FINAL REPORT BY TEAM 14

A DATA SCIENCE APPROACH TO FORECAST AND VISUALISE ELECTRICITY BIDS IN AUSTRALIA'S NATIONAL ENERGY MARKET

James Chen (z5118771), Alex Lee (z5207126), Yunzhang Liu (z5207360), Yang Cheng Hao (z5211261)

School of Mathematics and Statistics
UNSW Sydney

November 2020

Submitted in partial fulfilment of the requirements of
the capstone course DATA3001

Plagiarism statement

I declare that this thesis is my own work, except where acknowledged, and has not been submitted for academic credit elsewhere.

I acknowledge that the assessor of this thesis may, for the purpose of assessing it:

- Reproduce it and provide a copy to another member of the University; and/or,
- Communicate a copy of it to a plagiarism checking service (which may then retain a copy of it on its database for the purpose of future plagiarism checking).

I certify that I have read and understood the University Rules in respect of Student Academic Misconduct, and am aware of any potential plagiarism penalties which may apply.

By signing this declaration I am agreeing to the statements and conditions above.

Signed: Lyon Liu Date: 21/11/2020

Signed: James Chen Date: 21/11/2020

Signed: Alexander Lee Date: 21/11/2020

Yang Cheng Hao 21/11/2020

Abstract

The National Electricity Market (NEM) is a vital part of the electricity distribution process that facilitates the supply of energy from power generators to day-to-day consumers and businesses in all states and territories in Australia, barring the Northern Territory and Western Australia. Any improvements made to the decision-making or framework of the market would, on a national level, improve profitability for retailers and minimise costs for consumers, while also reducing damage to the environment from energy emissions. Currently, bid data used by the Australian Energy Market Operator (AEMO) is difficult to effectively visualise, and thus may result in less valuable insights into bidding patterns over time, and bidding patterns in response to changes in supply and demand. In this report, we offer a new visualisation technique than the current industry standard of bid stacks, based on Principle Component Analysis (PCA) which can identify market characteristics, using R's Shiny. This application allows for customisation based on raw data, as well as percentile data, and can be customised to account for specific variables. We also identified that the Energy Market has seen drastic increases in pricing over the past 5 years, and have attributed this to lack of competition, given the analysed behaviour of MURRAY and TUMUT4. We have also identified that of Australia's regions, the market structure of QLD, VIC, and SA have higher prices and lower quantity bands, and behave similarly. This information allows clearer micromanagement of the NEM and greater predictive power, to prevent market inefficiencies such as a possible duopoly.

Contents

1 Introduction

The Australian Energy Market Operator (AEMO) is the chief operator of Australia's National Energy Market (NEM). One of its key roles is to coordinate a dispatch process that aggregates electricity from geographically different generators and delivers the required calculated quantities of electricity to wholesale retailers around the Australian energy grid. To distribute the total supply of electricity efficiently, the dispatch process requires up to 10 price bands as bids from each generator every 5 minutes. The NEM functions as a spot market, meaning aggregate demand is calculated by the AEMO to fix a price that ensures that supply and demand are always equal in real time. In the process of visualising these big data in real time, it is hard to extrapolate useful information due to the size and structure of the datasets. It is difficult to also look at the bidding patterns of each generator and to build a more accurate forecast, it is important to evaluate more modern approaches for providing an improved forecast. With the guidance of Senior Economist Oliver Nunn of the AEMC, we discovered that the weakness of current data science structures in the Commission could be seen in its visualisation, with the current state-of-the-art being bid stacks, which can be unintuitive to analyse. Hence, we find alternative ways to wrangle the data, such as by creating more informative variables and then using more advanced data science techniques to gleam insights from the data, such as market trends, and subsequently create more in-depth and informative visualisations.

2 Brief Literary Review

Tomasz NIEDOBA (2014) provides a solution to multi-parameter data visualization complexities by using Principal Component Analysis (PCA), evaluating this techniques ability to qualitatively provide insights to viewers through a 2-dimensional biplot. It is a report that analyses various coal types and finds a way to determine significant differences between the various types of coal. Although we aren't looking at coal for our analysis, we are utilizing similar techniques such as PCA to present high dimensional data on a 2-dimensional plot. This report covers the algorithm behind PCA and provides us with a deeper understanding of this data science technique which will help us to interpret the results of our plots[1].

Visualising high dimensional data with PCA allows us to represent the information of all our variables into an easily interpretable plot. We can identify certain clusters and groups and determine what the values of the variables are for each plot by looking at the loadings and the lengths of the variable vectors on the plot. Also an advantage of using the PCA method is that we do not need to select any parameters in contrast to other methods of multiparameter data visualisation.

Shukla and Naganna (2014), reviews K-means and explained the method. Most importantly they highlighted that outlier must be removed for K-means to produce a sensible result, 'It has been observed by several researchers that, when the data contains outliers there will be a variation in the result that means no stable result from different executions on the same data'. Also, K-means produce 'not optimal results' when 'clusters are of different size, different densities and non globular shapes'. We are not sure our data have globular shapes, so maybe be other technic should be considered and applied[2].

Hagenbuchner and Tsoi (2016) evaluates Neural Network Models used by the AEMO and provides possible solutions to the identified problems in their current models. It is a report based on the Australian energy market and thus is relevant to our analysis as we deal with the same bidding data. It covers relevant applications of data science techniques that we can apply to our own report such as Unsupervised Learning methods like SOM, k-means clustering and prediction based on historical data[3].

3 Materials and methods

3.1 Software

We have chosen to use RStudio and a multitude of packages within this open-source software to do our analysis on. These packages include:

Dplyr, readxl, lubridate, shiny, gridExtra, ggplot2, shinyWidgets, tidyverse, cluster, factoextra.

3.2 Description of the Data

Initially we were given 2 files of data in excel spreadsheet format over Dropbox (Tab A3.2).

3.3 Pre-Processing Steps

General Set Up

Step 1 – Firstly, the data files were imported to R using the read.csv() command, naming these tables 'biddayoffer' and 'bidperoffer' to improve readability.

Step 2 – We then filtered for BIDTYPE = "ENERGY" for both tables using the filter() command as we were only concerned with this specific bid type (as instructed by our client).

Step 3 – The next step was to drop any irrelevant variables by selecting only what we needed from the two data frames using the select() command and renaming the tables to 'biddayoffer_final' and 'bidperoffer_final'.

Step 4 – We merged these two tables together into 'bid_merged' using merge() by DUID, SETTLEMENTDATE, VERSIONNO and OFFERDATE.

Step 5 – We had to reformat the variable INTERVAL_DATETIME by utilsing as.POSIXct(strptime()). This was done so that we could create some extra variables such as Year, Month, Month_name, Day_of_Month, Day_of_Year, Day name, Week, Hour and Minute from the reformatted date time variable.

Step 6 – We also found a list of generators and their fuel types as well as their regions on an external source. We imported this list using read_excel() and did a left_join() onto 'bid_merged' by DUID which resulted in an additional 2 columns.

Step 7 – Also we removed all tables except for 'bid_merged' to reduce overcrowding in the global environment.

[refer to **7.1 Initial Steps***]*

PCA Set Up
Step 1 – In order to set the PCA up, we created objects 'p' and 'b' containing the PCA data for pricebands and bandavails respectively from bid_merged as we wanted to have separate plots for each band types using the prcomp() command.

Step 2 – Next, we stored the summary information from the prcomp() objects to 'sp' and 'sb' for pricebands and bandavails respectively.

Step 3 – Then we created two tables 'bid_merged_pb' and 'bid_merged_ba' by extracting the priceband variables and bandavail variables respectively from bid_merged. We then added the PCA coordinates to their respective tables using the cbind() command.

[refer to **7.1 PCA Setup***]*

In order to represent time data on the 2-dimensional PCA plot, we prescribed each unique year a colour using the rainbow() command in conjunction with nested ifelse statements.
[refer to **7.1 Year Colours***]*

To make sure that our PCA was working properly we tested our results by creating a new observation that contained the averages of each price band and quantity band. Since the coordinates given to this observation via the PCA are at the origin, we can confirm that our method is working accurately.

[refer to **7.2 Diagnostic for PCA plot***]*

3.4 Data Cleaning
In terms of missing data in 'bid_merged' we have 548,600. We found that there was some missing data in INTERVAL_DATETIME which resulted in missing data in the other date variables we had created in Step 5 above for those observations. Thus, using sum(is.na()) we were able to cut down the missing data to 548,086.

The rest of the missing data were found to be contained in the Fuel_Type and Region which were the 2 variables that we added in Step 6 above. This missing data is acceptable as that was what we were expecting from an external source and our calculations will not be affected by these missing values anyhow. It just

means that some DUIDs were not mapped correctly to the DUIDs provided by our client. Furthermore, we made sure that there weren't any duplicate rows by using the distinct() command. *[refer to* **7.1 Data Cleaning***]*

3.5 Assumptions

Assuming that we do not have zero variance in any of our variables.

3.6 Prices at Percentiles of Maximum Generation A major problem that was anticipated was the difficulty in gleaming insights from the irregular banded structure of the data, given its incoherent nature and high dimensionality. By generating values based on the price at which each generator is willing to offer 25%, 50%, 75%, and 100% of its maximum generation, we can represent both price and quantity in 4 variables, practically reducing the multidimensionality problem by five-fold.

Moreover, there is an observed tendency of generator bids to not use the full capacity of the flexibility offered by the 10 price bands and quantity bands offered by the AEMC; that is, a vast majority of bids made submit bids only in one or two quantity bands, making the price bands values correlating to a quantity band of 0 redundant. The percentile variables overcome this problem by basing its price and values on the percentile of maximum generation, providing a smoother, two-dimensional curve akin to supply curves in basic economics which are easier to derive insights from.

3.7 Visualisation Methods

Using R Shiny, we created a tool to help visualize the multidimensional bid data. We did this via Principal Component Analysis (PCA) by plotting observations as coordinates in terms of the first 2 principal components PC1 and PC2. Additionally, we also added a bar chart next to the plot with 3 columns for each band in order to help explain the results to the user. We also added another tab that combines price bands and quantity bands together using 4 percentiles explaining at what price a generator provides 25%, 50%, 75% and 100% of its quantity.

Features and General Overview

Here is a snapshot of what the visualization tool looks like in action.

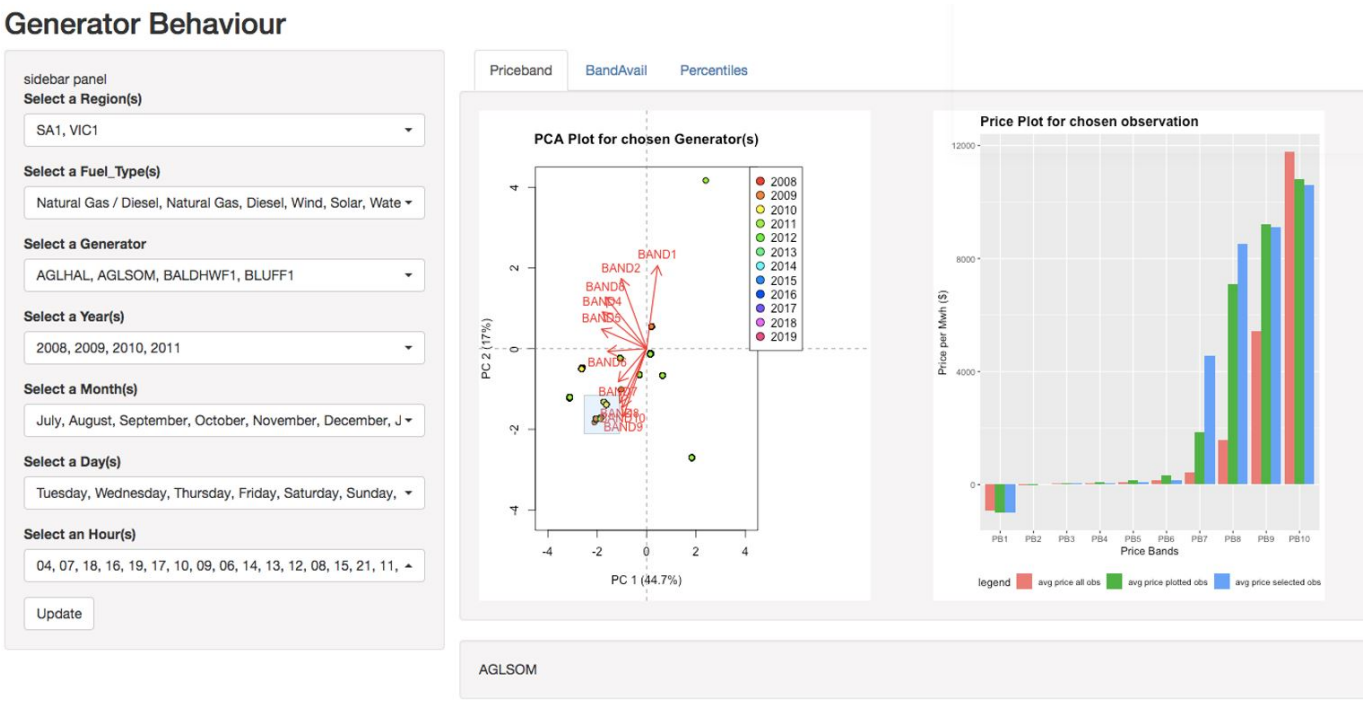


Figure 3.6.1: Overview of R Shiny app

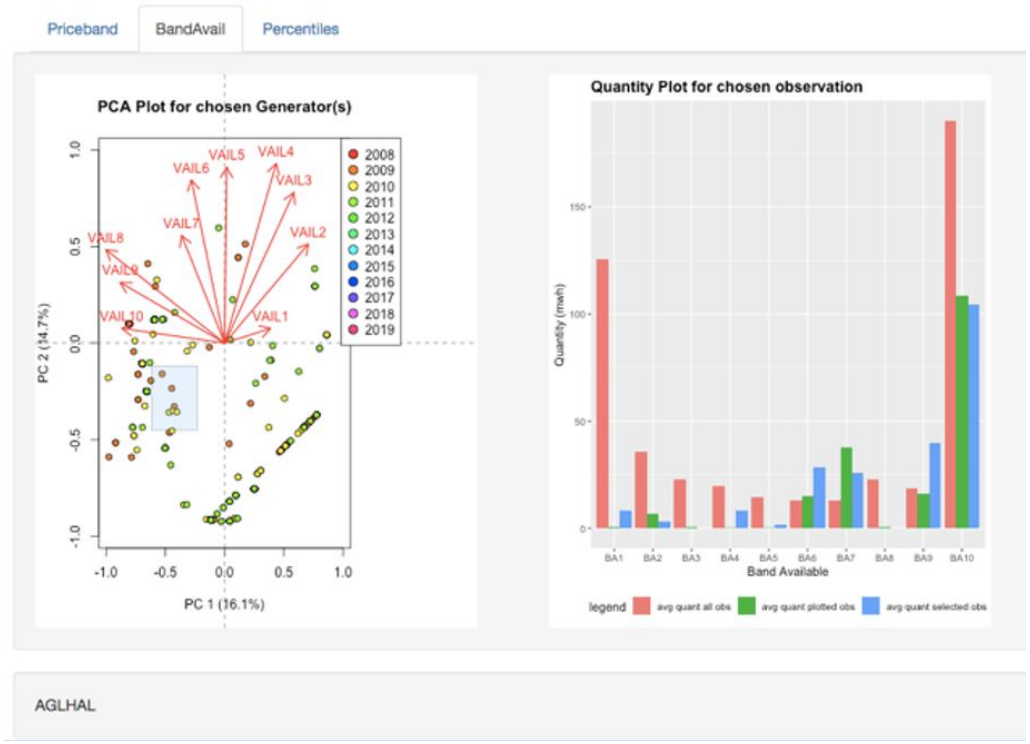


Figure 3.6.2: PCA plot and Quantity Plot for Band available data

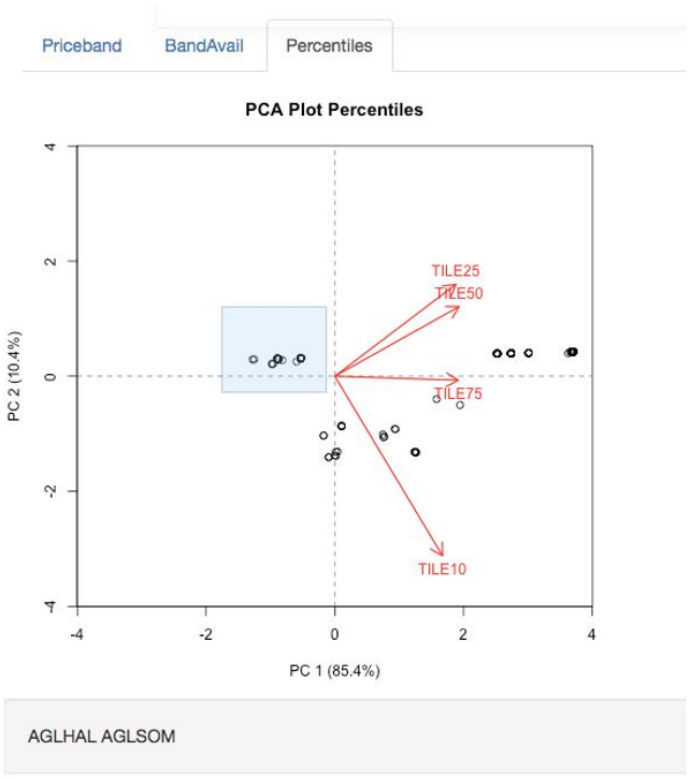


Figure 3.6.3: PCA plot percentile

As shown above (Fig 3.6.1, Fig 3.6.2 and Fig 3.6.3), we can select region, fuel_type, generator, year, month, day and hour to plot these observations. Each dropdown input is conditional on the dropdown(s) directly above it which means that you won't be selecting a combination that doesn't exist ensuring a valid plot each time you press update.

There is also a tab in which you can switch from Priceband and BandAvail and Percentiles as shown above which will then show their respective plots. We have also added a feature in which you can select a specific area. This selected area (the blue rectangle) will output the unique generators in that selection

in the box below as well as produce the blue bar on the histogram for the first 2 tabs that contain histograms.

Another feature we have added is a zoom option in which you can zoom into the plot by double clicking the selected area which will set the limits of the plot to the extremities of that selected area. To return to the initial bounds you can double click an unselected area and the plot will zoom back out. This zoom feature provides us with the ability to have a closer look into clusters that we would not initially have been able to see by just looking at every point at once.

Interpretations of Outputs
The PCA plots contain points which represent an observation as well as arrows indicating the variables or in our case the pricebands or bandavails. A more detailed explanation is covered in "4.2 Principal Component Analysis" later in the report.

The histogram is split into 3 bars; the red bar is the average price/quant of all observations in the whole data set provided to us which corresponds to the origin on the PCA plot. The green bar is the average price/quant of all the points that are displayed on the PCA plot giving a comparison of these plots compared to the average of all observations. Lastly, the blue bar is the average price/quant of the observations that are selected within the blue rectangle on the PCA plot.

4 Exploratory data analysis

4.1 Initial Analysis

We start with a basic summary of the data with a boxplot of each price band (Figure A4.1.1). From this distribution plot, it is evident that each price band has a wide range of prices shown by the number of points outside of the boxplot whiskers as well as the size of the Inter Quartile Ranges of some of these bands.

6 of the 10 price bands have a first quartile ranging from -0.01 to 72.39 whilst the other price bands behave in more distinct ranges. Most of the price bands have a maximum value that is significantly higher than their respective 3rd quartiles and a minimum value that is significantly lower than their respective 1st quartiles.

Since this boxplot (Fig A4.1.1) is not very useful to us for visualising the distribution of each price band at its current state, we created another set of boxplots without the outliers for a better understanding of the distributions (Fig A4.1.2 and Fig A4.1.3).

To obtain a more informative view on the outliers for each price band, we calculated the upper inner fence and lower inner fence values of each boxplot and also calculated the number of observations that were outside of those bounds (Table A4.1.1).

From this output we can see that price bands 1 to 8 all have a significant number of outliers above the upper inner fence, whereas price band 9 has no outliers at all. If we look at the outliers below the lower inner fence, price bands 2 and 10 have the highest number of outliers whereas price band 9 again has no outliers at all. Furthermore, by looking at the histogram plots for each priceband we can get another view of how the prices are distributed. (Fig A4.1.4)

[refer to **7.2 Initial Analysis Pricebands***]*

On the other hand for Bandavails, we again start with a basic summary of the data with a boxplot of each quantity band (Figure A4.1.5). It is evident that each quantity bands have very wide ranges of quantities within their respective bands. All of the bands have a first quartile at 0 (Fig A4.1.5) with a maximum value that is significantly higher than their respective 3rd quartiles. It is also evident that there are a substantial number of outliers for each band.

Since this boxplot (Fig A4.1.5) isn't very useful to us for visualising the distribution of each band at its current state, we created another set of boxplots without the outliers for a better understanding of the distributions (Fig A4.1.6 and Fig A4.1.7).

To obtain a more informative view on the outliers for each band, we calculated the upper inner fence and lower inner fence values of each boxplot and also calculated the number of observations that were outside of those bounds. (Table A4.1.5)

From this output we can see that price bands 5 to 9 all have a significant number of zeroes resulting in the outer and inner fence values to both be zeroes. If we look at the outliers, band 5 has the highest number of outliers whereas band 1 has the lowest.

Also, by looking at the histogram plots for each band we can get another view of how the quantities are distributed. Most of the observations are situated around 0 (Fig A4.1.8).
Moreover, the following plot of average price at percentiles of quantity offered shows us the general shape of electricity supply, averaged over all generators and times. There is an increasing positive relationship between percentiles and price per MWh (Fig 4.1.9).

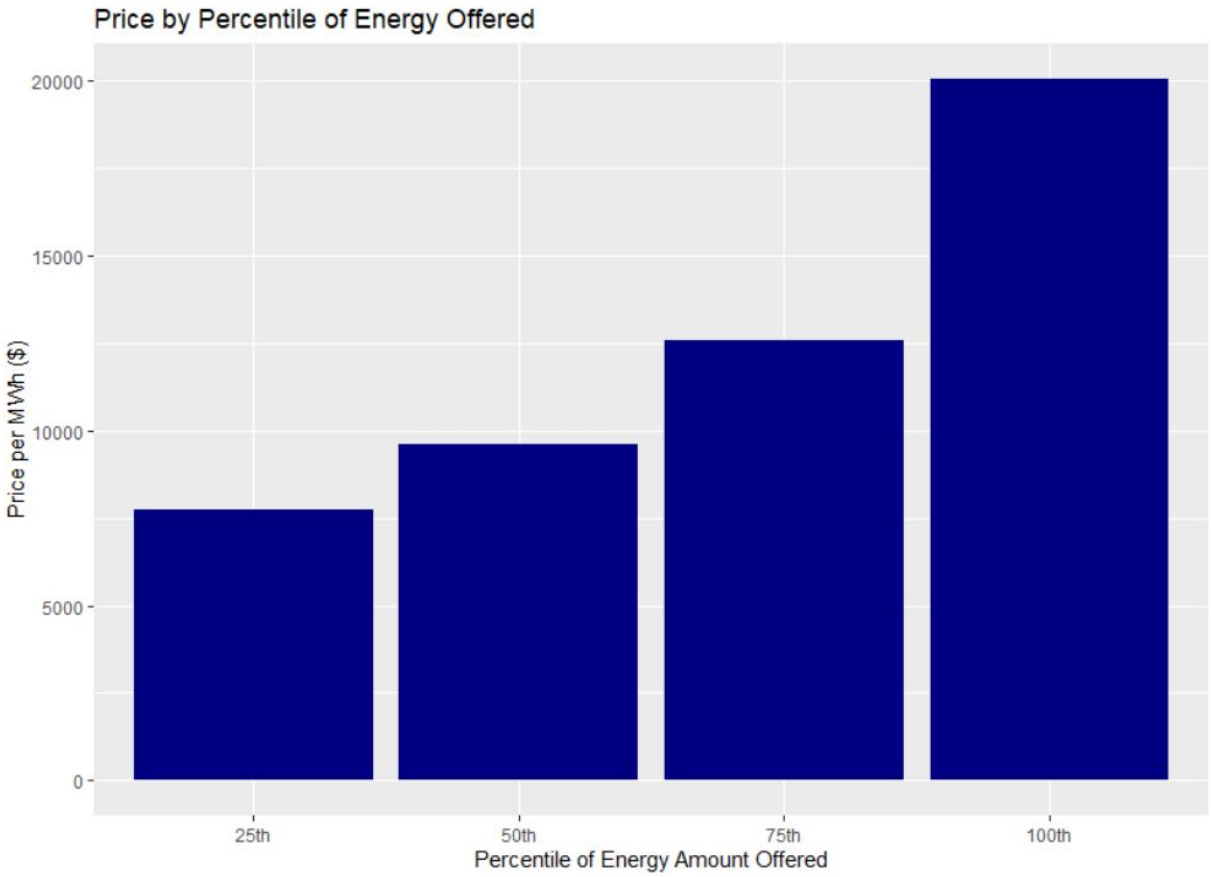


Figure 4.1.9: Relationship between percentiles and price per MWh

4.2 Principal Component Analysis

Principal Component Analysis (PCA) can be used to reduce the dimensions of data by projecting all the variables in terms of principal components. This is helpful in our case as we have 10 price bands and 10 quantity bands which are difficult to visualize as we cannot plot 20 or even 10-dimensional data without using some sort of dimension reduction such as PCA.

We can use PCA to replace a large set of correlated variables with a reduced set of uncorrelated variables however in our case, we won't be reducing the number of variables. Instead, we will be plotting the observations in terms of the first two Principal Components on a 2-dimensional scatter plot whilst keeping all the variable information represented as arrows on the scatter plot. This will provide easier visualisation from which similar types of observations will be grouped together.

For our visualisations, we decided to do separate PCA for price bands and band availables to see how each generator behaves within their bands respectively. An interaction between these bands is analysed with percentiles in another section.

The PCA results provides us with 10 components each for price bands and quantity bands which is a component for each variable. The importance of each component is determined by the proportion of variance each that each component explains. For price bands, the first two components (PC1 and PC2) account for 45% and 17% of the variance respectively which explains 62% cumulatively. For quantity bands, the first two components (PC1 and PC2) account for 16% and 15% of the variance respectively which explains 31% of the variance cumulatively. The components are ordered by how much variance is explained so PC1 is the highest followed by PC2, PC3 and etc. We can visualise the variances of each component via a line plot (Fig 4.2.1 and Fig 4.2.2).

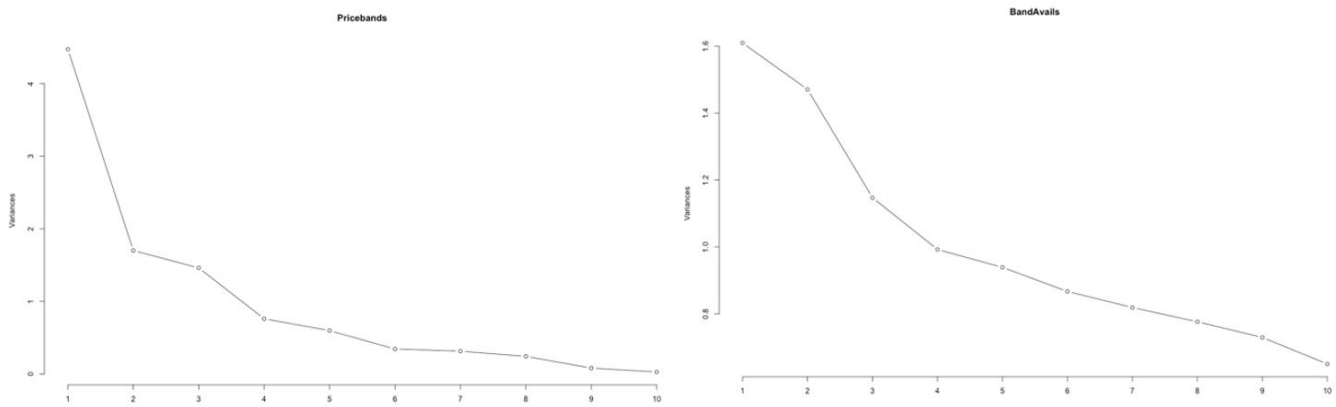


Figure 4.2.1: Plot of variance for each price band. Figure 4.2.2: Plot of variance for each band available *[refer to* **7.2 Scree Plots***]*

From these plots, it is clear that the first component is the most important component with not as much change in the amount of variance explained by the following components. Thus, we can use the first two components to explain the 10 price bands and 10 quantity bands.

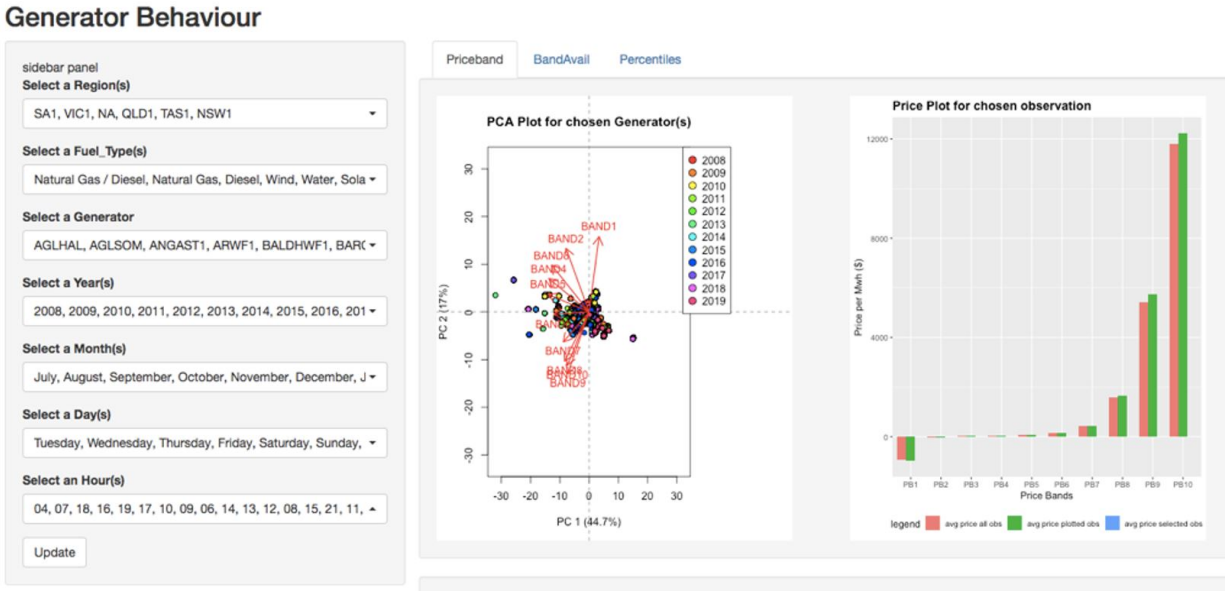


Figure 4.2.3: R Shiny tool displaying PCA plot for chosen generator and Price plot for chosen observation in the price band dataset

Fig 4.2.3 shows every observation for all generators in our data set visualised on a PCA biplot in terms of the first two principal components using our R Shiny tool. The biplot displays points as observations as well as vectors representing each variable (price bands) pointing away from the origin.

When interpreting the PCA plot, we consider the length of the vectors and the angles between them. The length of the vector represents how well the variable is represented by the data. Note that we did a PCA on the whole data set rather than running it for every plot in order to avoid zero variance issues. The vector length is also scaled by a factor to make it more interpretable for viewers however we can still compare the length of a vector in terms of the other vectors. Therefore, a longer vector represents a stronger weighting for its respective variable. The size of the angle between the vectors determine how correlated the variables are to each other. Hence, a 180-degree angle between two variables represents a negative correlation, a 90-degree angle indicates zero correlation, and a smaller angle will represent a strong positive correlation.

Another way to help us determine which variables are having the most influence on our principal components is to create loading plots. Since we used prcomp() to carry out the PCA, the loadings are contained in the PCA object 'p' as the matrix '$rotation'. From Fig 4.2.4 and Fig 4.2.5 we can see which variables have a positive and negative influence on the principal component and by how much in comparison to each other for price bands and band availables respectively. *[refer to* **7.2 Loadings Plots***]*

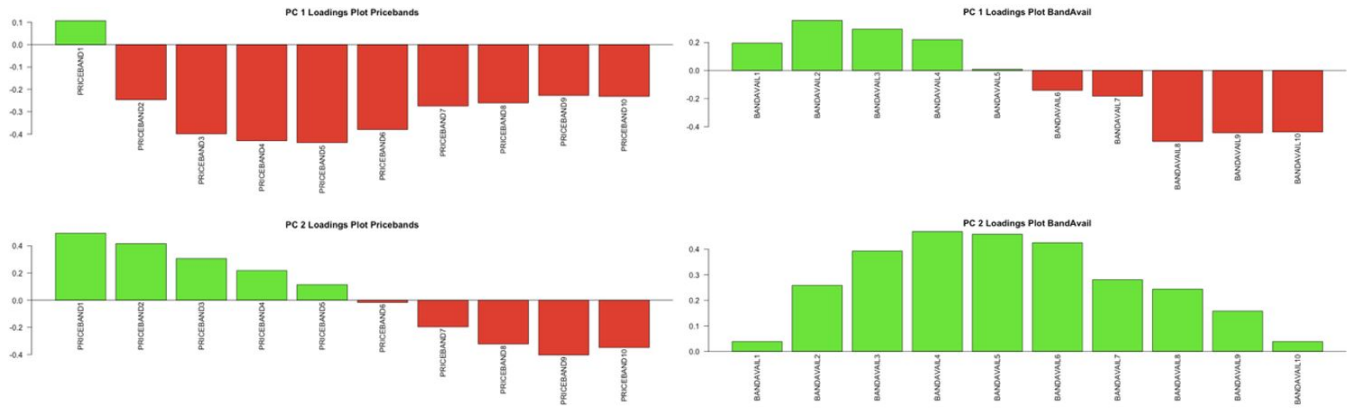


Figure 4.2.4: Loading plots for price bands. Figure 4.2.5: Loading plots for band availables

4.3 Percentiles Interaction

PCA analysis using the variables of PRICEBANDS and BANDAVAILS are useful for gleaming insights individually about the structure of prices and quantities offered by the generators at each band, but cannot demonstrate their combined changes, which may often be of more practical use. Through the calculation of four new variables, PERCENTILES1 – PERCENTILES4, each representing the price at which each generator is willing to offer 25%, 50%, 75%, and 100% of its maximum possible generation respectively, we effectively combine the twenty variables in PRICEBANDS and BANDAVAILS into four succinct variables which are more intuitive to interpret: essentially representing the supply curve for each data row and massively reducing the multidimensionality problem.

Techniques such as PCA and the following clustering techniques which did computation using the raw data values can thusly also conduct the same analysis using this percentile data, to give more interpretable plots useful for our client, at a reduced time complexity.

4.4 Unsupervised learning - Clustering Clustering is a class of algorithm that put observations with similar intra-cluster attributes in the same groups called clusters. Clustering tries to find out whether there is some relationship between the observations in the group. We intend to find different types of binding behaviour for generators in terms of their price offer and quantity offer. We looked at the price band and the band available data separately to see if there is a relationship between what the quantity offered at a certain price band and the quantity at a certain band availables.

In preparation for clustering, we need to ensure that the rows are observations and the columns are numeric variables. Any missing or NA values are removed from the dataset and the data is standardised or scaled to make variables comparable. This involves the transformation of the variables so that they are normally distributed with mean 0 and variance of 1.

4.4.1 K means Clustering Initially, we specify the number of clusters (K = n) to group our data into and the dataset to cluster (bandavailable or priceband) as numeric input values. We then initialise the first K clusters by taking the first 25 instances. The arithmetic mean of each of the K clusters formed are calculated to form the centroid and then K-means assigns a record of each observation to the nearest cluster using the distance of measure called the Euclidean distance. K-means then repeats this process by reassigning the record of each observation to the most similar cluster and recalculates the arithmetic mean of all the clusters (centroids) until all the records converge and the centroids no longer change.

We applied the K-means algorithm for bandavailable and priceband for K = 2, 3, 4, 5. By inspection, we can see that each cluster seems to have different shapes and different sizes for the price band dataset. In the bands available dataset, there is an overlap in the region of each cluster for K = 2, 3, 4, 5 and this suggests that kmeans does not have an intrinsic measure for uncertainty for the examples belong to the overlapping region in order to determine for which cluster to assign each data point(Fig 4.4.1).

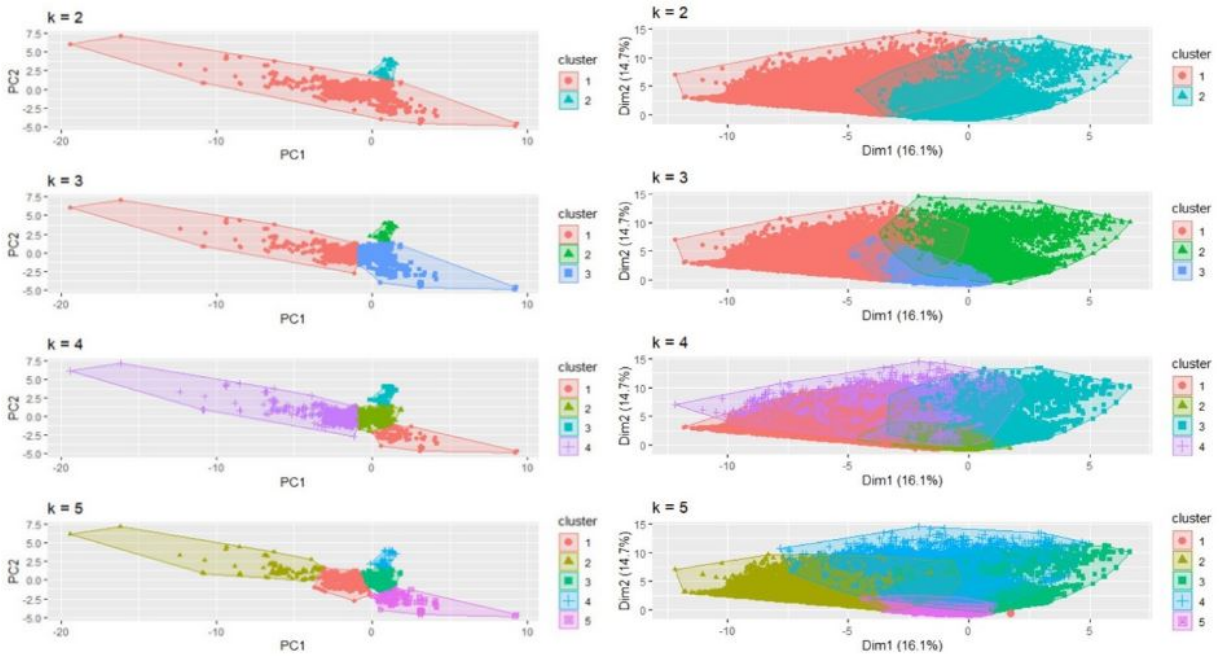


Figure 4.4.1: K-means plots for K= 2, 3, 4, 5 for priceband and bandavails datasets

K means clustering was chosen because it was relatively simple to implement and scaled well to large datasetss. To overcome the disadvantages of outliers and problems with high dimensionality, we removed outliers before clustering and reduced the dimensionality by using PCA on the dataset first. However, K means does not adapt well with clusters of different shapes such as spherical and elliptical clusters.

5 Results

5.1 Average price at different times At a given time interval, average price across bands for a specific generator can be calculate as:

$$AP_G = \frac{\sum_{i=1}^{10} p_i q_i}{\sum_{i=1}^{10} q_i}$$

Where i is the band number, p is the price band and q is the band available.
A good measurement of the market is the average price across all generator, calculated by:

$$AP = \frac{\sum_{i=1}^{g} AP_g}{g}$$

Where g is the number of the specific generator.
Average price calculated may be used as an approximation of the market performance but is not an accurate representation, as bids for low band might get selected but higher band might not get chosen, vice versa.

5.1.1 Observed yearly trends

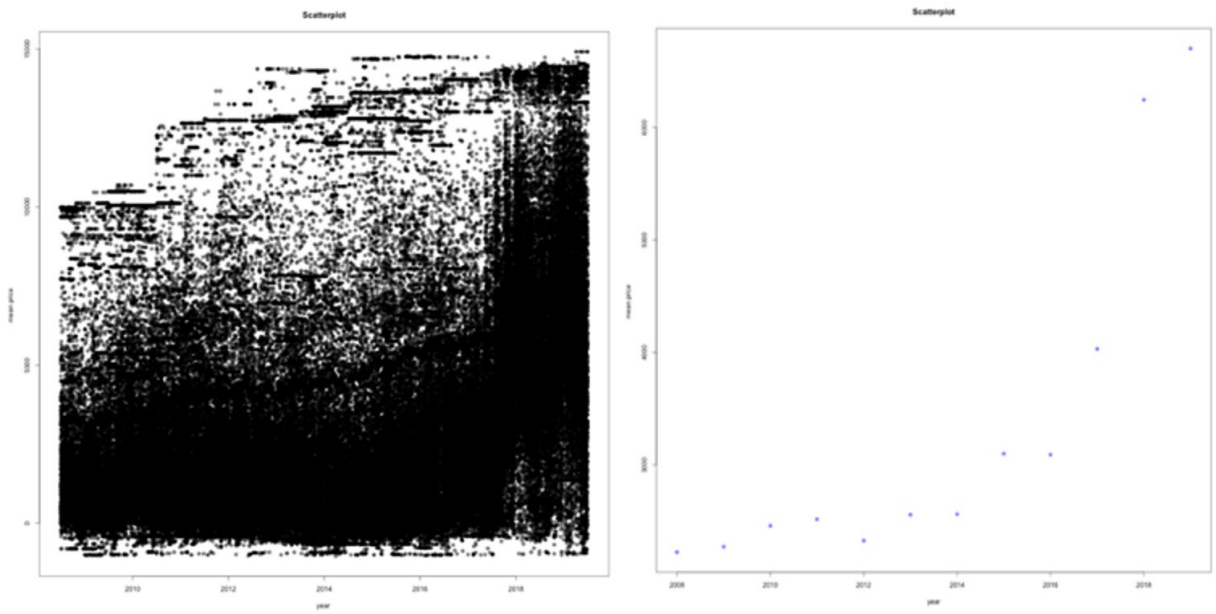


Figure 5.1.1: Initial scatterplot of all prices vs year and scatterplot of average price vs year

Over the 12-year period, despite a large variation in each year, we can observe the price of electricity offered has increased. From the pot of each year's average price, we notice that average price offered has skyrocketed from around 1500 dollars in 2014 to around 7000 dollars in 2019.

Naturally, we suspected increase of population and technological advancement (air conditioning penetration, electric vehicles) rapidly shifted demand in the past 5 years, generator can bid in a higher price and still get selected. However, statistics from Australian Energy Regulator (AER) declined the assumption.

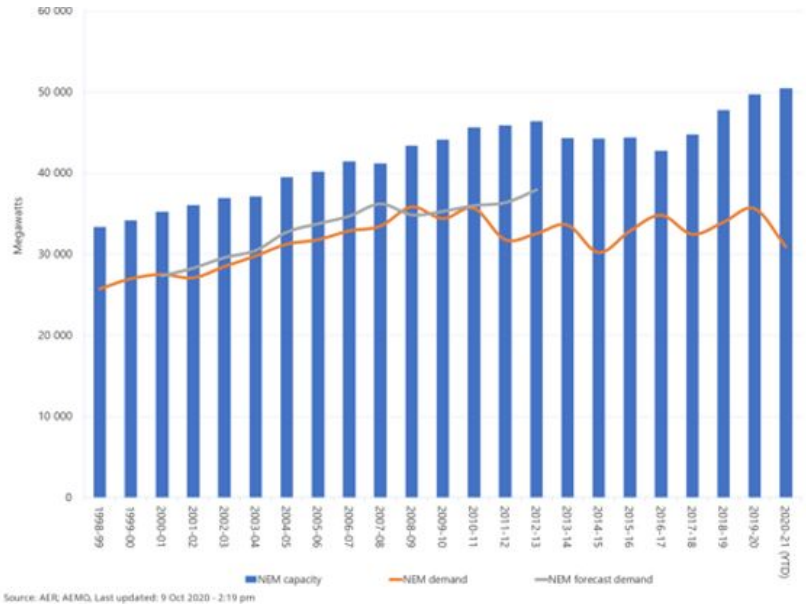


Figure 5.1.1.2: NEM demand, NEM forecast demand and NEM capacity in Megawatts

Figure 5.1.1.2 (from the AER) confirms that from 2014 to 2019, a significant increase in demand was not present. There are two reasons as to why this may occur, the first being generators colluding to increase prices and thus increase profit, and secondly market inefficiency that results from uncompetitive markets. Furthermore, spikes in natural resource prices may have been a contributing factor. We would like to know

which generators contribute to this trend.

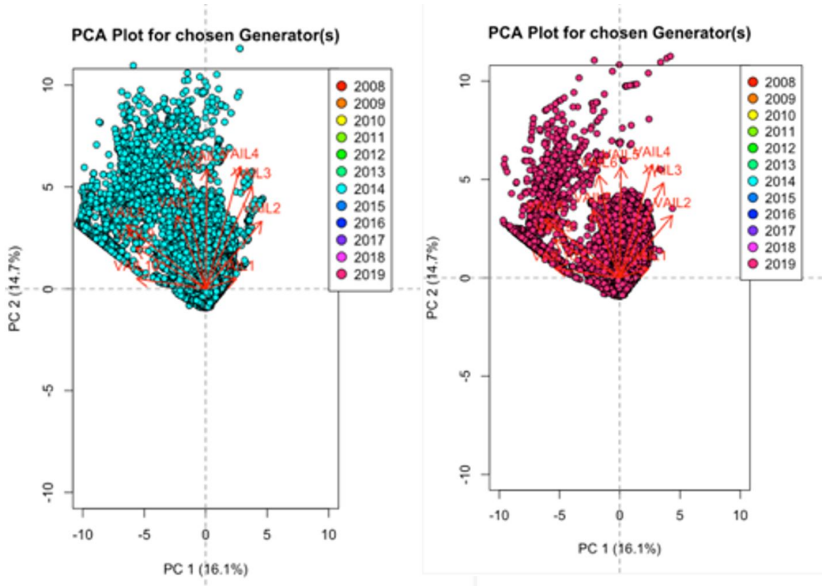


Figure 5.1.1.3: PCA plots for 2014 and 2019 for pricebands dataset
Using the visualisation tool which we developed, we can keep all other variables selected, and directly compare the years 2014 and 2019. By highlighting the points which deviate from the origin of the PCA, we observe that the prices don't vary significantly, only increasing slightly over time. This implies that generators change their bidding strategies by moving around the amount of power offered in their BandAvails rather than moving their prices within their pricebands.

By selecting the areas which deviate from the origin of the PCA band available plot, we notice that there are three major power delivering generators DDPS1, MURRAY and TUMUT3 in 2014. However, it reduced to only two in 2019, MURRAY and TUMUT3. With less generators to generate power in a large quantity, the market price of electricity saw an increase. By selecting the area close to the origin of the PCA BandAvail plot, we can identify the generators which offer small amounts of power for a given price. We noticed that there is an increase in the number of generators, which will lead to an increase in average prices. We further study the remaining big generators MURRAY and TUMUT3.

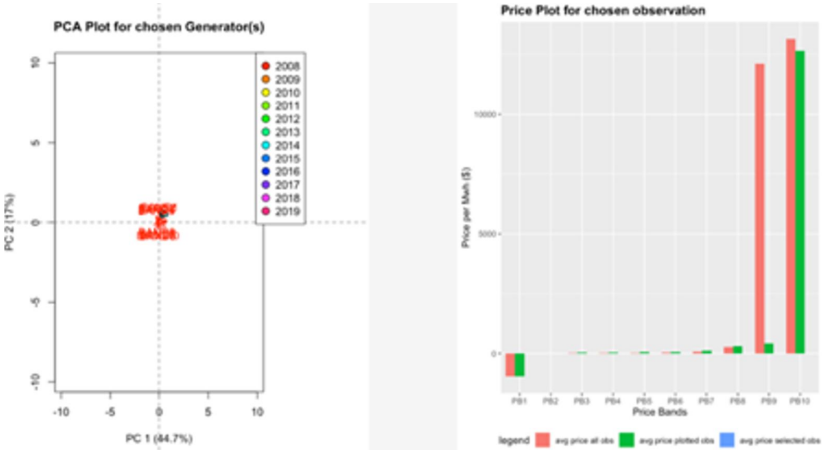


Figure 5.1.1.4: PCA plot and price plot for MURRAY generator for pricebands

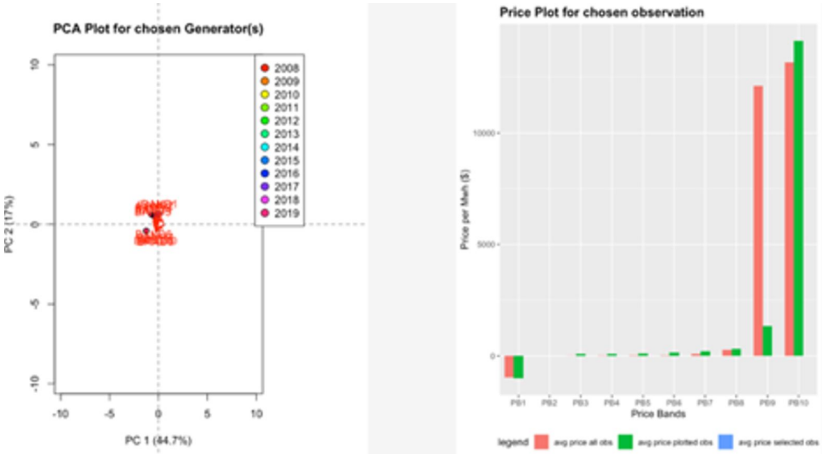


Figure 5.1.1.5: PCA plot and price plot for TUMUT3 generator for pricebands

From the PCA plot and bar charts above we can see that these two generators prices for each band is reasonable compared to the market. A subtle increase in price bands is also presented, adhering to the overall trend of the market.

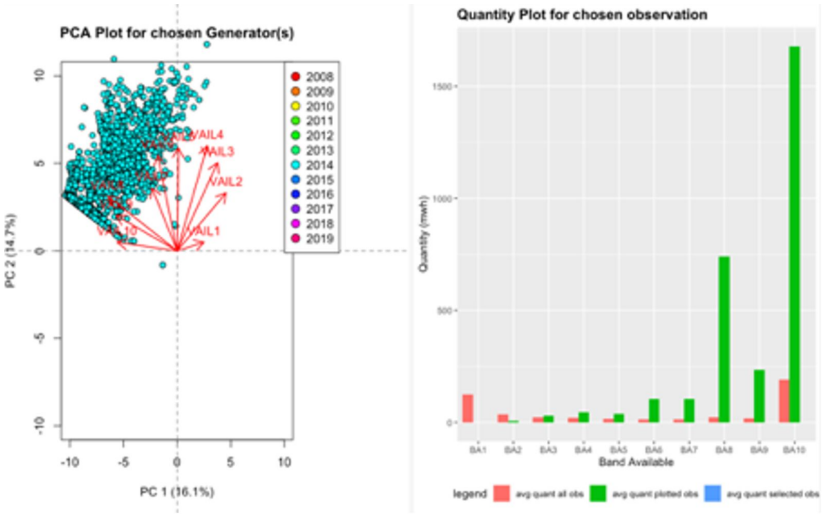


Figure 5.1.1.6: PCA plot and quantity plot for MURRAY generator for BandAvails

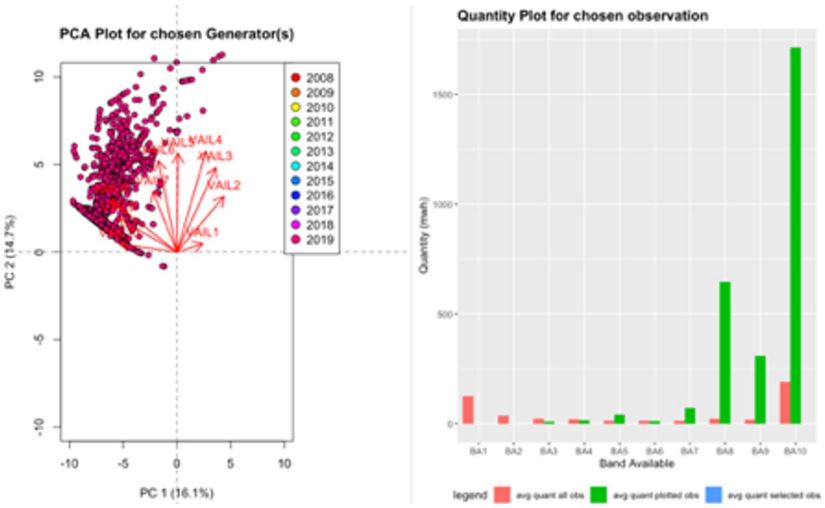


Figure 5.1.1.7: PCA plot and quantity plot for TUMUT3 generator for BandAvails
From analysing the PCA plots and the bar charts above for quantity bands, we can see that BandAvail2 and BandAvail8 has decreased. This confirms that generators are raising their price by reducing the quantity they offer for certain price bands.

In addition, these generators don't offer any bids in the first few bands which suggests that these generators are confident with their bidding strategy and do not need to sell power for cheaper prices. We mention this as we know that some generators sell energy at a significantly cheaper price as a countermeasure to ensure that their generators remain running to avoid shutting down which has a large associated cost. This suggests that big energy providers are stable, offering their electricity at reasonable market prices.

5.1.2 Observed monthly trends

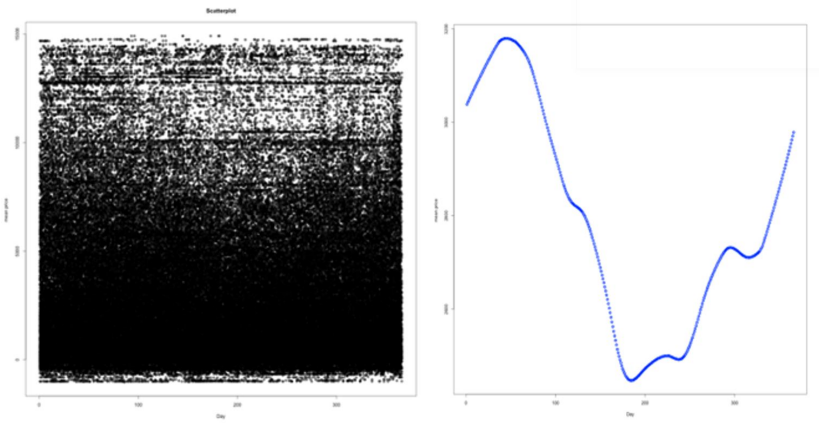


Figure 5.1.2.1: Initial scatterplot of all prices vs months and scatterplot of average price vs months Over the 12-year period given by the data, the market is highly volatile daily which initially suggested that there was no discernible trend at this level. However, with locally weighted polynomial regression, we were able to produce a smooth line in which we spotted Sine-wave like periodic behaviour with a market peak around the 50th day (February) and through around the 200th day (July) of each year.

In Fig 5.1.2.2, statistics from the AER suggests that energy demand has a strong seasonal trend. This confirms to us our previous finding on the price trend. As generator levels of production and optimal capacity is fixed, they are less likely to change the quantity delivered in response to the seasonality of demand. Therefore, compared to July, even if generators bid higher for the same amount of energy, they can still get selected due to a lack of supply in the market.

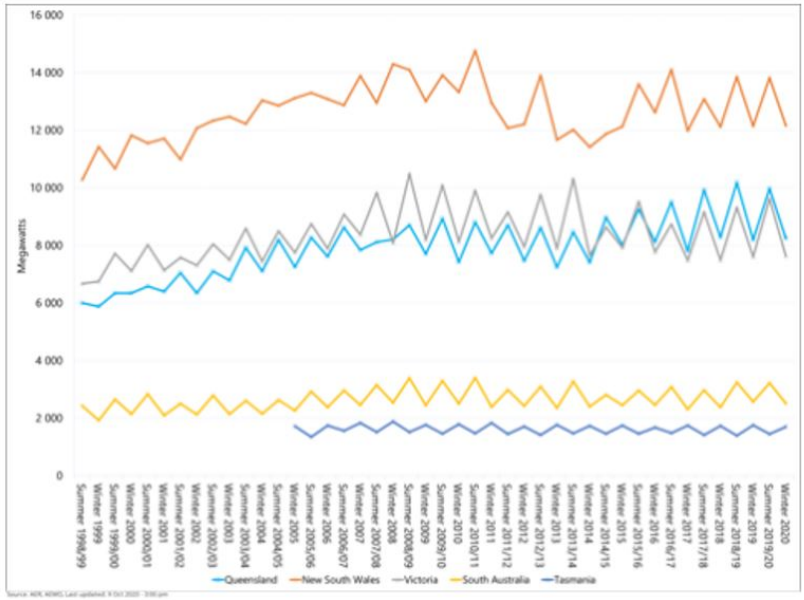


Figure 5.1.2.2: NEM demand by region in Megawatts over time

To study what types of bidding behaviour causes this trend, we use the shiny tool developed. As discussed before, generators change their bids though quantities offered in their respective bandAvails.

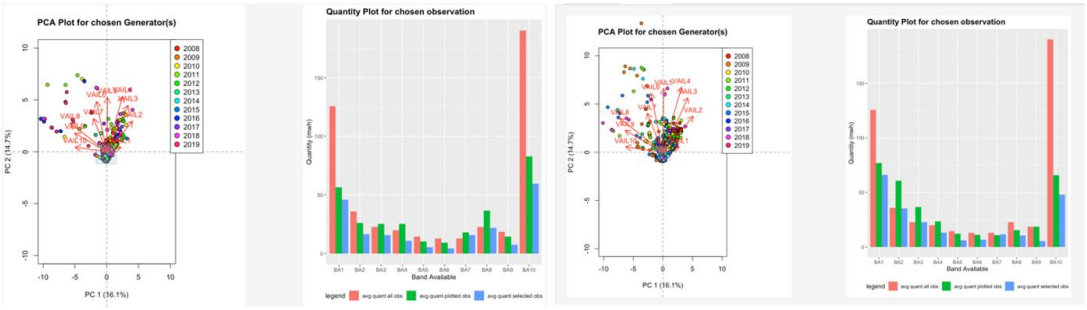


Figure 5.1.2.3: PCA and Quantity plots for BandAvails

From the PCA bandAvail plots above, we observe that the shape is roughly the same. However, there is more bids in July. This suggests that more competition brings the price down. We select the area around the origin and thus observe more generators and compare the bar charts. This low production generator on average offers more electricity at each band, bring the market price down. For example, lets consider the generator AGLHAL.

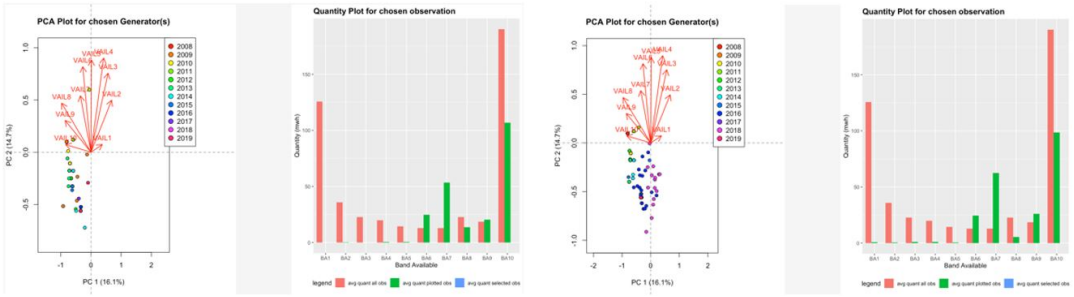


Figure 5.1.2.4: PCA and Quantity plots for AGLHAL generator for BandAvails

The PCA plot comparison shows that AGLHAL offers more bids for low level bands in July as more points appeared on the right and centre. We observe an increase in average electricity offered for band 1 to 7 in the bar chart. Offering on cheaper bands with more quantity will bring the average price down, meaning AGLHAL is bidding less aggressively in July compared to in February. This sets an example for why the market is cheaper, as AGLHAL is one of the many producers changing their bidding strategy.

Observed Hourly Trends

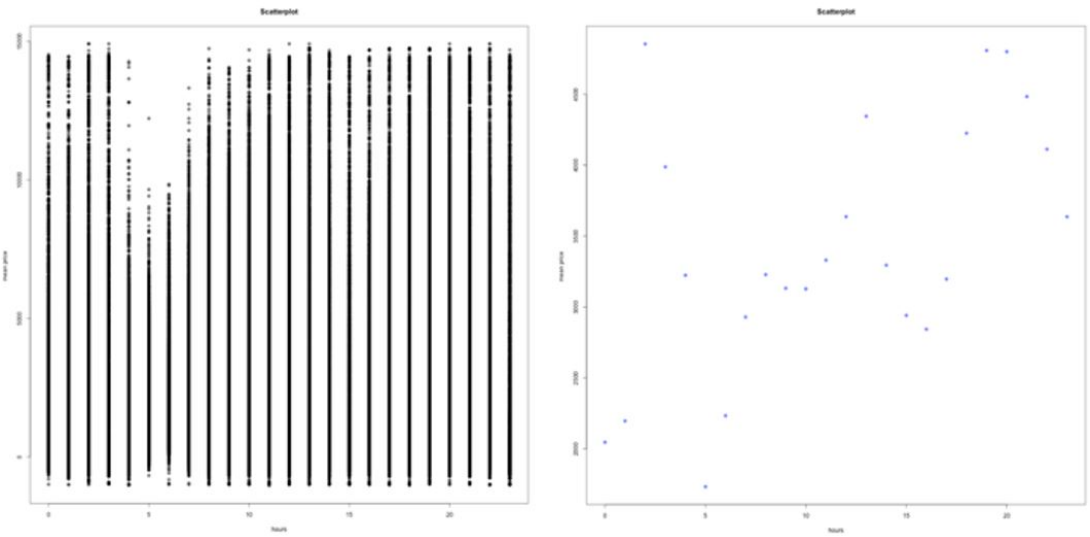


Figure 5.1.2.5: Initial scatterplot of all prices vs hours and scatterplot of average price vs hours

Again, over the 12-year period, the market is volatile every hour, and dips at 5 am. It reaches as high as 15000 dollars per band offer and as low as around –2000 dollars, whereas the mean is around 0 to 5000 dollars. This is likely to be caused by the large variation over the years and months. After plotting the

hourly average graph, we observe no clear cyclical trend of the market. However, we can observe that from 3 am to 5am the market significantly drops on average, with the market gradually rising to its peak from 4pm to 8pm then dropping again gradually from 8pm to 12pm. Nevertheless, across time, the market on average is at its lowest around 5am and peaks around 8pm. Just like for years and months, we would like to know which generators from which company cause this phenomenon.

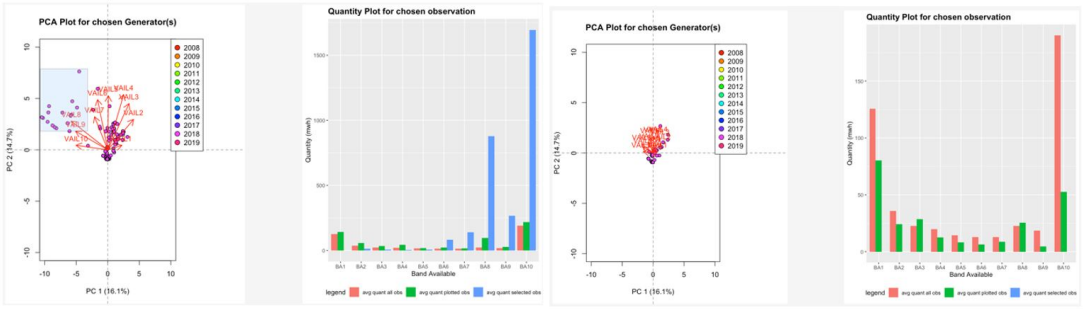


Figure 5.1.2.6: PCA and quantity plots for BandAvails

Firstly, we detect that PCA is more spread out, implying that across all bands, all generators offer more for the same price at 5 am. This is confirmed by the average quantity of plotted observations (green) being higher than average quantity of all observations (red), where for 8 pm the average quantity is always higher. We highlight on the top left corner and observe that these bids offer exceptional amounts of electricity at high bands, bringing the prices down. Those high band bids are again from heavy generators MURRAY and TUMUT3, below we study them closely again.

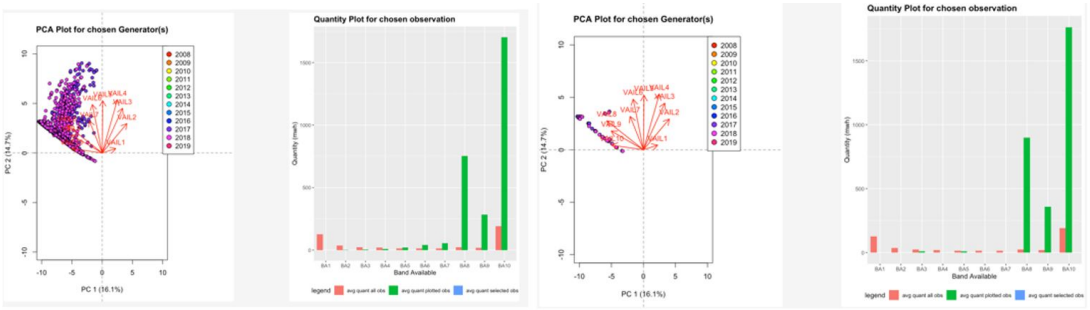


Figure 5.1.2.7: PCA and quantity plots for MURRAY and TUTMUT for BandAvails

The PCA quantity plot is more spread out to the right instead of clustering on the left, implying that two generators bid more on bands 5-7 rather than focusing on higher paying bands 8-10 which brings down their average price down at 5 am. Strategies like this, which decrease their average price by offering more electricity, brings down the market at 5 am, and therefore scarcity causes a rise in the market at 8 pm.

5.2 Overall Regional Generator Behaviour We started our analysis by looking at different subsets of the data by regions (NSW, QLD, VIC, SA, TAS) to get a general understanding of the generators within their respective regions and to identify any strange behaviour. Following is a grouped bar graph outlining the average supply curve for generators of each state, given by percentile data.

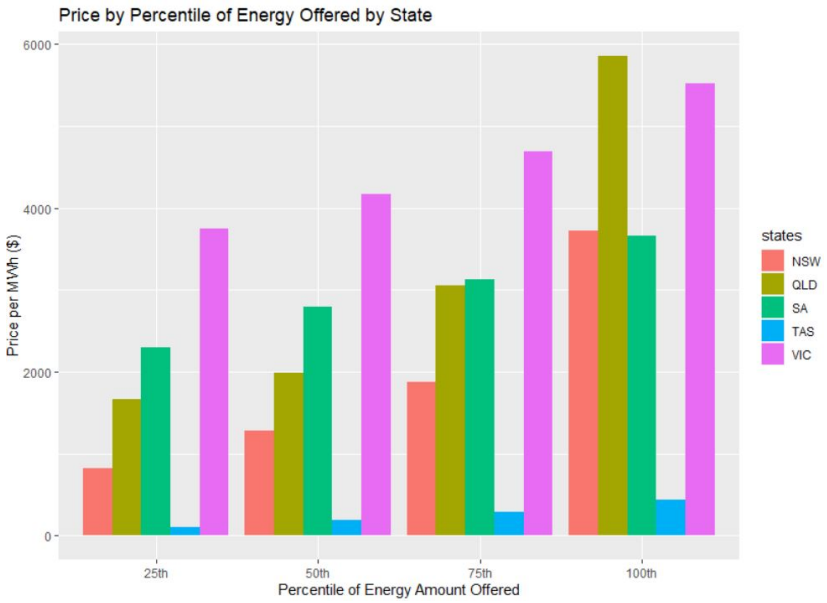


Figure 5.2.1: Price by percentiles of energy offered by different states

NSW

Looking at the average price of plotted points (green bar) in the price band histogram, we can see that NSW has lower prices for most of their bands in comparison to the average price of all observations in all regions (red bar). It is notable that Priceband10 is significantly high however still falling short of the average. For quantity bands, we can see that NSW has higher quantities in all bands in comparison to the average except for bandAvail10.

Additionally, we identified a generator 'SITHE01' to be behaving strangely in terms of price bands. This generator is significantly far away from the other generators in NSW. We can see on the histogram that all the price bands of that selected area are negative which is significantly different to the other observations. Upon further inspection of SITHE01, we found that all the observations in the selected area ranged from 2008 to 2017 which are 14,287 records. The two clusters closer to the origin are from 2017 to 2019 which are 595 observations cumulatively.

In terms of quantity bands, we can see that SITHE01 only has quantities in BandAvail1 and BandAvail10 with the rest of the bands empty.

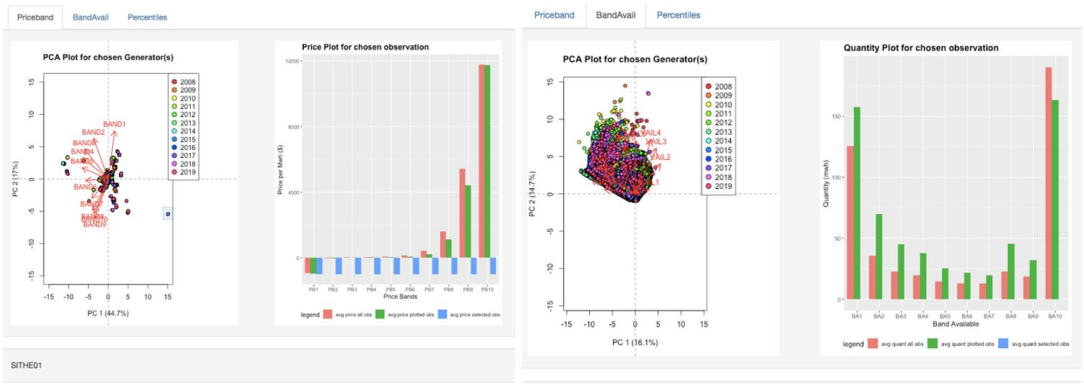


Figure 5.2.2: PCA plots, Price and Quantity plots for NSW

QLD

For price bands we see that QLD has noticeably higher prices in all bands compared to the average and NSW. It is also interesting to note that although QLD has higher prices, we see lower quantities for all bandAvails in QLD compared to NSW suggesting an inverse relationship between price and quantity.

Furthermore, we identified the generator 'SMCFS1' to be behaving differently to the other generators in QLD. The average price of selected observations (blue bar) in the histogram shows that Priceband1 to Priceband9 are negative for this generator which results in the observation being plotted far to the right and lower than the origin as the price bands above the origin have a stronger influence. The earliest records for this generator are from July 2018 which is represented on the points in the selected area with 57 records. 2 months later in September 2018, the generator shifted its behaviour and set their price bands at a more normal price rather than all negative which is represented by the points closest to the origin at 296 observations.

In terms of quantity bands we can see that 'SMCFS1' only has quantities in BandAvail1 with a small amount in BandAvail10.

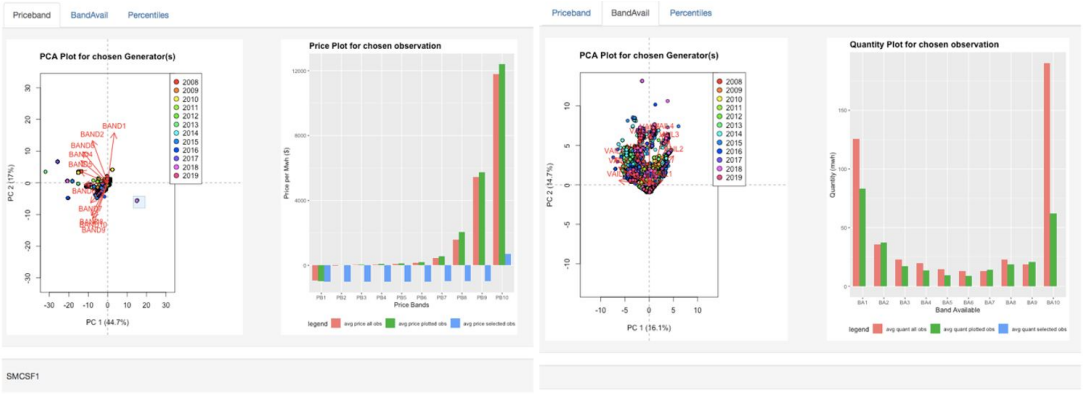


Figure 5.2.3: PCA plots, Price and Quantity plots for QLD

SA

SA has the highest prices in almost all their price bands with Priceband2 being slightly negative. For quantities we can see that SA has the lowest quantities in each band compared to NSW and QLD reinforcing the inverse relationship between price and quantity.

Furthermore, for SA we identified the generators 'QPS1, QPS2, QPS3, QPS4 and QPS5' which seemed to have unusually high prices in price bands 6 to price bands 10 which resulted in the point being to the far left and below the origin. Upon further inspection we found that this was only the case for May of 2013 as the majority of the observations were situated around the origin with records ranging from 2008 to 2019.

In terms of quantity bands, we found that 'QPS1, QPS2, QPS3, QPS4 and QPS5' put majority of their quantities in the last 2 bands with little to no quantity in the others.

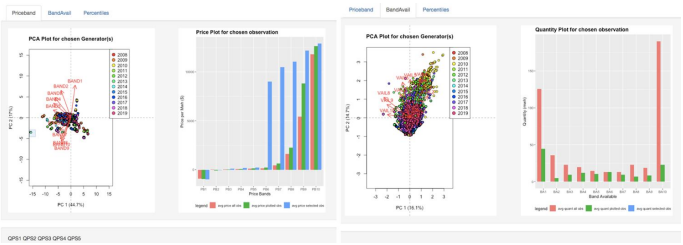


Figure 5.2.4: PCA plots, Price and Quantity plots for SA

VIC

VIC also has prices that are above average for all of their price bands and quantities that are lower than the average by a small amount with bands3 to 8 being higher than average. It seems as though VIC behaves differently compared to NSW, QLD and SA on these terms.

For VIC we identified 2 interesting groups, one to the bottom left in the selected area and one to the top right. Upon further inspection of these two groups, the selected area is 'JLAO1, JLAO2, JLAO3, JLAO4' which recorded only 4 entries in April 2013. The group on the top right is also an insignificant number of records from a group of generators.

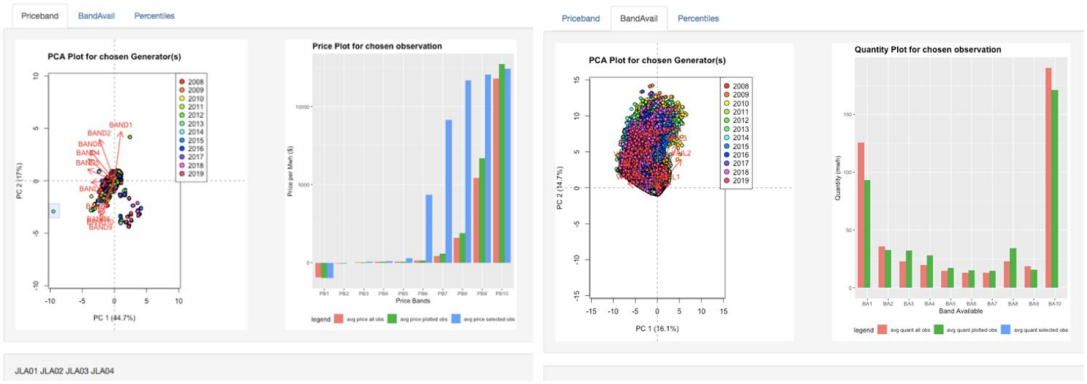


Figure 5.2.5: PCA plots, Price and Quantity plots for VIC

TAS

Tasmania's price bands and quantity bands are all significantly lower than all other regions except for Priceband10 which is very close to the average. It is also notable that bandAvails 9 and 10 show the highest quantities for Tasmania in conjunction with Priceband10 being a very high value.

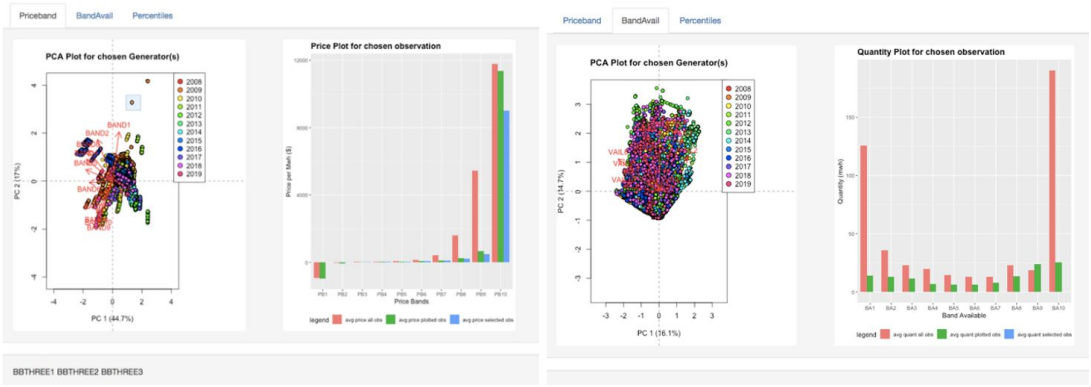


Figure 5.2.6: PCA plots, Price and Quantity plots for TAS

Queensland, South Australia and Victoria all have prices that are above average for most, if not all, of their price bands from 2008 to 2019. South Australia has the highest price average in Priceband9 compared to all regions significantly. New South Wales and Tasmania have prices that are below average for all price bands. Tasmania has a significantly lower average price in Priceband2 to Priceband9 compared to New South Wales for those same bands.

Although QLD and SA have the highest average prices for their bids, they also have very low average quantities in their quantity bands. On the other hand, NSW which was known for lower average prices, has the highest average quantities in their quantity bands. This suggests that higher average prices result in lower average quantities and vice versa.

It is important to note that whilst VIC has high average prices, they are behaving differently compared to QLD and SA in their quantity bands as we can see that VIC actually has relatively high average quantities in their bands. Whilst the quantities are lower than that of NSW, it is still significantly different to QLD and SA which suggests that higher average prices don't necessarily result in lower average quantities in this case.

Note that TAS whilst having low prices also has low quantities, however we suspect this to be due to the differences in population sizes and land size with TAS being known to be the smallest out of the other regions and thus having the least amount of aggregate energy supply.

Regarding strange generator behaviours within each region, we can see that SITHE01 from NSW and SMCFS1 from QLD behave similarly in the sense that they initially had all their price bands at a significantly negative value but end up converging in the end as time goes on. QPS1, QPS2, QPS3, QPS4 and QPS5 from

SA and JLAO1, JLAO2, JLAO3, JLAO4 from VIC were also identified as behaving strangely. However, when investigated further we found that both these instances were either for a very short period of time or were only a few observations.

5.3 Regional Generator Trends and Trajectory

In the last section we analysed each region in terms of their overall shape and identified any strange points within their whole lifespan. In this section we look at how each region changes every year from 2008 to 2019 in order to find some sort of pattern or trend. We did this by splitting the 12 years into 4 groups of 3 starting from 2008/2009/2010, . . . , 2017/2018/2019 for each region.

NSW

In terms of price bands, we can see that there have been significant fluctuations in the last 4 price bands (priceband7, priceband8, priceband9, priceband10). From 2008 to 2019, Priceband10 has been increasing over time and has reached past the average by 2014 ˜ 2016. On the other hand, Priceband8 and Priceband9 were initially increasing in 2011˜2013 however experienced a drop in the following year group 2014˜2016. This is also apparent in the PCA plots as we can see a shift in points downwards over time indicating higher prices in the higher price bands.

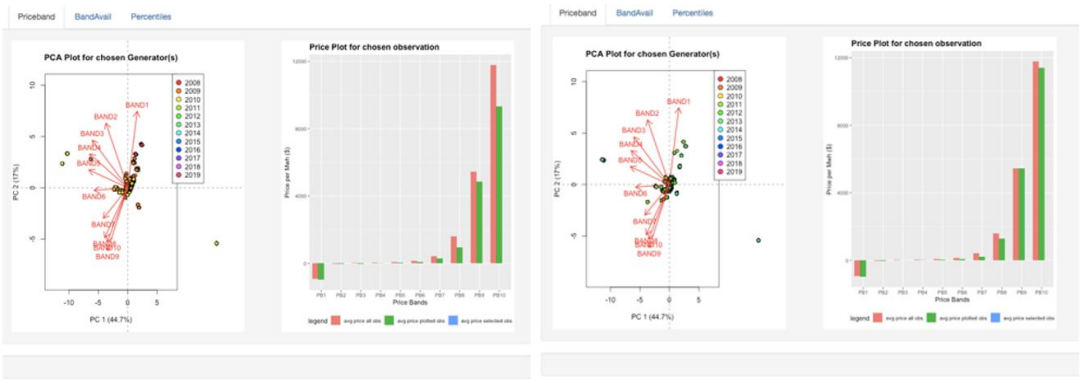


Figure 5.3.1: PCA plots and price plots for NSW regions in 2008 to 2019 (1)

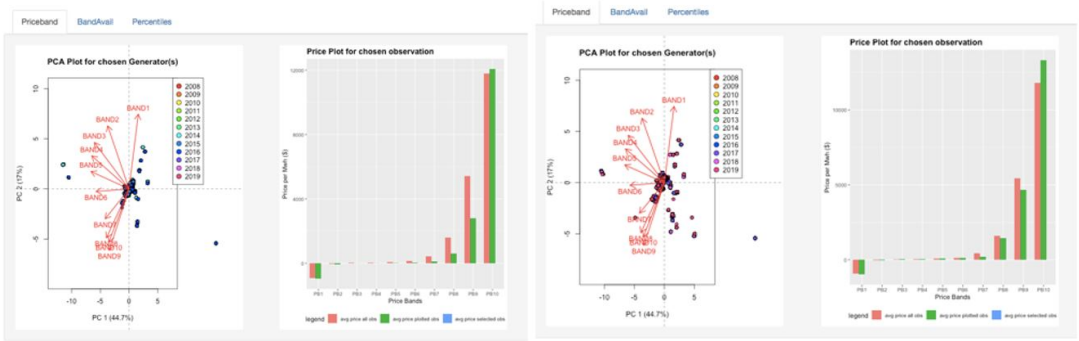


Figure 5.3.2: PCA plots and price plots for NSW regions in 2008 to 2019 (2)

In terms of quantity bands, we can see a major shift in bandAvail2, dropping by almost double the quantity from 2008˜2010 into the following years. In 2011˜2013 we saw higher bands distributed between BandAvail3 to BandAvail8 with BandAvail8 skyrocketing in 2014˜2016. From this we can see that quantity bands in NSW are subject to large changes often fluctuating from highs and lows.

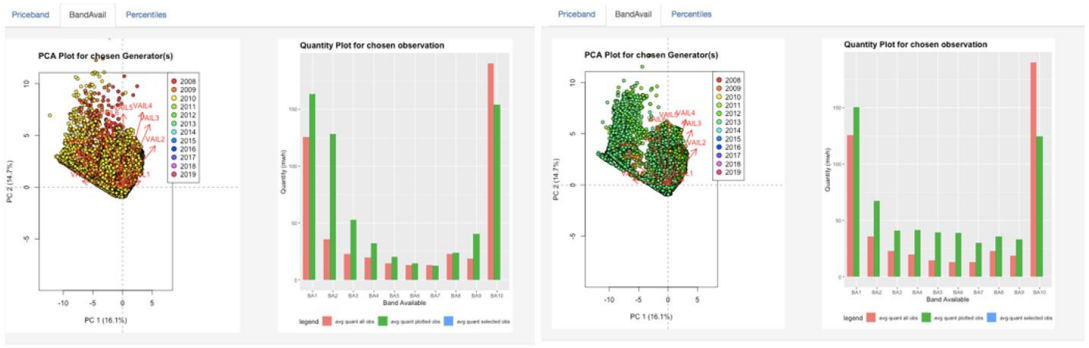


Figure 5.3.3: PCA plots and quantity plots for NSW regions in 2008 to 2019 (1)

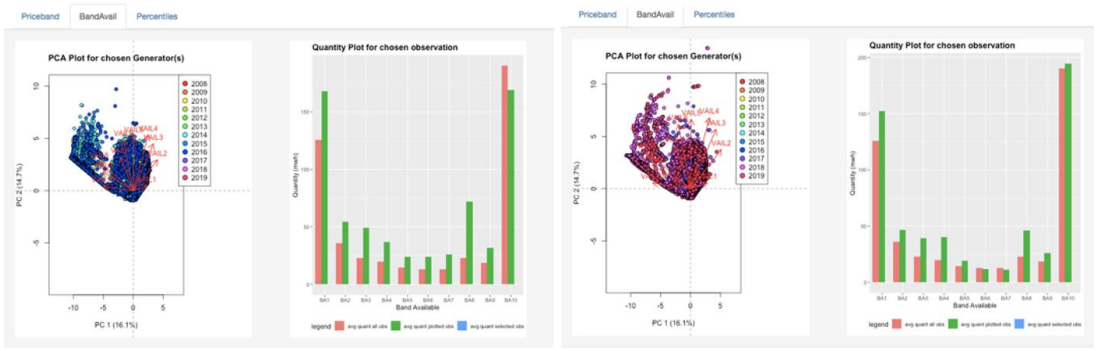


Figure 5.3.4: PCA plots and quantity plots for NSW regions in 2008 to 2019 (2)

QLD In terms of price bands, we can see that there have been significant increases in the last 4 price bands (priceband7, priceband8, priceband9, priceband10). From 2008 to 2019, these 4 pricebands have all increased over time exceeding the average for all pricebands by 2014˜2016. This is also apparent in the PCA plots as we can see a shift in points downwards towards the left over time indicating higher prices in the higher price bands.

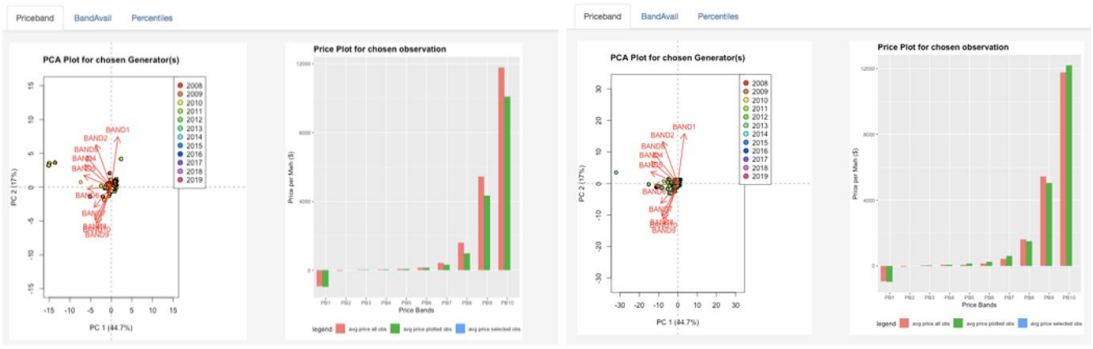


Figure 5.3.5: PCA plots and price plots for QLD regions in 2008 to 2019 (1)

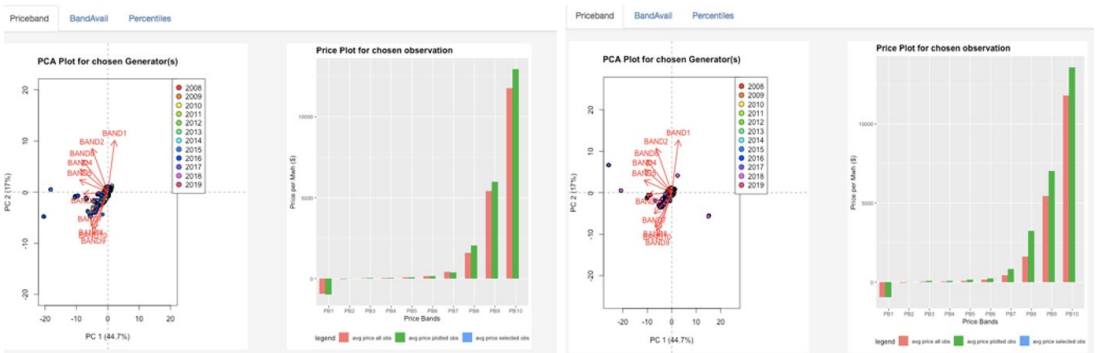


Figure 5.3.6: PCA plots and price plots for QLD regions in 2008 to 2019 (1)

In terms of quantity bands, we are seeing less quantities compared to NSW for the same bands with most of their bands below average. BandAvail1 and BandAvail10 are the highest quantities for QLD with BandAvail2 being the largest in 2011˜2013 (Fig 5.3.7 and 5.3.8).

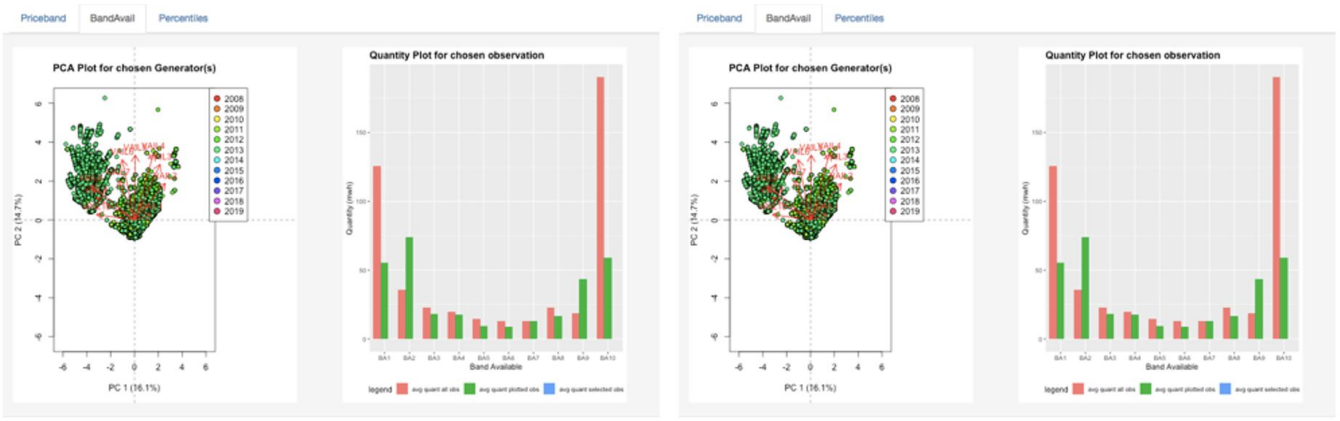


Figure 5.3.7: PCA plots and quantity plots for QLD regions in 2008 to 2019

SA In terms of price bands, we can see that there have been significant increases in the last 4 price bands (priceband7, priceband8, priceband9, priceband10) which is similar to NSW and QLD. We can see in the PCA plots how the points are shifting downwards suggesting this increase in higher pricebands(Fig 5.3.9 and Fig 5.3.10).

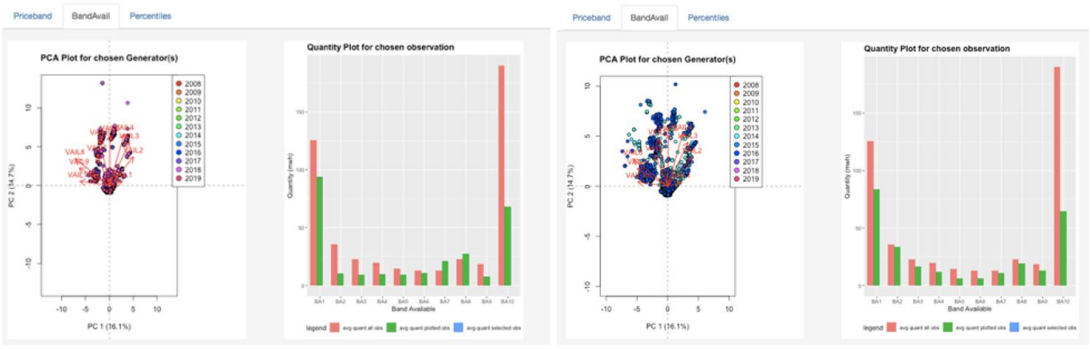


Figure 5.3.8: PCA plots and price plots for SA regions in 2008 to 2019 (1)

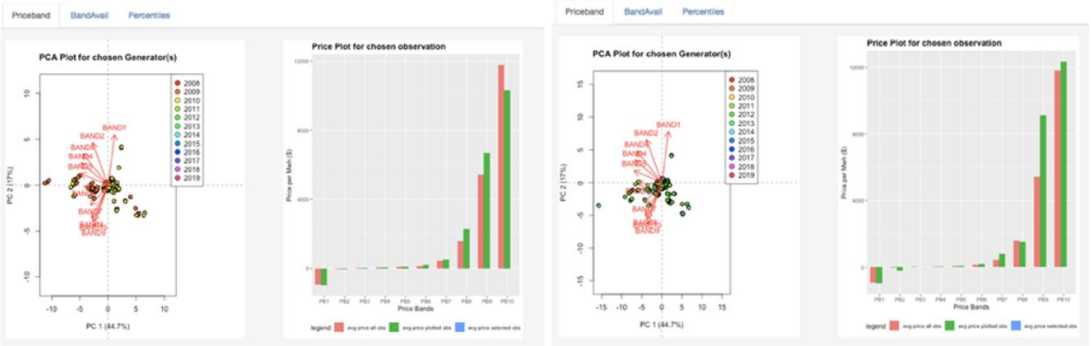


Figure 5.3.9: PCA plots and price plots for SA regions in 2008 to 2019 (2)

In terms of quantity bands, we are seeing low quantities compared to the other states with bandAvail1 being the largest contributor. It is notable that BandAvail10 has increased in 2017˜2019 for SA but is still far below the average.

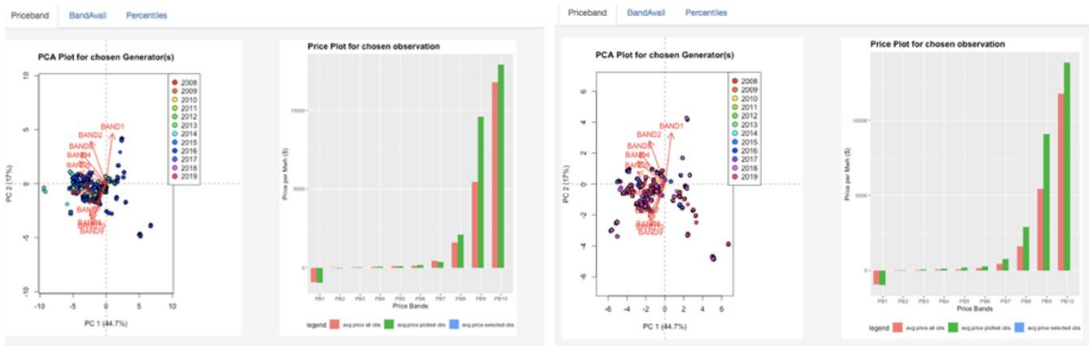


Figure 4.3.10: PCA plots and quantity plots for SA regions in 2008 to 2019 (1)

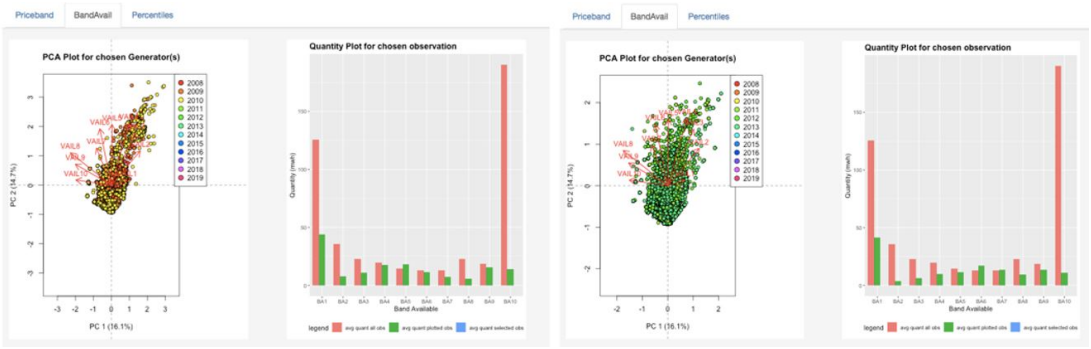


Figure 4.3.11: PCA plots and quantity plots for SA regions in 2008 to 2019 (2)

VIC

In terms of price bands, we can see that there have been significant increases in priceband9 and priceband10 whilst the other bands have been dropping over time (Fig 5.3.13 and Fig 5.3.14).

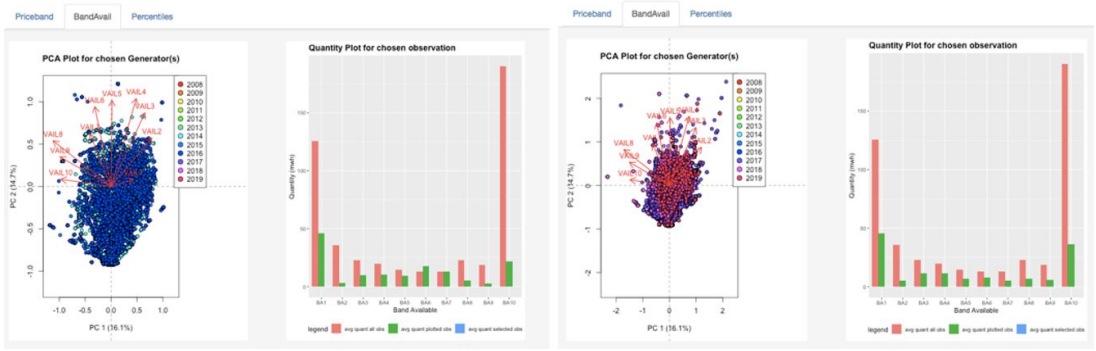


Figure 5.3.12: PCA plots and price plots for VIC regions in 2008 to 2019 (1)

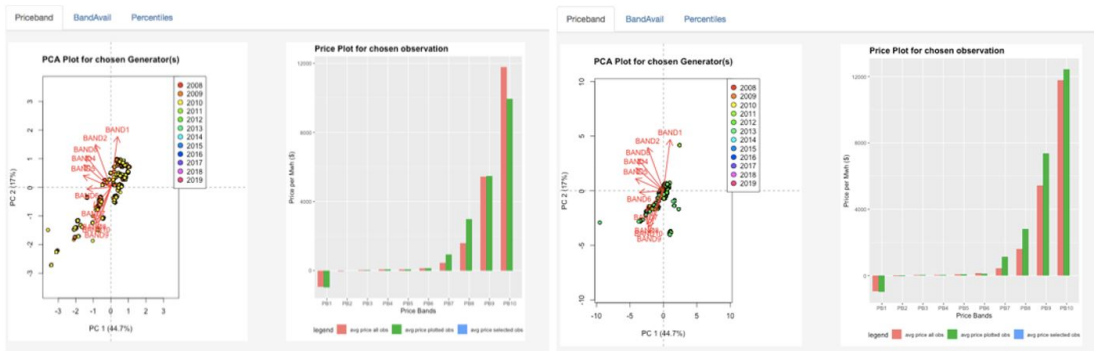


Figure 5.3.13: PCA plots and price plots for VIC regions in 2008 to 2019 (2)

In terms of quantity bands, BandAvail10 has dropped from far above the average to below average and has stayed there after 2011˜2013 inclusive. BandAvail2 to BandaAvail9 used to be above average in 2008˜2011

but now only BandAvail3, BandAvail4 and BandAvail9 remain above average. On the other hand, BandAvail1 has been increasing over time.

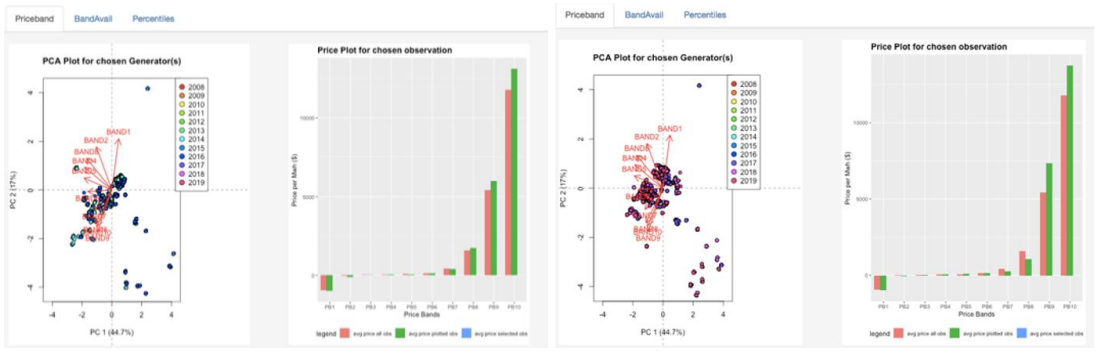


Figure 5.3.14: PCA plots and quantity plots for VIC regions in 2008 to 2019 (1)

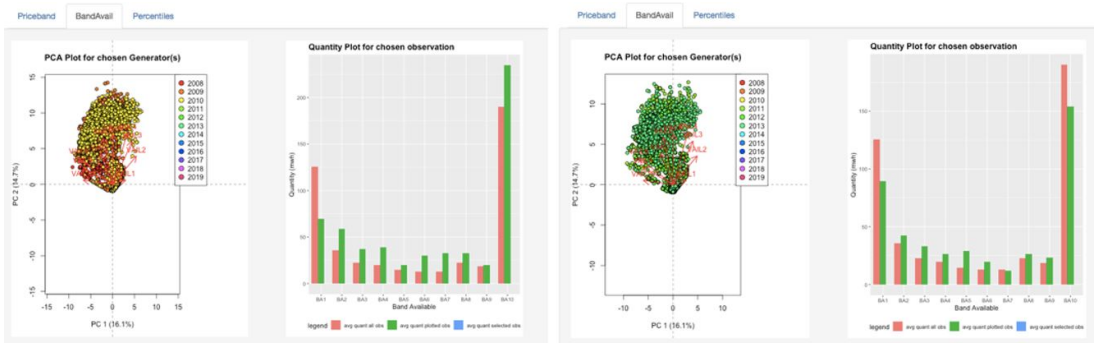


Figure 5.3.15: PCA plots and quantity plots for VIC regions in 2008 to 2019 (2)

TAS

Tasmania started 2008˜2011 with lower-than-average price bands for most of their bands. We can see that over time, whilst the other bands shifted even lower, we see priceband10 has been continually increasing reaching above average in 2011˜2013.

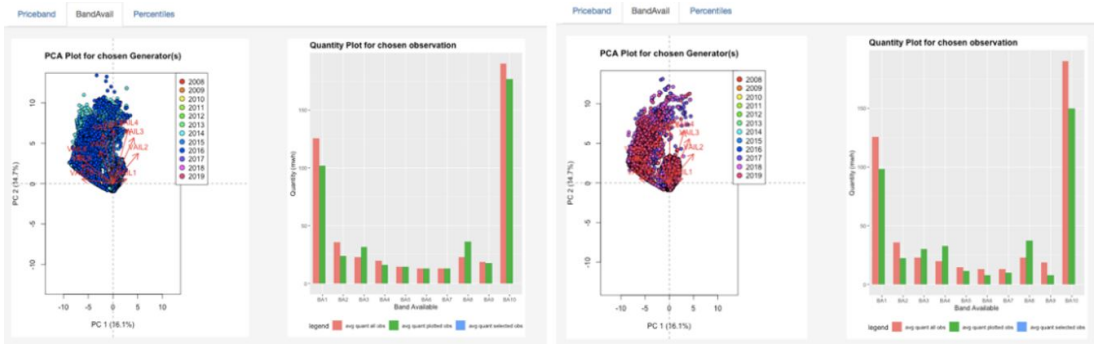


Figure 5.3.16: PCA plots and price plots for TAS regions in 2008 to 2019 (1)

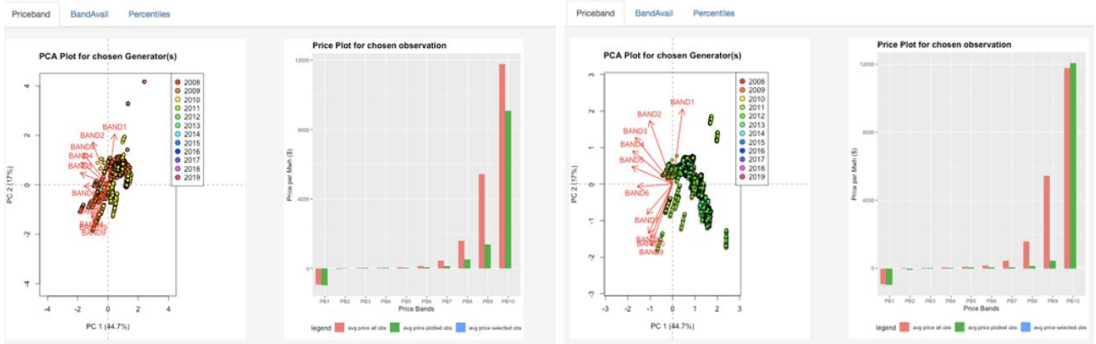


Figure 5.3.17: PCA plots and price plots for TAS regions in 2008 to 2019 (2)

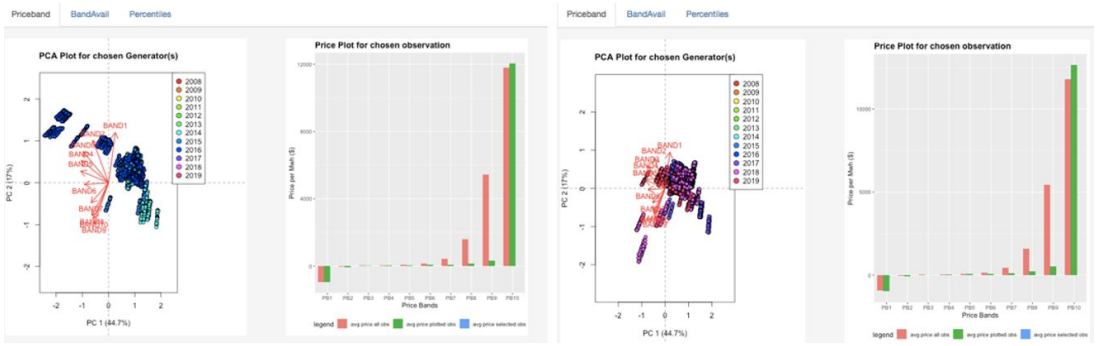


Figure 5.3.18: PCA plots and price plots for TAS regions in 2008 to 2019 (3)

Tasmania's quantity bands are also below average which isn't a surprise considering the population density and limited land for energy production in this region compared to the other coastal regions of Australia. It is notable however that BandAvail9 has been increasing over the years above the average for that band whilst the other bands remained rather fixed.

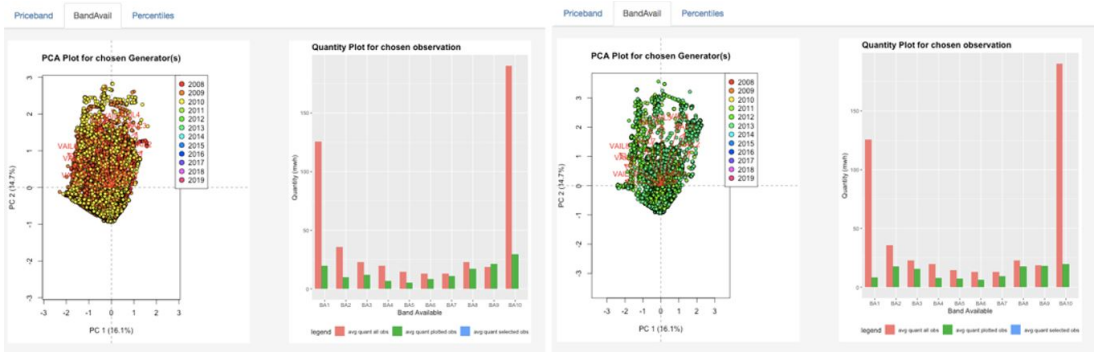


Figure 5.3.19: PCA plots and quantity plots for TAS regions in 2008 to 2019 (1)

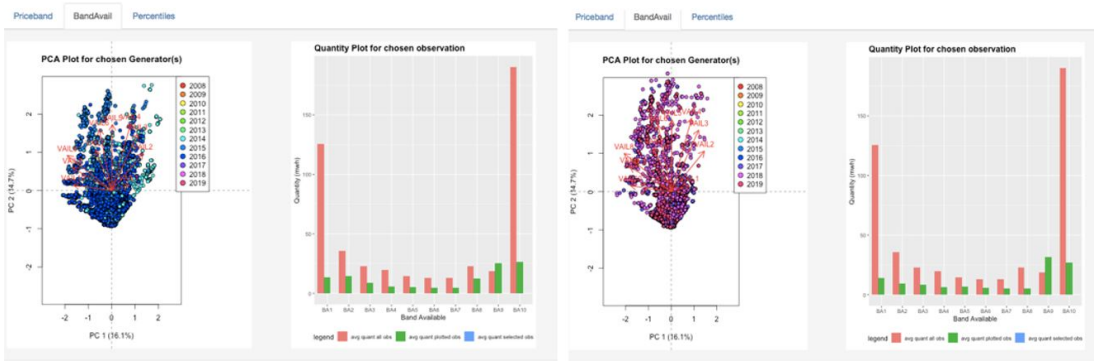


Figure 5.3.20: PCA plots and quantity plots for TAS regions in 2008 to 2019 (2)
We can see that all regions have an increasing price trend as time goes on. QLD and SA have increasing average prices for majority of their price bands with VIC increasing mainly in Priceband9 and Priceband10. On the other hand, NSW and TAS have both increased average prices in Priceband10. Nevertheless, this increasing average prices of all regions may indicate increasing demand for energy as population sizes increase by the year.

In terms of quantity bands QLD, SA and VIC all have been seeing lower average quantities in all their bands except for BandAvail1 and BandAvail10. NSW is experiencing increasing average quantities for BandAvail10. TAS has slowly been increasing quantities for all bands but is again still significantly lower than the average.

6 Conclusion

Our analysis was mostly focused on trying to identify any trends and behaviours of generators in Australia. We identified trends over years and months however we couldn't glean an informative understanding of behaviour by hours and even minutes. The average prices in the past 5 years have increased drastically which

could be due to either population increase, an increase in costs for generators or a lack of competition in the market resulting in generators to bid their prices at higher prices. Although population would have an impact, we ultimately concluded that the prices were driven by a lack of competition. In terms of monthly trends, we found that generator prices tend to peak in February and are lowest in July in which prices begin to rise again after July all the way until February the next year. This translates to energy demand peaking in the summer months with lower demand during the winter months. Since there is a lack of competition in the market, we see generators increasing their prices in these months as they know that there is a higher chance of their energy being bought in peak demand irrespective of their prices. Using the R Shiny tool that we have created, we can observe these behaviours and identify the generators that are driving these trends.

Further issues

1. **Percentiles**: Due to the sheer size of the data, calculating the percentile information took many hours of computation to produce the information we needed. Initially, ten percentile values ranging from 10-100% of quantity generated was attempted but resulted in unreachable levels of code runtime required, resulting in the sacrifice of precision with the adoption of only 4 percentile values. Difficulties with both lack of coding experience and the large time complexity of calculating 4 percentile values using for loops, iterating approximately 100 times for each of the more than two million data points resulted in percentile data having only been completed about a week before the final report due date. As a result, much of the analysis was conducted on the raw data given as PRICEBAND and BANDAVAIL variables, resulting in a much less intuitive analysis, and weaker visualisations. Although, PCA analysis has been completed with percentile values for practical use in the Shiny app.
2. **Strange bids**: Some generators had entries that seemed like outliers as their prices were not sensible and seemed to be incorrect entries. However, we could not just remove these generators as these generators also had a majority of valid prices and entries and we had no way of coding a way to remove such strange behaviours due to the size of the data.
3. **Hierarchical and K-means clustering**: Due to the computational complexity of calculating the Euclidean distance for each observation in hierarchical and K-means clustering, we could not find the optimal number of clusters in hierarchical and K-means clustering. Usually, we can visually identify the optimal number of clusters using the Elbow method, average silhouette method and the cluster gap method (gap statistic) but the resulting vector for this calculation was over 15000GB which could not be run on our machine. Hence, we could not find significant insights on the behaviour of certain generators within each cluster.

Limitations

We have some issues relating to the R Shiny tool that we have produced.

1. **Range and domain**: When the PCA plots are initially loaded, the x-axis and y-axis limits are pre-set to a given range. Whilst you can double click the plot to resize the plot to more sensible bounds, it is an extra step that could have been fixed with further fixing of the code.
2. **Loading time**: Another issue is the time it takes to load some of these plots when you press update on the tool. Depending on the number of generators that are selected, these plots may take up to a few seconds to compute.
3. **Conditional Inputs**: We have made the drop downs such that they are conditional on the dropdowns you have selected above. This was done so that we can only select choices that are valid and will ensure that we always have a plot. However, because of this you have to wait a few seconds after inputting a choice in the dropdown before inputting the next selection as it takes times to compute. Another issue with the dropdowns is that you need to enter a selection for each dropdown in order to update the plot.
4. **Colour Palette Visualisation**: We have used the rainbow() colour palette in order to give each year a colour in the PCA plots. This has been very useful to us for our analysis however we were unable to give colours for other groups such as Months, Days and Hours. Another issue is that the rainbow() colour palette loops back to red by 2019 which is a similar shade to 2008 making it harder to discern which year it is.
5. **Frequency Visualisation**: Our PCA plot is lacking frequency information. Since all the points are the same size on the plot we cannot tell whether an area has just 1 observation or 100 observations stacked on top of one another. Although we have added a table output to print out the observations in a selected area, it is not a very neat way of representing that form of data.

7 Appendix

Codes

7.1 Initial Steps, data cleaning, assumptions, PCA Set Up, Year Colours

```
library(dplyr)
library(readxl)
library(lubridate) #for dates

# ------------------------------ Using biddayoffer and bidperofffer ---------------------- #
# We are importing the data provided to us, dropping irrelevant variables and merging the     #
# 2 tables into 1 by DUID and settlement date. We then partially clean the data for the       #
# variables Priceband and Bandavail. I also obtained an external source for region and        #
# fuel_type information. NOTE *the sql stuff is going to take some time to run.               #
# ------------------------------------------------------------------------------------------- #

### ------------------------------------------------ 7.1 Initial Steps ----------------------

# Reading in initial data provided from client
biddayoffer <- read.csv("~/Desktop/biddayoffer_fy2009-2019.csv")
bidperoffer <- read.csv("~/Desktop/bidperoffer_fy2009-2019_v2.csv")

# Filtering for ENERGY bidtype only
biddayoffer <- filter(biddayoffer, biddayoffer$BIDTYPE == "ENERGY")
bidperoffer <- filter(bidperoffer, bidperoffer$BIDTYPE == "ENERGY")

# Dropping irrelevant variables from biddayoffer and bidperoffer
biddayoffer_final <- select(biddayoffer, c("DUID", "SETTLEMENTDATE", "OFFERDATE", "VERSIONNO", "P
                                           "ENTRYTYPE", "PRICEBAND1", "PRICEBAND2", "PRICEBAND3",
                                           "PRICEBAND5", "PRICEBAND6", "PRICEBAND7", "PRICEBAND8"
                                           "PRICEBAND10"))

bidperoffer_final <- select(bidperoffer, c("DUID", "SETTLEMENTDATE", "INTERVAL_DATETIME", "OFFERD
                                           "PERIODID", "PASAAVAILABILITY", "MAXAVAIL", "BANDAVAIL
                                           "BANDAVAIL3", "BANDAVAIL4", "BANDAVAIL5", "BANDAVAIL6"
                                           "BANDAVAIL8", "BANDAVAIL9", "BANDAVAIL10"))

# Merging the two bid tables
bid_merged <- merge(biddayoffer_final, bidperoffer_final,
                    by.x = c("DUID", "SETTLEMENTDATE", "VERSIONNO", "OFFERDATE"),
                    by.y = c("DUID", "SETTLEMENTDATE", "VERSIONNO","OFFERDATE"))

# Re-formatting the variable INTERVAL_DATETIME and creating new time variables
bid_merged$INTERVAL_DATETIME <- as.POSIXct(strptime(bid_merged$INTERVAL_DATETIME, "%Y-%m-%d %H:%M
bid_merged$Year <- format(bid_merged$INTERVAL_DATETIME, format="%Y")
bid_merged$Month <- format(bid_merged$INTERVAL_DATETIME, format="%m") # 01, ..., 12
bid_merged$Month_name <- format(bid_merged$INTERVAL_DATETIME, format="%B") # January, ..., Decemb
bid_merged$Day_of_Month <- format(bid_merged$INTERVAL_DATETIME, format="%d") # 01, ..., 31
bid_merged$Day_of_Year <- format(bid_merged$INTERVAL_DATETIME, format="%j") # 001, ..., 366
bid_merged$Day_name <- format(bid_merged$INTERVAL_DATETIME, format="%A") # Monday, ... , Sunday
bid_merged$Week <- format(bid_merged$INTERVAL_DATETIME, format="%W") #0, ..., 53 (with Monday as
bid_merged$Hour <- format(bid_merged$INTERVAL_DATETIME, format="%H")
bid_merged$Minute <- format(bid_merged$INTERVAL_DATETIME, format="%M")

# Reading in AEMO Generator List obtained online
AEMO_Generator_List <- read_excel("Desktop/AEMO_Generator_List.xlsx", col_types = c("text", "text
                                                                                    "text", "text
```

```r
AEMO_Generator_List <- select(AEMO_Generator_List, c("DUID", "Region", "Fuel_Type"))

# Adding in 2 new variables Region and Fuel Type to bid_merged (note that there are some missing
bid_merged <- left_join(bid_merged, AEMO_Generator_List)
bid_merged <- distinct(bid_merged)

# Removing irrelevant tables from your global environment
rm(bidperoffer, biddayoffer, bidperoffer_final, biddayoffer_final, AEMO_Generator_List)

### ------------------------------------------------ 7.2 Data Cleaning ------------------------

# Checks to see if there is any missing data
if(sum(is.na(bid_merged)) != 0) {
  bid_merged <- bid_merged[!is.na(bid_merged$INTERVAL_DATETIME), ] # Removing NA rows
}

# making sure that we don't have any duplicate rows
bid_merged <- distinct(bid_merged)

### ------------------------------------------------ 7.3 Assumptions --------------------------


### ------------------------------------------------------ 7.4 PCA Setup ----------------------
p <- prcomp(bid_merged[,c(7:16)], scale=TRUE) # Priceband (7:16) PCA
b <- prcomp(bid_merged[,c(21:30)], scale=TRUE) # Bandavail (21:30) PCA

sp <- summary(p)
sb <- summary(b)

# MAKE SURE TO ONLY RUN THE NEXT 4 LINES ONCE
bid_merged_pb <- bid_merged[,c(1,7:16, 17, 40, 41)] # Selecting DUID (1) and pricebands (7:16) an
bid_merged_ba <- bid_merged[,c(1,21:30, 17, 40, 41)] # Selecting DUID (1) and bandavails (21:30)
bid_merged_pb <- cbind(bid_merged_pb, p$x[,1:2]) # Appending the PCA coordinates for PC1 (1) and
bid_merged_ba <- cbind(bid_merged_ba, b$x[,1:2]) # Appending the PCA coordinates for PC1 (1) and

# Percentiles Code
bid_merged_percentiles <- read_csv("Desktop/bid_merged_percentiles.csv")

bid_merged_percentiles <- bid_merged_percentiles[,c(3 ,19, 33:47)]

percentiles.pca <- prcomp(bid_merged_percentiles[,14:17], scale = TRUE)
summary_percentiles <- summary(percentiles.pca)

bid_merged_percentiles <- cbind(bid_merged_percentiles, percentiles.pca$x[,1:2])

### ------------------------------------------------ 7.5 Year Colours -------------------------
# We want to add colours to bid_merged_pb and bid_merged_ba by their years (2008 ~ 2019)
colVec <- rainbow(12) # a vector of 12 colours generated
pch.group <- c(rep(21, times = nrow(bid_merged)))

col.group <- ifelse(bid_merged$Year==2008, colVec[1],
             ifelse(bid_merged$Year==2009, colVec[2],
             ifelse(bid_merged$Year==2010, colVec[3],
             ifelse(bid_merged$Year==2011, colVec[4],
             ifelse(bid_merged$Year==2012, colVec[5],
```

```
            ifelse(bid_merged$Year==2013, colVec[6],
            ifelse(bid_merged$Year==2014, colVec[7],
            ifelse(bid_merged$Year==2015, colVec[8],
            ifelse(bid_merged$Year==2016, colVec[9],
            ifelse(bid_merged$Year==2017, colVec[10],
            ifelse(bid_merged$Year==2018, colVec[11],
            ifelse(bid_merged$Year==2019, colVec[12],
            "NO COL"
            ))))))))))
### ------------------------------------------------ 7.8 Diagnostic for PCA Plot  ---------------
# Create a new row that is the average of all observations. This point have PCA coordinates appro
test_row <- data.frame("TEST", "2000-01-01 00:00:00", 1, "2000-01-01 00:00:00", "TEST", "TEST",
                       mean(bid_merged$PRICEBAND1), mean(bid_merged$PRICEBAND2), mean(bid_merged$
                       mean(bid_merged$PRICEBAND4), mean(bid_merged$PRICEBAND5), mean(bid_merged$
                       mean(bid_merged$PRICEBAND7), mean(bid_merged$PRICEBAND8), mean(bid_merged$
                       mean(bid_merged$PRICEBAND10),
                       "2000-01-01 00:00:00", 1, 1, 1,
                       mean(bid_merged$BANDAVAIL1), mean(bid_merged$BANDAVAIL2), mean(bid_merged$
                       mean(bid_merged$BANDAVAIL4), mean(bid_merged$BANDAVAIL5), mean(bid_merged$
                       mean(bid_merged$BANDAVAIL7), mean(bid_merged$BANDAVAIL8), mean(bid_merged$
                       mean(bid_merged$BANDAVAIL10),
                       2000, 1, "TEST", 1, 1, "TEST", 1, 1, 1, "TEST", "TEST")

names(test_row) <- c("DUID", "SETTLEMENTDATE", "VERSIONNO", "OFFERDATE", "REBIDEXPLANATION", "ENT
                     "PRICEBAND1", "PRICEBAND2", "PRICEBAND3", "PRICEBAND4", "PRICEBAND5", "PRICE
                     "PRICEBAND7", "PRICEBAND8","PRICEBAND9", "PRICEBAND10", "INTERVAL_DATETIME",
                     "PASAAVAILABILITY", "MAXAVAIL", "BANDAVAIL1", "BANDAVAIL2", "BANDAVAIL3", "B
                     "BANDAVAIL5", "BANDAVAIL6", "BANDAVAIL7", "BANDAVAIL8","BANDAVAIL9", "BANDAV
                     "Year", "Month", "Month_name", "Day_of_Month", "Day_of_Year", "Day_name", "W
                     "Minute", "Region", "Fuel_Type")

test_df <- rbind(bid_merged, test_row) #Using rbind() function to insert above observation

p_test <- prcomp(test_df[,c(7:16)], scale=TRUE)
bid_merged_pb_test <- test_df[,c(1, 7:16, 17, 40, 41)] # Selecting DUID (1) and pricebands (7:16)
bid_merged_pb_test <- cbind(bid_merged_pb, p_test$x[,1:2]) # Appending the PCA coordinates for PC
tail(bid_merged_pb_test, 1) # This row should have PC1 ~ 0 and PC2 ~ 0
rm(p_test)

b_test <- prcomp(test_df[,c(21:30)], scale=TRUE)
bid_merged_ba_test <- test_df[,c(1, 21:30, 17, 40, 41)] # Selecting DUID (1) and bandavails (21:3
bid_merged_ba_test <- cbind(bid_merged_ba, b_test$x[,1:2]) # Appending the PCA coordinates for PC
tail(bid_merged_ba_test, 1) # This row should have PC1 ~ 0 and PC2 ~ 0
rm(b_test)

rm(test_df, test_row, bid_merged_ba_test, bid_merged_pb_test)
```

7.2 Initial Analysis Pricebands, Initial Analysis BandAvails, Diagnostic for PCA plot, Scree Plots, Trend Analysis

```
library(dplyr)
library(ggplot2)
library(reshape)
library(gridExtra)

### ------------------------------------ 7.6 Initial Analysis Pricebands ---------------------
```

```r
# Priceband Boxplot Fig 4.1.1
boxplot(bid_merged_pb$PRICEBAND1, bid_merged_pb$PRICEBAND2, bid_merged_pb$PRICEBAND3, bid_merged_
        bid_merged_pb$PRICEBAND5, bid_merged_pb$PRICEBAND6, bid_merged_pb$PRICEBAND7, bid_merged_
        bid_merged_pb$PRICEBAND9, bid_merged_pb$PRICEBAND10,
        names = c("PB1", "PB2", "PB3", "PB4", "PB5", "PB6", "PB7", "PB8", "PB9", "PB10"),
        main = "Price Distribution", xlab = "Pricebands", ylab = "Price ($AUD)")

# Priceband Boxplot (without outliers + subset 1) Fig 4.1.2
par(mfrow= c(2,2))
boxplot(bid_merged_pb$PRICEBAND1, main = "Price Distribution", xlab = "Priceband1", ylab = "Price
boxplot(bid_merged_pb$PRICEBAND8, main = "Price Distribution", xlab = "Priceband8", ylab = "Price
boxplot(bid_merged_pb$PRICEBAND9, main = "Price Distribution", xlab = "Priceband9", ylab = "Price
boxplot(bid_merged_pb$PRICEBAND10, main = "Price Distribution", xlab = "Priceband10", ylab = "Pri


# Priceband Boxplot (without outliers + subset 2) Fig 4.1.3
par(mfrow= c(1,1))
boxplot(bid_merged_pb$PRICEBAND2, bid_merged_pb$PRICEBAND3, bid_merged_pb$PRICEBAND4,
        bid_merged_pb$PRICEBAND5, bid_merged_pb$PRICEBAND6, bid_merged_pb$PRICEBAND7,
        names = c("PB2", "PB3", "PB4", "PB5", "PB6", "PB7"),
        main = "Price Distribution", xlab = "Pricebands", ylab = "Price ($AUD)", outline = FALSE)

# Summaries of priceband quartiles Out 4.1.1
summary(bid_merged_pb$PRICEBAND1)
summary(bid_merged_pb$PRICEBAND2)
summary(bid_merged_pb$PRICEBAND3)
summary(bid_merged_pb$PRICEBAND4)
summary(bid_merged_pb$PRICEBAND5)
summary(bid_merged_pb$PRICEBAND6)
summary(bid_merged_pb$PRICEBAND7)
summary(bid_merged_pb$PRICEBAND8)
summary(bid_merged_pb$PRICEBAND9)
summary(bid_merged_pb$PRICEBAND10)

# find the upper inner fence value for each priceband
maxval <- apply(bid_merged_pb[,2:11], MARGIN = 2,
                function(x) {quantile(x, probs = 0.75) + (IQR(x) * 1.5)})
print(maxval)

# find number of outliers in each priceband (above upper inner fence)
maxvals <- data.frame("PB1" = 0, "PB2" = 0, "PB3" = 0, "PB4" = 0, "PB5" = 0,
                      "PB6" = 0, "PB7" = 0, "PB8" = 0, "PB9" = 0, "PB10" = 0)
maxvals[,1] <- sum(bid_merged_pb$PRICEBAND1 > maxval[1])
maxvals[,2] <- sum(bid_merged_pb$PRICEBAND2 > maxval[2])
maxvals[,3] <- sum(bid_merged_pb$PRICEBAND3 > maxval[3])
maxvals[,4] <- sum(bid_merged_pb$PRICEBAND4 > maxval[4])
maxvals[,5] <- sum(bid_merged_pb$PRICEBAND5 > maxval[5])
maxvals[,6] <- sum(bid_merged_pb$PRICEBAND6 > maxval[6])
maxvals[,7] <- sum(bid_merged_pb$PRICEBAND7 > maxval[7])
maxvals[,8] <- sum(bid_merged_pb$PRICEBAND8 > maxval[8])
maxvals[,9] <- sum(bid_merged_pb$PRICEBAND9 > maxval[9])
maxvals[,10] <- sum(bid_merged_pb$PRICEBAND10 > maxval[10])

# find the lower inner fence value for each priceband
minval <- apply(bid_merged_pb[,2:11], MARGIN = 2,
```

```
                function(x) {quantile(x, probs = 0.25) - (IQR(x) * 1.5)})
print(minval)

# find number of outliers in each priceband (below lower inner fence)
minvals <- data.frame("PB1" = 0, "PB2" = 0, "PB3" = 0, "PB4" = 0, "PB5" = 0,
                      "PB6" = 0, "PB7" = 0, "PB8" = 0, "PB9" = 0, "PB10" = 0)
minvals[,1] <- sum(bid_merged_pb$PRICEBAND1 < minval[1])
minvals[,2] <- sum(bid_merged_pb$PRICEBAND2 < minval[2])
minvals[,3] <- sum(bid_merged_pb$PRICEBAND3 < minval[3])
minvals[,4] <- sum(bid_merged_pb$PRICEBAND4 < minval[4])
minvals[,5] <- sum(bid_merged_pb$PRICEBAND5 < minval[5])
minvals[,6] <- sum(bid_merged_pb$PRICEBAND6 < minval[6])
minvals[,7] <- sum(bid_merged_pb$PRICEBAND7 < minval[7])
minvals[,8] <- sum(bid_merged_pb$PRICEBAND8 < minval[8])
minvals[,9] <- sum(bid_merged_pb$PRICEBAND9 < minval[9])
minvals[,10] <- sum(bid_merged_pb$PRICEBAND10 < minval[10])

outliers <- data.frame(cbind(maxval, t(maxvals), minval, t(minvals)))
colnames(outliers) <- c("Upper inner fence value", "Number outliers Upper",
                        "Lower inner fence value", "Number outliers Lower")
print(outliers)
write.csv(outliers,"outliers_table_pb.csv")

# Find the Variances of each band
var <- data.frame("PB1" = 0, "PB2" = 0, "PB3" = 0, "PB4" = 0, "PB5" = 0,
                  "PB6" = 0, "PB7" = 0, "PB8" = 0, "PB9" = 0, "PB10" = 0)
var[,1] <- var(bid_merged_pb$PRICEBAND1)
var[,2] <- var(bid_merged_pb$PRICEBAND2)
var[,3] <- var(bid_merged_pb$PRICEBAND3)
var[,4] <- var(bid_merged_pb$PRICEBAND4)
var[,5] <- var(bid_merged_pb$PRICEBAND5)
var[,6] <- var(bid_merged_pb$PRICEBAND6)
var[,7] <- var(bid_merged_pb$PRICEBAND7)
var[,8] <- var(bid_merged_pb$PRICEBAND8)
var[,9] <- var(bid_merged_pb$PRICEBAND9)
var[,10] <- var(bid_merged_pb$PRICEBAND10)

# Histograms of each priceband Fig 4.1.4
grid.arrange(ggplot(bid_merged_pb, aes(bid_merged_pb$PRICEBAND1)) +
             geom_histogram(col = "black", fill = "light blue") +
             theme(axis.text = element_text(size = 11),
                   axis.title = element_text(size = 13, face = "bold")) +
             labs(x = "PRICEBAND1"),

           ggplot(bid_merged_pb, aes(bid_merged_pb$PRICEBAND2)) +
             geom_histogram(col = "black", fill = "light blue") +
             theme(axis.text = element_text(size = 11),
                   axis.title = element_text(size = 13, face = "bold")) +
             labs(x = "PRICEBAND2"),

           ggplot(bid_merged_pb, aes(bid_merged_pb$PRICEBAND3)) +
             geom_histogram(col = "black", fill = "light blue") +
             theme(axis.text = element_text(size = 11),
                   axis.title = element_text(size = 13, face = "bold")) +
             labs(x = "PRICEBAND3"),
```

```
          ggplot(bid_merged_pb, aes(bid_merged_pb$PRICEBAND4)) +
            geom_histogram(col = "black", fill = "light blue") +
            theme(axis.text = element_text(size = 11),
                  axis.title = element_text(size = 13, face = "bold")) +
            labs(x = "PRICEBAND4"),

          ggplot(bid_merged_pb, aes(bid_merged_pb$PRICEBAND5)) +
            geom_histogram(col = "black", fill = "light blue") +
            theme(axis.text = element_text(size = 11),
                  axis.title = element_text(size = 13, face = "bold")) +
            labs(x = "PRICEBAND5"),

          ggplot(bid_merged_pb, aes(bid_merged_pb$PRICEBAND6)) +
            geom_histogram(col = "black", fill = "light blue") +
            theme(axis.text = element_text(size = 11),
                  axis.title = element_text(size = 13, face = "bold")) +
            labs(x = "PRICEBAND6"),

          ggplot(bid_merged_pb, aes(bid_merged_pb$PRICEBAND7)) +
            geom_histogram(col = "black", fill = "light blue") +
            theme(axis.text = element_text(size = 11),
                  axis.title = element_text(size = 13, face = "bold")) +
            labs(x = "PRICEBAND7"),

          ggplot(bid_merged_pb, aes(bid_merged_pb$PRICEBAND8)) +
            geom_histogram(col = "black", fill = "light blue") +
            theme(axis.text = element_text(size = 11),
                  axis.title = element_text(size = 13, face = "bold")) +
            labs(x = "PRICEBAND8"),

          ggplot(bid_merged_pb, aes(bid_merged_pb$PRICEBAND9)) +
            geom_histogram(col = "black", fill = "light blue") +
            theme(axis.text = element_text(size = 11),
                  axis.title = element_text(size = 13, face = "bold")) +
            labs(x = "PRICEBAND9"),

          ggplot(bid_merged_pb, aes(bid_merged_pb$PRICEBAND10)) +
            geom_histogram(col = "black", fill = "light blue") +
            theme(axis.text = element_text(size = 11),
                  axis.title = element_text(size = 13, face = "bold")) +
            labs(x = "PRICEBAND10"),
  ncol = 5, nrow = 2)

### ------------------------------------- 7.7 Initial Analysis Bandavails ------------------

# Band Available Boxplot Fig 4.1.5
boxplot(bid_merged_ba$BANDAVAIL1, bid_merged_ba$BANDAVAIL2, bid_merged_ba$BANDAVAIL3, bid_merged_
        bid_merged_ba$BANDAVAIL5, bid_merged_ba$BANDAVAIL6, bid_merged_ba$BANDAVAIL7, bid_merged_
        bid_merged_ba$BANDAVAIL9, bid_merged_ba$BANDAVAIL10,
        names = c("BA1", "BA2", "BA3", "BA4", "BA5", "BA6", "BA7", "BA8", "BA9", "BA10"),
        main = "Quantity Distribution", xlab = "BandAvails", ylab = "Quantity (mwh)")

# BANDAVAIL Boxplot (without outliers + subset 1) Fig 4.1.6
par(mfrow= c(2,2))
boxplot(bid_merged_ba$BANDAVAIL1, main = "Quantity Distribution", xlab = "BANDAVAIL1", ylab = "Qu
```

```r
boxplot(bid_merged_ba$BANDAVAIL8, main = "Quantity Distribution", xlab = "BANDAVAIL8", ylab = "Qu
boxplot(bid_merged_ba$BANDAVAIL9, main = "Quantity Distribution", xlab = "BANDAVAIL9", ylab = "Qu
boxplot(bid_merged_ba$BANDAVAIL10, main = "Quantity Distribution", xlab = "BANDAVAIL10", ylab = '


# BANDAVAIL Boxplot (without outliers + subset 2) Fig 4.1.7
par(mfrow= c(1,1))
boxplot(bid_merged_ba$BANDAVAIL2, bid_merged_ba$BANDAVAIL3, bid_merged_ba$BANDAVAIL4,
        bid_merged_ba$BANDAVAIL5, bid_merged_ba$BANDAVAIL6, bid_merged_ba$BANDAVAIL7,
        names = c("BA2", "BA3", "BA4", "BA5", "BA6", "BA7"),
        main = "Quantity Distribution", xlab = "BANDAVAILs", ylab = "Quantity (mwh)", outline = F

# Summaries of BANDAVAIL quartiles Out 4.1.5
summary(bid_merged_ba$BANDAVAIL1)
summary(bid_merged_ba$BANDAVAIL2)
summary(bid_merged_ba$BANDAVAIL3)
summary(bid_merged_ba$BANDAVAIL4)
summary(bid_merged_ba$BANDAVAIL5)
summary(bid_merged_ba$BANDAVAIL6)
summary(bid_merged_ba$BANDAVAIL7)
summary(bid_merged_ba$BANDAVAIL8)
summary(bid_merged_ba$BANDAVAIL9)
summary(bid_merged_ba$BANDAVAIL10)

# find the upper inner fence value for each BANDAVAIL
maxval <- apply(bid_merged_ba[,2:11], MARGIN = 2,
                function(x) {quantile(x, probs = 0.75) + (IQR(x) * 1.5)})
print(maxval)

# find number of outliers in each BANDAVAIL (above upper inner fence)
maxvals <- data.frame("BA1" = 0, "BA2" = 0, "BA3" = 0, "BA4" = 0, "BA5" = 0,
                      "BA6" = 0, "BA7" = 0, "BA8" = 0, "BA9" = 0, "BA10" = 0)
maxvals[,1] <- sum(bid_merged_ba$BANDAVAIL1 > maxval[1])
maxvals[,2] <- sum(bid_merged_ba$BANDAVAIL2 > maxval[2])
maxvals[,3] <- sum(bid_merged_ba$BANDAVAIL3 > maxval[3])
maxvals[,4] <- sum(bid_merged_ba$BANDAVAIL4 > maxval[4])
maxvals[,5] <- sum(bid_merged_ba$BANDAVAIL5 > maxval[5])
maxvals[,6] <- sum(bid_merged_ba$BANDAVAIL6 > maxval[6])
maxvals[,7] <- sum(bid_merged_ba$BANDAVAIL7 > maxval[7])
maxvals[,8] <- sum(bid_merged_ba$BANDAVAIL8 > maxval[8])
maxvals[,9] <- sum(bid_merged_ba$BANDAVAIL9 > maxval[9])
maxvals[,10] <- sum(bid_merged_ba$BANDAVAIL10 > maxval[10])

# find the lower inner fence value for each BANDAVAIL
minval <- apply(bid_merged_ba[,2:11], MARGIN = 2,
                function(x) {quantile(x, probs = 0.25) - (IQR(x) * 1.5)})
print(minval)

# find number of outliers in each BANDAVAIL (below lower inner fence)
minvals <- data.frame("BA1" = 0, "BA2" = 0, "BA3" = 0, "BA4" = 0, "BA5" = 0,
                      "BA6" = 0, "BA7" = 0, "BA8" = 0, "BA9" = 0, "BA10" = 0)
minvals[,1] <- sum(bid_merged_ba$BANDAVAIL1 < minval[1])
minvals[,2] <- sum(bid_merged_ba$BANDAVAIL2 < minval[2])
minvals[,3] <- sum(bid_merged_ba$BANDAVAIL3 < minval[3])
minvals[,4] <- sum(bid_merged_ba$BANDAVAIL4 < minval[4])
```

```
minvals[,5] <- sum(bid_merged_ba$BANDAVAIL5 < minval[5])
minvals[,6] <- sum(bid_merged_ba$BANDAVAIL6 < minval[6])
minvals[,7] <- sum(bid_merged_ba$BANDAVAIL7 < minval[7])
minvals[,8] <- sum(bid_merged_ba$BANDAVAIL8 < minval[8])
minvals[,9] <- sum(bid_merged_ba$BANDAVAIL9 < minval[9])
minvals[,10] <- sum(bid_merged_ba$BANDAVAIL10 < minval[10])

outliers <- data.frame(cbind(maxval, t(maxvals), minval, t(minvals)))
colnames(outliers) <- c("Upper inner fence value", "Number outliers Upper",
                        "Lower inner fence value", "Number outliers Lower")
print(outliers)
write.csv(outliers,"outliers_table_BA.csv")

# Find the Variances of each band
var <- data.frame("BA1" = 0, "BA2" = 0, "BA3" = 0, "BA4" = 0, "BA5" = 0,
                  "BA6" = 0, "BA7" = 0, "BA8" = 0, "BA9" = 0, "BA10" = 0)
var[,1] <- var(bid_merged_ba$BANDAVAIL1)
var[,2] <- var(bid_merged_ba$BANDAVAIL2)
var[,3] <- var(bid_merged_ba$BANDAVAIL3)
var[,4] <- var(bid_merged_ba$BANDAVAIL4)
var[,5] <- var(bid_merged_ba$BANDAVAIL5)
var[,6] <- var(bid_merged_ba$BANDAVAIL6)
var[,7] <- var(bid_merged_ba$BANDAVAIL7)
var[,8] <- var(bid_merged_ba$BANDAVAIL8)
var[,9] <- var(bid_merged_ba$BANDAVAIL9)
var[,10] <- var(bid_merged_ba$BANDAVAIL10)

# Histograms of each BANDAVAIL Fig 4.1.8
grid.arrange(ggplot(bid_merged_ba, aes(bid_merged_ba$BANDAVAIL1)) +
               geom_histogram(col = "black", fill = "light blue") +
               theme(axis.text = element_text(size = 11),
                     axis.title = element_text(size = 13, face = "bold")) +
               labs(x = "BANDAVAIL1"),

             ggplot(bid_merged_ba, aes(bid_merged_ba$BANDAVAIL2)) +
               geom_histogram(col = "black", fill = "light blue") +
               theme(axis.text = element_text(size = 11),
                     axis.title = element_text(size = 13, face = "bold")) +
               labs(x = "BANDAVAIL2"),

             ggplot(bid_merged_ba, aes(bid_merged_ba$BANDAVAIL3)) +
               geom_histogram(col = "black", fill = "light blue") +
               theme(axis.text = element_text(size = 11),
                     axis.title = element_text(size = 13, face = "bold")) +
               labs(x = "BANDAVAIL3"),

             ggplot(bid_merged_ba, aes(bid_merged_ba$BANDAVAIL4)) +
               geom_histogram(col = "black", fill = "light blue") +
               theme(axis.text = element_text(size = 11),
                     axis.title = element_text(size = 13, face = "bold")) +
               labs(x = "BANDAVAIL4"),

             ggplot(bid_merged_ba, aes(bid_merged_ba$BANDAVAIL5)) +
               geom_histogram(col = "black", fill = "light blue") +
               theme(axis.text = element_text(size = 11),
```

```
                  axis.title = element_text(size = 13, face = "bold")) +
            labs(x = "BANDAVAIL5"),

          ggplot(bid_merged_ba, aes(bid_merged_ba$BANDAVAIL6)) +
            geom_histogram(col = "black", fill = "light blue") +
            theme(axis.text = element_text(size = 11),
                  axis.title = element_text(size = 13, face = "bold")) +
            labs(x = "BANDAVAIL6"),

          ggplot(bid_merged_ba, aes(bid_merged_ba$BANDAVAIL7)) +
            geom_histogram(col = "black", fill = "light blue") +
            theme(axis.text = element_text(size = 11),
                  axis.title = element_text(size = 13, face = "bold")) +
            labs(x = "BANDAVAIL7"),

          ggplot(bid_merged_ba, aes(bid_merged_ba$BANDAVAIL8)) +
            geom_histogram(col = "black", fill = "light blue") +
            theme(axis.text = element_text(size = 11),
                  axis.title = element_text(size = 13, face = "bold")) +
            labs(x = "BANDAVAIL8"),

          ggplot(bid_merged_ba, aes(bid_merged_ba$BANDAVAIL9)) +
            geom_histogram(col = "black", fill = "light blue") +
            theme(axis.text = element_text(size = 11),
                  axis.title = element_text(size = 13, face = "bold")) +
            labs(x = "BANDAVAIL9"),

          ggplot(bid_merged_ba, aes(bid_merged_ba$BANDAVAIL10)) +
            geom_histogram(col = "black", fill = "light blue") +
            theme(axis.text = element_text(size = 11),
                  axis.title = element_text(size = 13, face = "bold")) +
            labs(x = "BANDAVAIL10"),
          ncol = 5, nrow = 2)

# Histograms but domain 0 to 500

grid.arrange(ggplot(bid_merged_ba, aes(bid_merged_ba$BANDAVAIL1)) +
            geom_histogram(col = "black", fill = "light blue") +
            theme(axis.text = element_text(size = 11),
                  axis.title = element_text(size = 13, face = "bold")) +
            labs(x = "BANDAVAIL1") +
            xlim(0,500),

          ggplot(bid_merged_ba, aes(bid_merged_ba$BANDAVAIL2)) +
            geom_histogram(col = "black", fill = "light blue") +
            theme(axis.text = element_text(size = 11),
                  axis.title = element_text(size = 13, face = "bold")) +
            labs(x = "BANDAVAIL2") +
            xlim(0,500),

          ggplot(bid_merged_ba, aes(bid_merged_ba$BANDAVAIL3)) +
            geom_histogram(col = "black", fill = "light blue") +
            theme(axis.text = element_text(size = 11),
                  axis.title = element_text(size = 13, face = "bold")) +
            labs(x = "BANDAVAIL3") +
```

```
  xlim(0,300),

ggplot(bid_merged_ba, aes(bid_merged_ba$BANDAVAIL4)) +
  geom_histogram(col = "black", fill = "light blue") +
  theme(axis.text = element_text(size = 11),
        axis.title = element_text(size = 13, face = "bold")) +
  labs(x = "BANDAVAIL4") +
  xlim(0,200),

ggplot(bid_merged_ba, aes(bid_merged_ba$BANDAVAIL5)) +
  geom_histogram(col = "black", fill = "light blue") +
  theme(axis.text = element_text(size = 11),
        axis.title = element_text(size = 13, face = "bold")) +
  labs(x = "BANDAVAIL5") +
  xlim(0,200),

ggplot(bid_merged_ba, aes(bid_merged_ba$BANDAVAIL6)) +
  geom_histogram(col = "black", fill = "light blue") +
  theme(axis.text = element_text(size = 11),
        axis.title = element_text(size = 13, face = "bold")) +
  labs(x = "BANDAVAIL6") +
  xlim(0,200),

ggplot(bid_merged_ba, aes(bid_merged_ba$BANDAVAIL7)) +
  geom_histogram(col = "black", fill = "light blue") +
  theme(axis.text = element_text(size = 11),
        axis.title = element_text(size = 13, face = "bold")) +
  labs(x = "BANDAVAIL7") +
  xlim(0,200),

ggplot(bid_merged_ba, aes(bid_merged_ba$BANDAVAIL8)) +
  geom_histogram(col = "black", fill = "light blue") +
  theme(axis.text = element_text(size = 11),
        axis.title = element_text(size = 13, face = "bold")) +
  labs(x = "BANDAVAIL8") +
  xlim(0,200),

ggplot(bid_merged_ba, aes(bid_merged_ba$BANDAVAIL9)) +
  geom_histogram(col = "black", fill = "light blue") +
  theme(axis.text = element_text(size = 11),
        axis.title = element_text(size = 13, face = "bold")) +
  labs(x = "BANDAVAIL9") +
  xlim(0,200),

ggplot(bid_merged_ba, aes(bid_merged_ba$BANDAVAIL10)) +
  geom_histogram(col = "black", fill = "light blue") +
  theme(axis.text = element_text(size = 11),
        axis.title = element_text(size = 13, face = "bold")) +
  labs(x = "BANDAVAIL10") +
  xlim(0,500),

ncol = 5, nrow = 2)


### ------------------------------------ 7.9 Scree Plots ----------------------------------
```

```r
plot(p, type = "lines", main = "Pricebands")
sp # summary of p
plot(b, type = "lines", main = "BandAvails")
sb # summary of b

### ------------------------------------- 7.10 Loadings Plots ------------------------------
# Pricebands

par(mfrow = c(2,1))

par(mar=c(8,3,2,1)) # Set margins
n.pc1_pb <- ifelse(p$rotation[,1] > 0, yes = -0.01, no = p$rotation[,1] - 0.01)
c.pc1_pb <- ifelse(p$rotation[,1] > 0, yes = "green2", no = "red2")
pc1_pb <- barplot(p$rotation[,1], main="PC 1 Loadings Plot Pricebands",
        las = 2, col = c.pc1_pb, axisnames = FALSE)
        abline(h = 0) # Add horizontal line
text(x = pc1_pb, y = n.pc1_pb, labels = names(p$rotation[,1]), adj = 1, srt = 90, xpd = TRUE) # A

rm(pc1_pb, n.pc1_pb, c.pc1_pb)

par(mar=c(8,3,2,1)) # Set margins
n.pc2_pb <- ifelse(p$rotation[,2] > 0, yes = -0.01, no = p$rotation[,2] - 0.01)
c.pc2_pb <- ifelse(p$rotation[,2] > 0, yes = "green2", no = "red2")
pc2_pb <- barplot(p$rotation[,2], main="PC 2 Loadings Plot Pricebands",
                  las = 2, col = c.pc2_pb, axisnames = FALSE)
abline(h = 0) # Add horizontal line
text(x = pc2_pb, y = n.pc2_pb, labels = names(p$rotation[,2]), adj = 1, srt = 90, xpd = TRUE) # A

rm(pc2_pb, n.pc2_pb, c.pc2_pb)

# Band Avail
par(mar=c(8,3,2,1)) # Set margins
n.pc1_ba <- ifelse(b$rotation[,1] > 0, yes = -0.01, no = b$rotation[,1] - 0.01)
c.pc1_ba <- ifelse(b$rotation[,1] > 0, yes = "green2", no = "red2")
pc1_ba <- barplot(b$rotation[,1], main = "PC 1 Loadings Plot BandAvail",
                  las = 2, col = c.pc1_ba, axisnames = FALSE)
abline(h=0) # Add horizontal line
text(x = pc1_ba, y = n.pc1_ba, labels = names(b$rotation[,1]), adj = 1, srt = 90, xpd = TRUE) # A

rm(pc1_ba, n.pc1_ba, c.pc1_ba)

par(mar=c(8,3,2,1)) # Set margins
n.pc2_ba <- ifelse(b$rotation[,2] > 0, yes = -0.01, no = b$rotation[,2] - 0.01)
c.pc2_ba <- ifelse(b$rotation[,2] > 0, yes = "green2", no = "red2")
pc2_ba <- barplot(b$rotation[,2], main="PC 2 Loadings Plot BandAvail",
                  las = 2, col = c.pc2_ba, axisnames = FALSE)
abline(h = 0) # Add horizontal line
text(x = pc2_ba, y = n.pc2_ba, labels = names(b$rotation[,2]), adj = 1, srt = 90, xpd = TRUE) # A

rm(pc2_ba, n.pc2_ba, c.pc2_ba)
```

7.3 K-means

```r
library(tidyverse)  # data manipulation
library(cluster)    # clustering algorithms
library(factoextra) # clustering algorithms & visualization
```

```r
library(dendextend) # for comparing two dendrograms
library(dplyr)      # Working with data frames
library(gridExtra)  # For displaying graph

data_pb <- read.csv("bid_merged_pb_FIN.csv", header = TRUE)
data_ba <- read.csv("bid_merged_ba_FIN.csv", header = TRUE)

set.seed(123)

# select the pc1 and pc2 for price bands and band availability
pca_pb <- select(data_pb, PC1, PC2)
pca_ba <- select(data_ba, PC1, PC2)

pca_ba2 <- select(data_ba, BANDAVAIL1,BANDAVAIL2, BANDAVAIL3, BANDAVAIL4, BANDAVAIL5,
                  BANDAVAIL6, BANDAVAIL7, BANDAVAIL8, BANDAVAIL9, BANDAVAIL10)
pca_ba2_scale <- scale(pca_ba2)
head(pca_ba2_scale)
summary(pca_ba2_scale)

k2b <- kmeans(pca_ba2_scale, centers = 2, nstart = 25)
k3b <- kmeans(pca_ba2_scale, centers = 3, nstart = 25)
k4b <- kmeans(pca_ba2_scale, centers = 4, nstart = 25)
k5b <- kmeans(pca_ba2_scale, centers = 5, nstart = 25)

p2b <- fviz_cluster(k2b, geom = "point", data = pca_ba2_scale) + ggtitle("k = 2")
p3b <- fviz_cluster(k3b, geom = "point", data = pca_ba2_scale) + ggtitle("k = 3")
p4b <- fviz_cluster(k4b, geom = "point", data = pca_ba2_scale) + ggtitle("k = 4")
p5b <- fviz_cluster(k5b, geom = "point", data = pca_ba2_scale) + ggtitle("k = 5")

library(gridExtra)
grid.arrange(p2b, p3b, p4b, p5b, nrow = 4)

# Price band kmeans
pca_pb2 <- select(data_pb, PRICEBAND1,PRICEBAND2, PRICEBAND3, PRICEBAND4, PRICEBAND5,
                   PRICEBAND6, PRICEBAND7, PRICEBAND8, PRICEBAND9, PRICEBAND10)
pca_pb2_scale <- scale(pca_pb2)
head(pca_pb2_scale)
summary(pca_pb2_scale)

k2c <- kmeans(pca_pb2_scale, centers = 2, nstart = 25)
k3c <- kmeans(pca_pb2_scale, centers = 3, nstart = 25)
k4c <- kmeans(pca_pb2_scale, centers = 4, nstart = 25)
k5c <- kmeans(pca_pb2_scale, centers = 5, nstart = 25)

p2c <- fviz_cluster(k2c, geom = "point", data = pca_pb2_scale) + ggtitle("k = 2")
p3c <- fviz_cluster(k3c, geom = "point", data = pca_pb2_scale) + ggtitle("k = 3")
p4c <- fviz_cluster(k4c, geom = "point", data = pca_pb2_scale) + ggtitle("k = 4")
p5c <- fviz_cluster(k5c, geom = "point", data = pca_pb2_scale) + ggtitle("k = 5")

grid.arrange(p2c, p3c, p4c, p5c, nrow = 4)
```

Outputs

Fig A3.2 - Dataset descriptions

	biddayoffer_fy2009-2019.csv	*bidperoffer_fy2009-2019_v2.csv*
Format	csv	csv
Size	739.9 megabytes 27 variables 3,761,813 observations	2.64 gigabytes 29 variables 14,218,111 observations
Variables	DUID BIDTYPE SETTLEMENTDATE OFFERDATE VERSIONNO PARTICIPANTID DAILYENERGYCONSTRAINT REBIDEXPLANATION PRICEBAND1 PRICEBAND2 PRICEBAND3 PRICEBAND4 PRICEBAND5 PRICEBAND6 PRICEBAND7 PRICEBAND8 PRICEBAND9 PRICEBAND10 MINIMUMLOAD T1 T2 T3 T4 NORMALSTATUS LASTCHANGED MR_FACTOR ENTRYTYPE	DUID BIDTYPE SETTLEMENTDATE BIDSETTLEMENTDATE OFFERDATE VERSIONNO PERIODID MAXAVAIL FIXEDLOAD ROCUP ROCDOWN ENABLEMENTMIN ENABLEMENTMAX LOWBREAKPOINT HIGHBREAKPOINT BANDAVAIL1 BANDAVAIL2 BANDAVAIL3 BANDAVAIL4 BANDAVAIL5 BANDAVAIL6 BANDAVAIL7 BANDAVAIL8 BANDAVAIL9 BANDAVAIL10 LASTCHANGED PASAAVAILABILITY INTERVAL_DATETIME MR_CAPACITY
Data Types	Numerical Text Datetime	Numerical Text Datetime

Fig A4.1.1 – Priceband Distributions

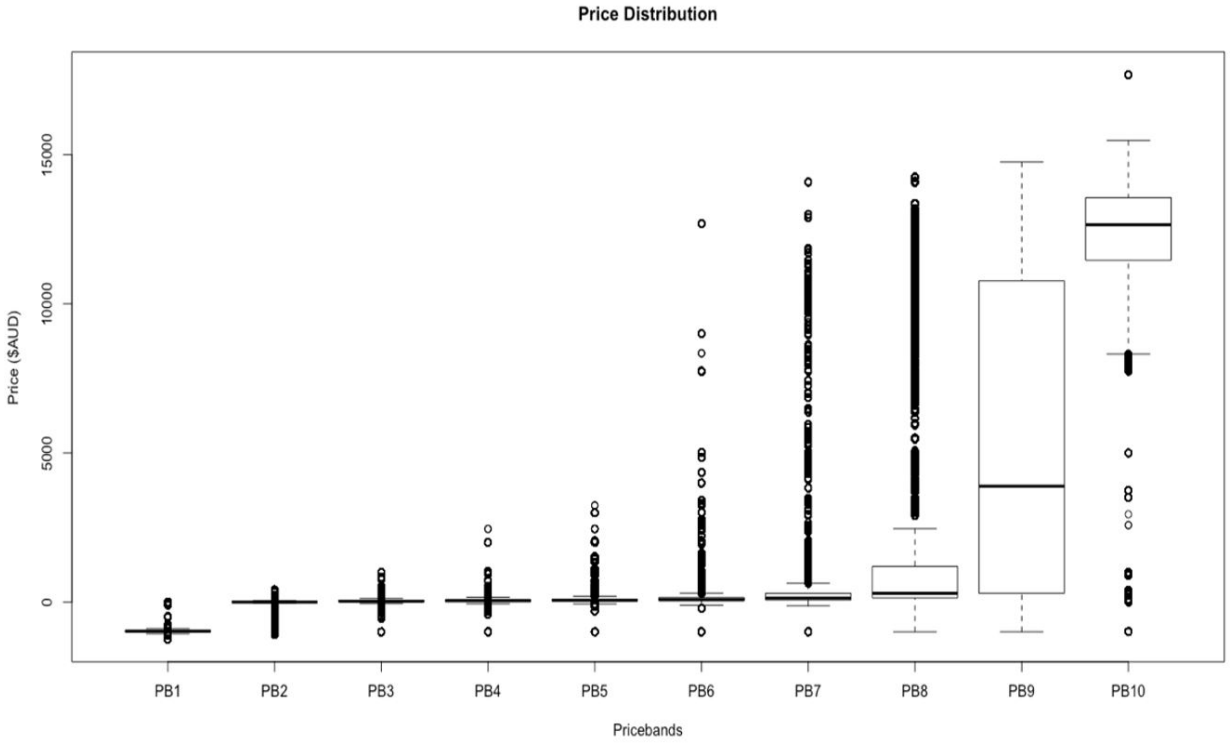


Fig A4.1.2 – Priceband Distributions (Set 1)

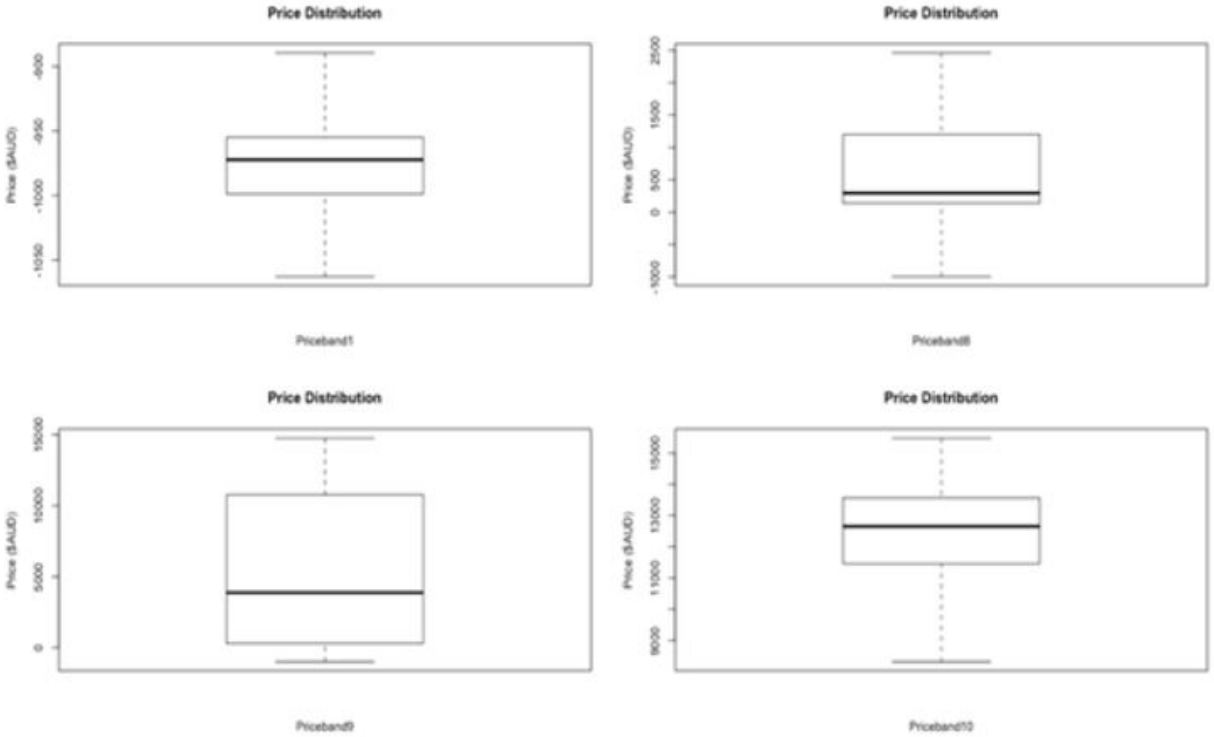
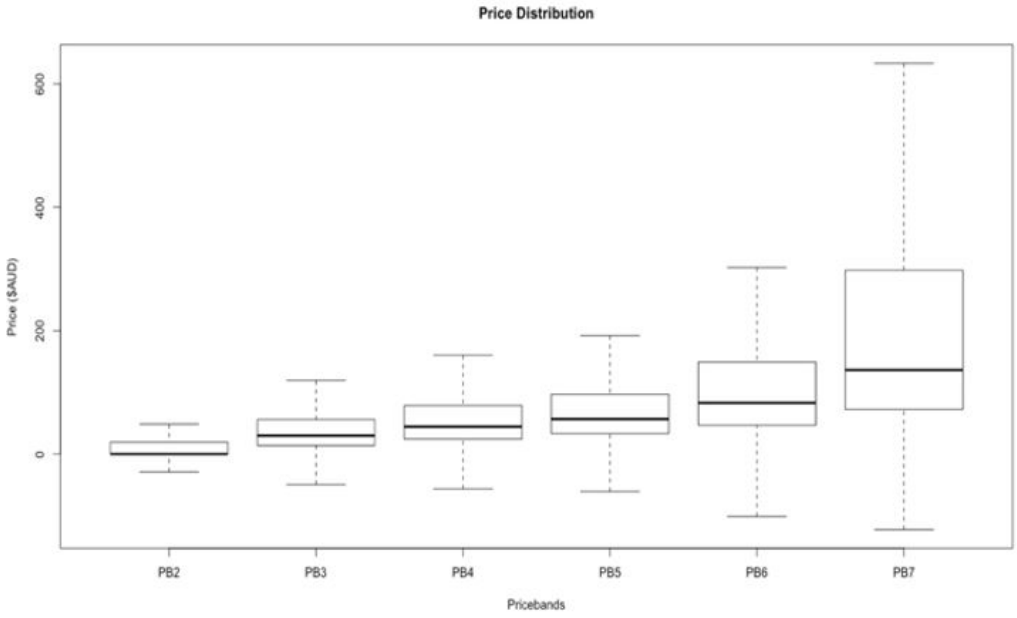


Fig A4.1.3 – Priceband Distributions (Set 2)



Tab A4.1.1 – Outliers Pricebands

Pricebands	Upper inner fence value	Number outliers Upper	Lower inner fence value	Number outliers Lower
PRICEBAND1	-889.35	122,272	-1064.15	25,186
PRICEBAND2	48.49	226,481	-29.11	231,210
PRICEBAND3	119.47	64,054	-49.49	73,219
PRICEBAND4	160.235	91,484	-57.325	53,399
PRICEBAND5	192.345	190,417	-62.255	33,847
PRICEBAND6	302.425	197,083	-106.735	20,374
PRICEBAND7	636.34	219,899	-265.98	14,344
PRICEBAND8	2786.05	329,968	-1447.23	0
PRICEBAND9	26475.865	0	-15405.455	0
PRICEBAND10	16711.405	381	8308.725	126,692

Fig A4.1.4 – Priceband Histograms

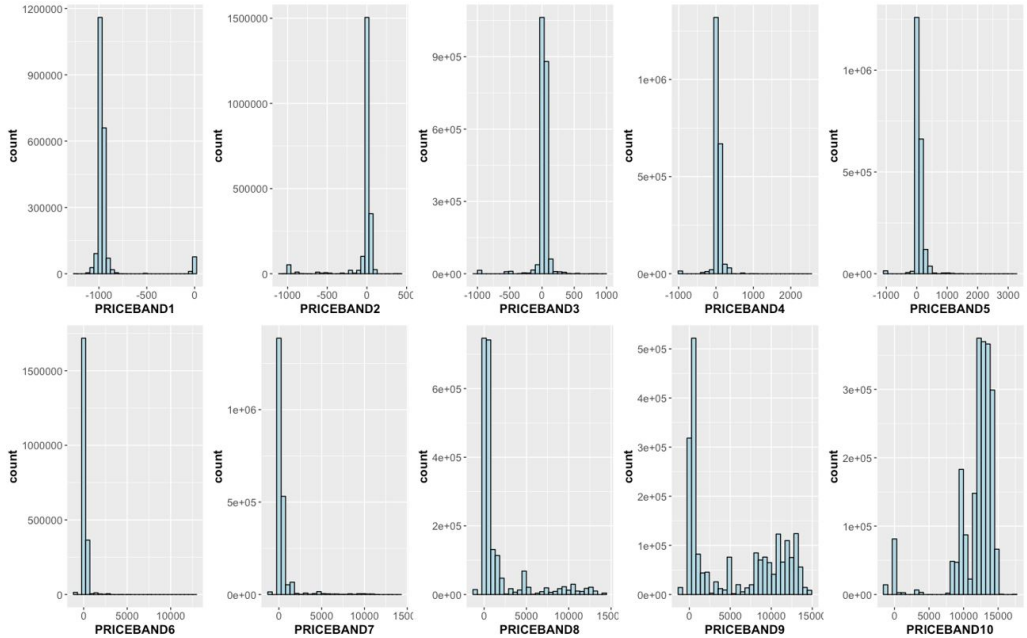


Fig A4.1.5 – BandAvail Distributions

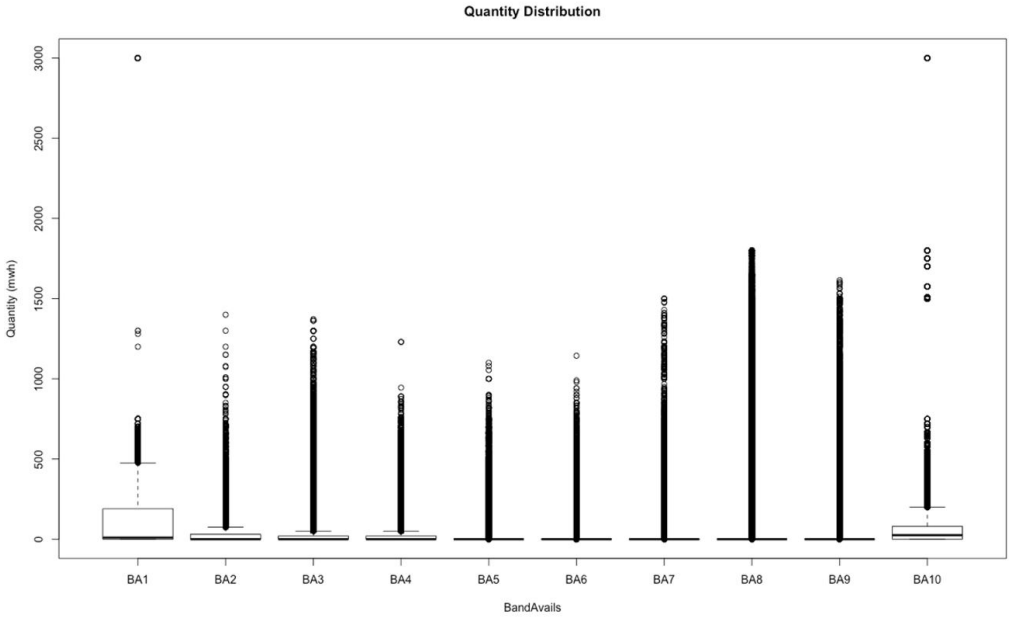


Fig A4.1.6 – BandAvail Distributions (Set1)

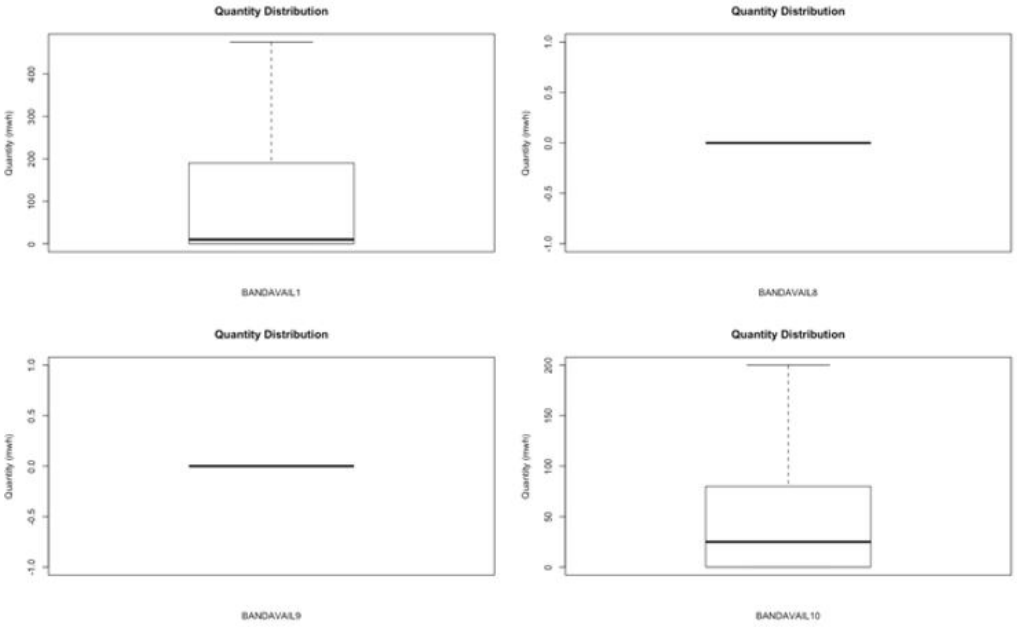
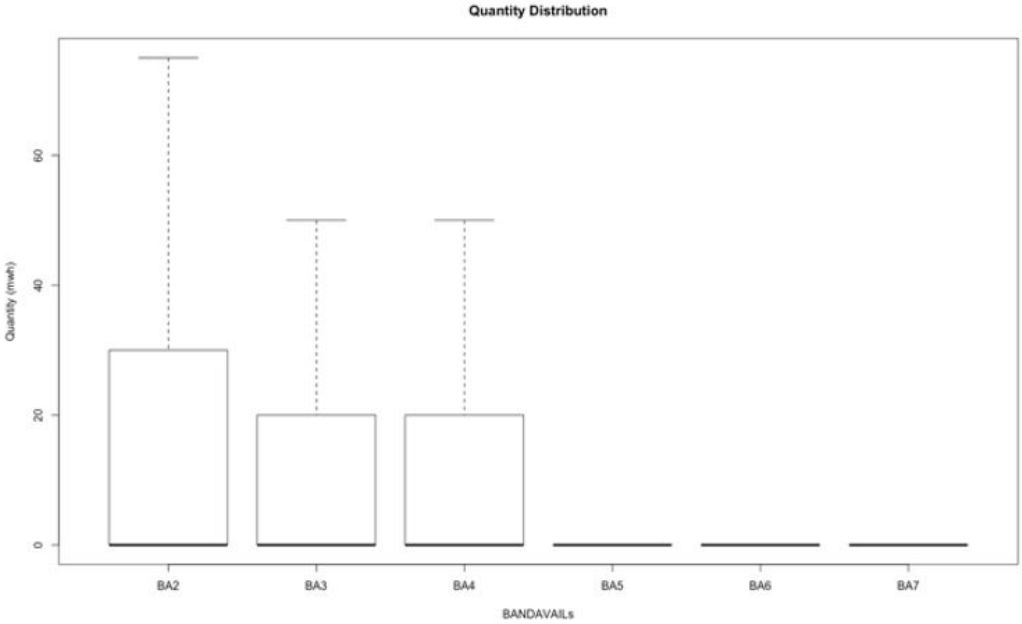


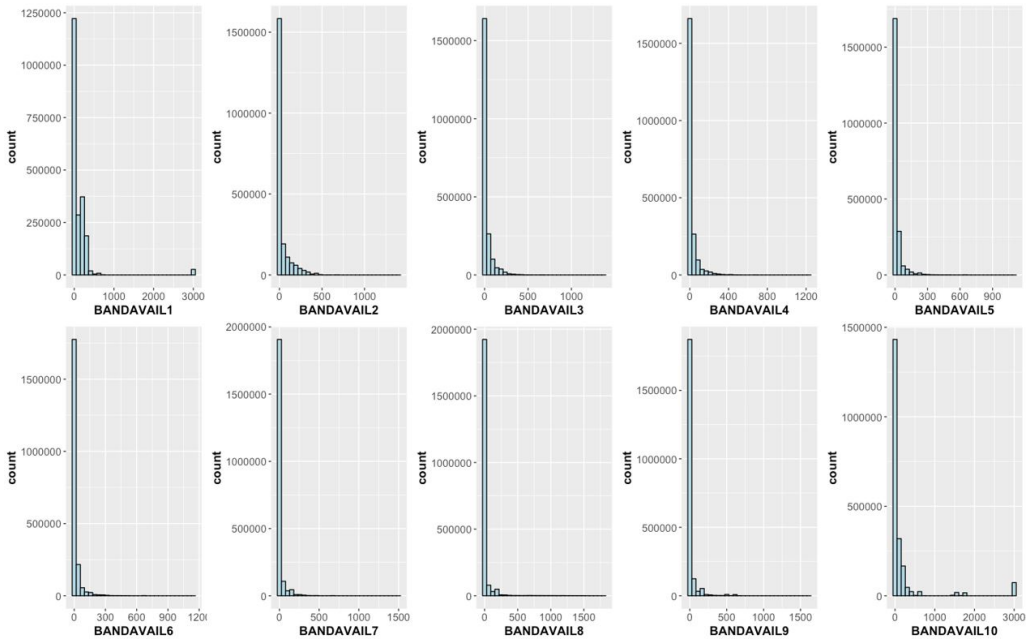
Fig A4.1.7 – BandAvail Distributions (Set2)



Tab A4.1.5 – Outliers BandAvails

BandAvails	Upper inner fence value	Number outliers Upper	Lower inner fence value	Number outliers Lower
BANDAVAIL1	475	40,088	-285	0
BANDAVAIL2	75	347,626	-45	0
BANDAVAIL3	50	294,365	-30	0
BANDAVAIL4	50	242,712	-30	0
BANDAVAIL5	0	514,601	0	0
BANDAVAIL6	0	437,182	0	0
BANDAVAIL7	0	370,122	0	0
BANDAVAIL8	0	371,043	0	0
BANDAVAIL9	0	387,791	0	0
BANDAVAIL10	200	234,784	-120	0

Fig A4.1.8 – BandAvail Histograms



References

[1] T. NIEDOBA, MULTI-PARAMETER DATA VISUALIZATION BY MEANS OF PRINCIPAL COMPONENT ANALYSIS (PCA) IN QUALITATIVE EVALUATION OF VARIOUS COAL TYPES, Physicochem. Probl. Miner. Process., 2014.
URL http://www.minproc.pwr.wroc.pl/journal/pdf/ppmp50-2.575-589.pdf

[2] S. Shukla, N. S., A Review ON K-means DATA Clustering APPROACH, International Research Publications House., 2014.
URL https://www.ripublication.com/irph/ijict_spl/ijictv4n17spl_15.pdf

[3] M. Hagenbuchner, A. C. Tsoi, Evaluation of Neural Network Models for Australian Energy Market Operators Five Minute Electricity Demand Forecasting, University of Wollongong, 2016.
URL https://www.aemc.gov.au/sites/default/files/content/924537dd-1f48-4550-a134-78b3b7d3ba70/Neural-Network-Model-Wollongong-University-final-report.pdf