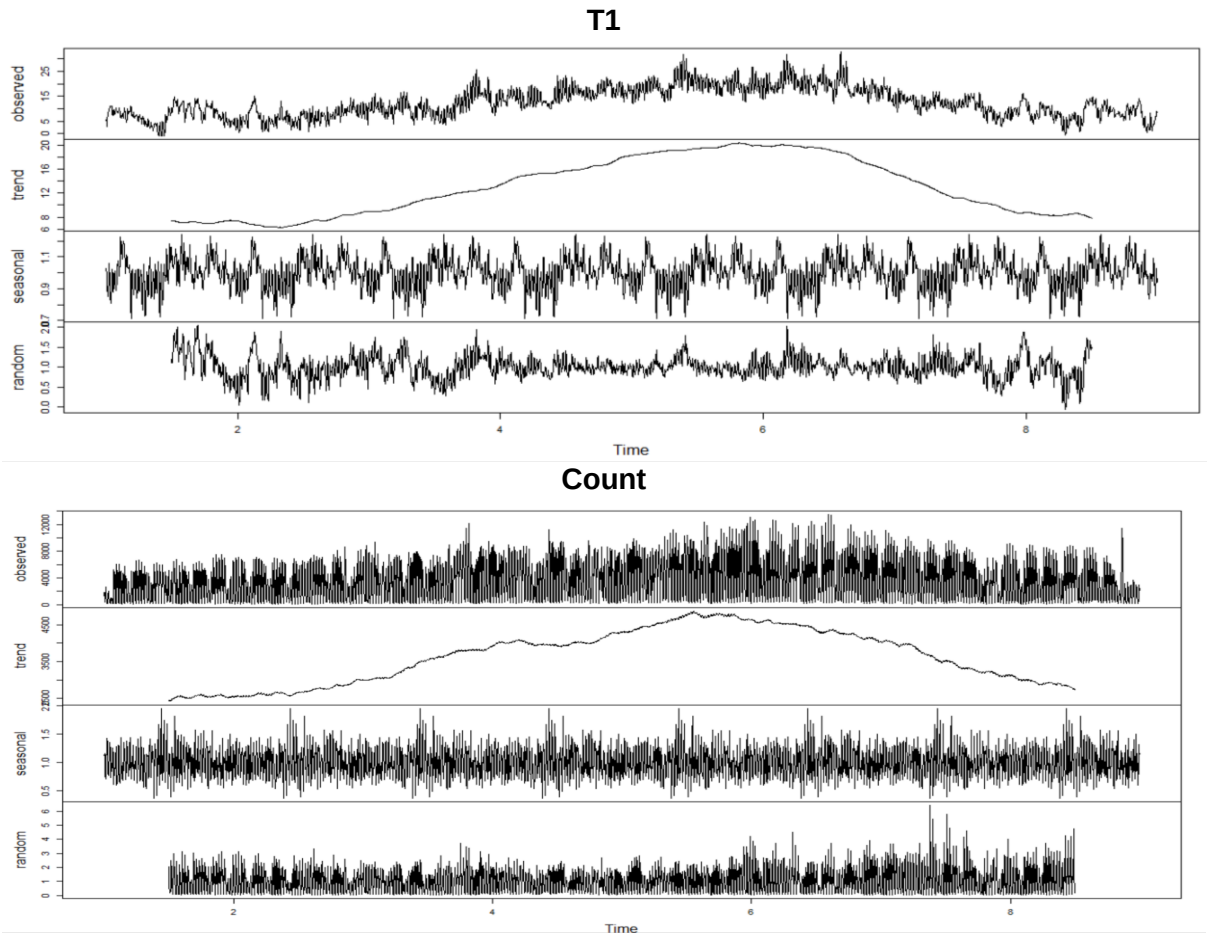
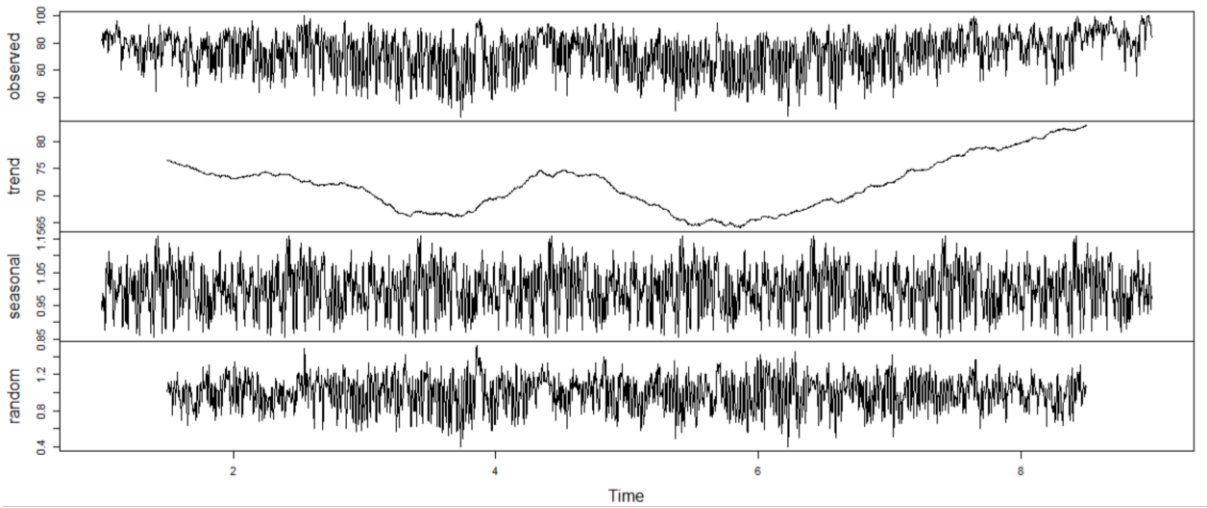


Figure 55: T1 and bike count analysed over 2016 including weekdays

This shows the overall trend of t1 as well as bike count over the course of the year. What is interesting to note is how you can see how the trend for bike count shows a gradual increase that matches the trend shown for temperature well, showing there is also a correlation between temperature and bike count.

Humidity



Count

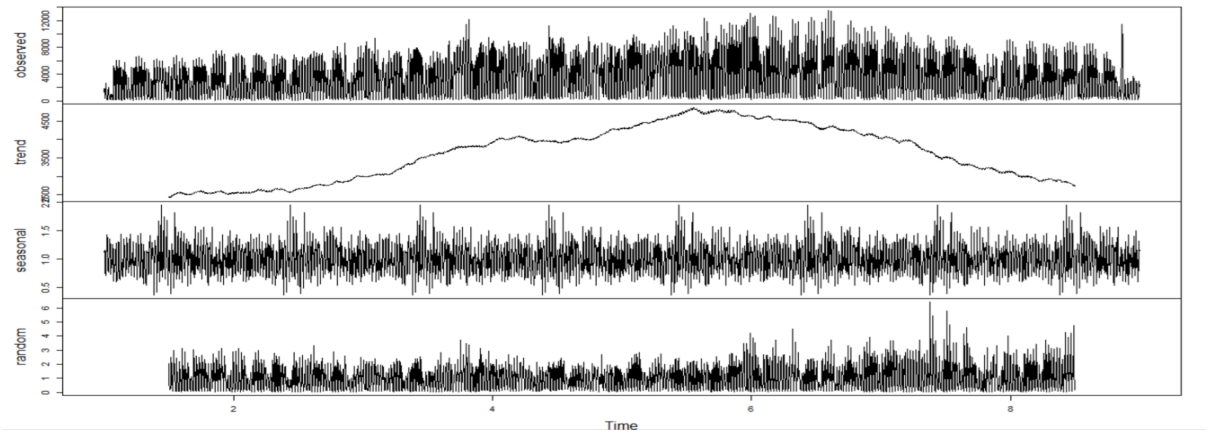


Figure 56: Humidity analysed over 2016 including weekends

This shows the overall trend of humidity as well as bike count over the course of the year. There seems to be little similarities in trend that we can gather from these graphs, other than that the lowest point of humidity coincided with the highest point in bike count

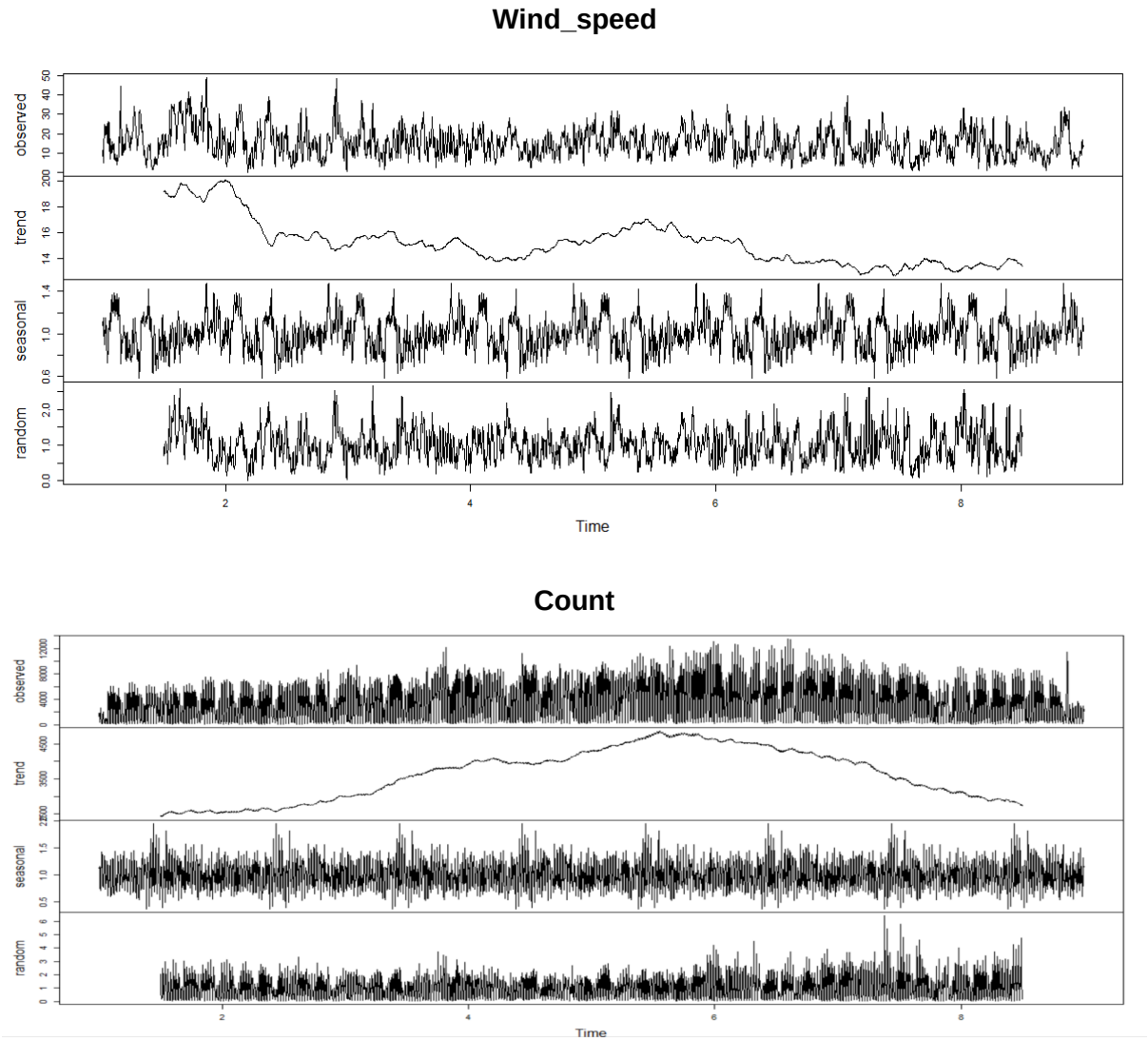


Figure 57: Wind_Speed analysed over 2016 including weekends

This shows the overall trend of wind_speed as well as bike count over the course of the year. Trends do not seem to match in any way, other than wind_speed is highest at the beginning of the year when bike count is lowest.

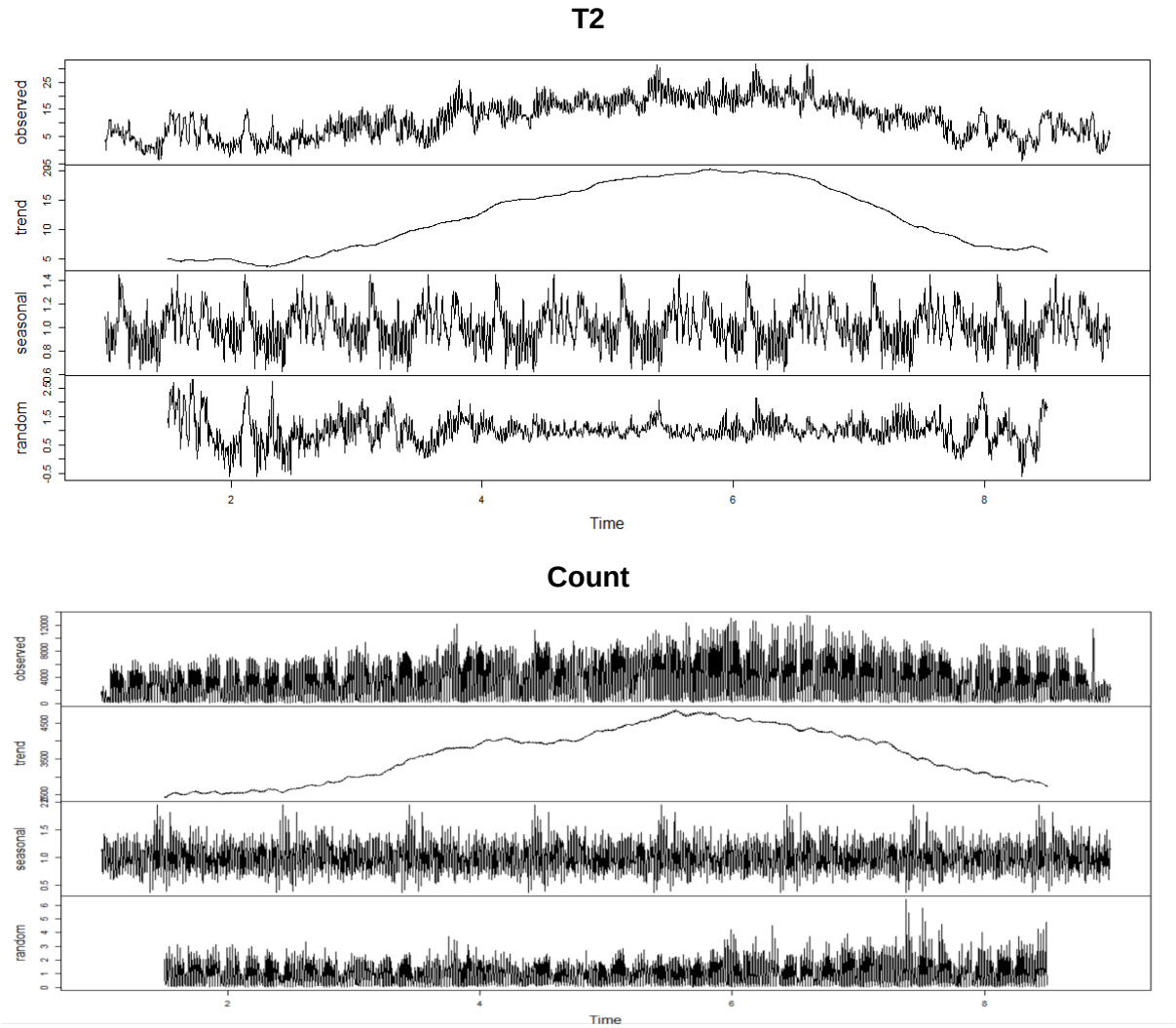


Figure 58: t2 analysed over 2016 including weekends

This shows the overall trend of t2 as well as bike count over the course of the year. What is interesting to note is how you can see how the trend for bike count shows a gradual increase that matches the trend shown for temperature well, showing there is also a correlation between temperature and bike count.

Appendix D – Model Accuracy

Throughout this report, it will continually compare the results against RMSE,MAE and R^2. It is important to understand what these values represent and why they are appropriate to the coursework (Datatechnotes.com, 2019).

- MAE - Mean Absolute Error: Is the absolute difference between the expected values and the values predicted by the classifier divided by the number of values in the dataset.
- MSE - Mean Squared Error: Is the difference between the expected values and the values predicted by the classifier squared divided by the number of values in the dataset.
- RSME - Root Mean Squared Error: Is the square root of the MSE previously calculated
- R^2: Is a coefficient which represents how well your predicted values fit the expected values between 0 and 1, with the higher "generally" being the better.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}|$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2$$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2}$$

$$R^2 = 1 - \frac{\sum(y_i - \hat{y})^2}{\sum(y_i - \bar{y})^2}$$

Where,

$\hat{y}$ – *predicted value of y*

$\bar{y}$ – *mean value of y*

The Main issue with R-Squared is that it does not determine whether the results produced are biased or overfitted, to overcome this you must assess the residual plots. The R squared value does not inherently have to be close to 1f. The Main issue with R-Squared is that it does not determine whether the results produced are biased or overfitted, to overcome this you must assess the residual plots. The R squared value does not inherently have to be close to 1 for it to be viable. That is if your predictors are still statistically significant (Report on r^2) you can still draw important conclusions. It is also important to note that a High R^2 value does not indicate that your model is a good fit.

Although R^2 is good for estimated the strength of the relationship between your variables and the model it does not provide a test for your hypothesis. or it to be viable. That is if your predictors are still statistically significant you can still draw important conclusions. It is also important to note that a High R^2 value does not indicate that your model is a good fit.
Although R^2 is good for estimated the strength of the relationship between your variables and the model it does not provide a test for your hypothesis (Regression Analysis: How Do I Interpret R-squared and Assess the Goodness-of-Fit?, 2013).

References

Datatechnotes.com. (2018). *Z-score calculation with R*. [online] Available at: https://www.datatechnotes.com/2018/02/z-score-with-r.html [Accessed 23 Nov. 2019].

Datatechnotes.com. (2019). *Regression Model Accuracy (MAE, MSE, RMSE, R-squared) Check in R*. [online] Available at: https://www.datatechnotes.com/2019/02/regression-model-accuracy-mae-mse-rmse.html [Accessed 28 Nov. 2019].

En.wikipedia.org. (2019). *Feedforward neural network*. [online] Available at: https://en.wikipedia.org/wiki/Feedforward_neural_network [Accessed 26 Nov. 2019].

Regression Analysis: How Do I Interpret R-squared and Assess the Goodness-of-Fit?. (2013). [Blog] *The Minitab Blog*. Available at: https://blog.minitab.com/blog/adventures-in-statistics-2/regression-analysis-how-do-i-interpret-r-squared-and-assess-the-goodness-of-fit [Accessed 1 Dec. 2019].

Licences

Dataset License

These licence terms and conditions apply to TfL's free transport data service and are based on version 2.0 of the Open Government Licence with specific amendments for Transport for London (the "Licence"). TfL may at any time revise this Licence without notice. It is up to you ("You") to regularly review the Licence, which will be available on this website, in case there are any changes. Your continued use of the transport data feeds You have opted to receive ("Information") after a change has been made to the Licence will be treated as Your acceptance of that change.

Using Information under this Licence TfL grants You a worldwide, royalty-free, perpetual, non-exclusive Licence to use the Information subject to the conditions below (as varied from time to time).

This Licence does not affect Your freedom under fair dealing or fair use or any other copyright or database right exceptions and limitations.

This Licence shall apply from the date of registration and shall continue for the period the Information is provided to You or You breach the Licence.

Rights You are free to:

Copy, publish, distribute and transmit the Information Adapt the Information and Exploit the Information commercially and non-commercially for example, by combining it with other Information, or by including it in Your own product or application Requirements You must, where You do any of the above:

Acknowledge TfL as the source of the Information by including the following attribution statement 'Powered by TfL Open Data' Acknowledge that this Information contains Ordnance Survey derived data by including the following attribution statement: 'Contains OS data © Crown copyright and database rights 2016' and Geomni UK Map data © and database rights [2019] Ensure our intellectual property rights, including all logos, design rights, patents and trademarks, are protected by following our design and branding guidelines Limit traffic requests up to a maximum of 300 calls per minute per data feed. TfL reserves the right to throttle or limit access to feeds when it is believed the overall service is being degraded by excessive use and Ensure the information You provide on registration is accurate These are important conditions of this Licence and if You fail to comply with them the rights granted to You under this Licence, or any similar licence granted by TfL, will end automatically.

Exemptions This Licence does not:

Transfer any intellectual property rights in the Information to You or any third party Include personal data in the Information Provide any rights to use the Information after this Licence has ended Provide any rights to use any other intellectual property rights, including patents, trade marks, and design rights or permit You to: Use data from the Oyster, Congestion Charging and Santander Cycles websites to populate or update any other software or database or Use any automated system, software or process to extract content and/or data, including trawling, data mining and screen scraping in relation to the Oyster, Congestion Charging and Santander Cycles websites, except where expressly permitted under a written licence agreement with TfL. These are important conditions of this Licence and, if You fail to comply with them, the rights granted to You under this Licence, or any similar licence granted by TfL, will end automatically.

Non-endorsement This Licence does not grant You any right to use the Information in a way that suggests any official status or that TfL endorses You or Your use of the Information.

Dataset Content (Required)

The data is acquired from 3 sources:
- https://cycling.data.tfl.gov.uk/ 'Contains OS data © Crown copyright and database rights 2016' and Geomni UK Map data © and database rights [2019] 'Powered by TfL Open Data'
- freemeteo.com - weather data
- https://www.gov.uk/bank-holidays