Analysis of laundromats and vending machine events and transactions for I.L.R Limited using R Shiny

Table of Contents

ABSTRACT **2**

1. INTRODUCTION **2**

1.1. ORGANISATION: 2
1.2. GOALS: 3
1.3. CONSTRAINTS 3

2. DATA **3**

2.1. DESCRIPTIVE STATISTICS: 3
Transaction data *3*
Error data *3*
2.2. MISSING: 4

3. METHODOLOGY **4**

3.1. DATA RETRIEVAL AND CLEANING: 4
Transaction data *4*
Error data *5*
3.2. METHODS: 5

4. RESULTS: **6**

4.1. CHARTS AND TABLES 6
Transaction data *6*
Error data *15*
4.2. PREFERRED REPORTING PLATFORM: 21

5. CONCLUSIONS/DISCUSSION: **22**

5.1. AUTHENTICITY OF THE SUMMARIES AND VISUALISATIONS: 22
5.2. FUTURE WORK: 23
5.3. PROBLEMS: 23

6. REFERENCES: **24**

7. APPENDICES: **25**

Abstract

The report describes how Exploratory Data Analysis (EDA) was used to analyse transaction and machine Error data for laundromats and vending machines at ILR Limited. The analysis involved building two R Shiny applications to construct all the summaries and visualisations in this report.
A total of 20 visualisations were created. The report also compares the charts created in R Shiny with those made in Power BI, a platform which the company was interested in exploring.

1. Introduction

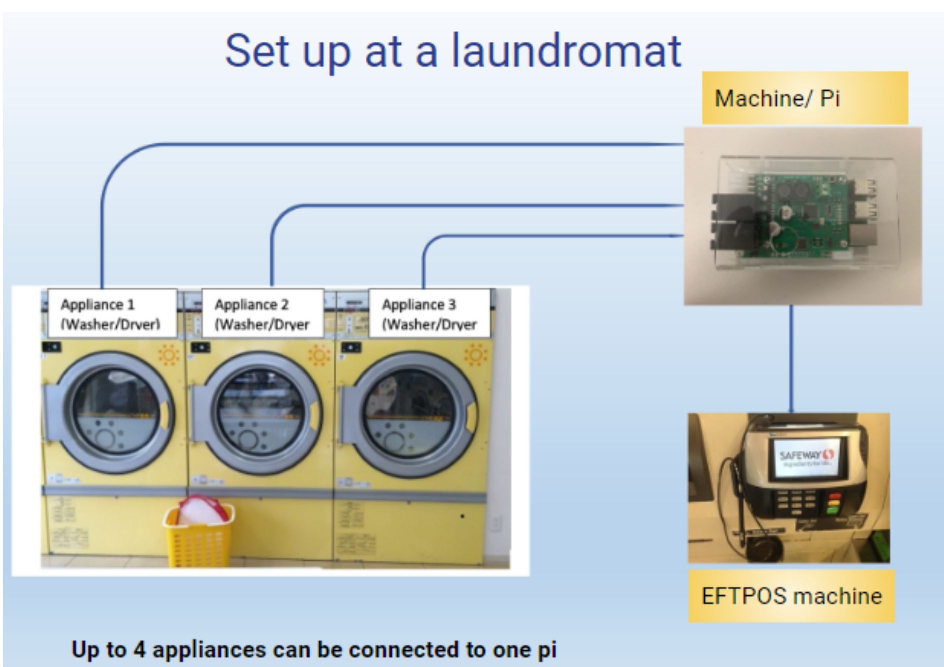
1.1. Organisation:

ILR Limited specialises in hardware and embedded software engineering. They offer design and engineering expertise to manufacturing and engineering companies that require hardware and embedded software engineering. The company is based in Christchurch and only has six employees.

One of their clients provides EFTPOS support for laundromats and vending machines in the North and South Islands of New Zealand. Since laundromats and vending machines operate 24 hours a day, large amounts of data on machine status and EFTPOS transactions were sent to the cloud for analysis, with about 15 000 records of daily transaction data.

At laundromats, the washers and dryers (appliances) connect to a machine (also called a pi), which then connects to a payment terminal, usually an EFTPOS or coin-drop machine. A machine/pi can connect to more than one washer or dryer. Figure 1.1 below illustrates this connection.

Figure 1.1. Diagram showing how appliances were connected to machines/pis



1.2. Goals:

This project aimed to use Exploratory Data Analysis (EDA) to understand what was happening at each site and minimise the impact of machine errors by proactively sending technicians to fix them.

1.3. Constraints

The data did not have information on the sites' latitudes and longitudes, which could have been used to create a map in R Shiny showing the locations of the appliances. A map would be helpful to new technicians visiting sites to attend to machine or appliance faults.

It would have been good to know more about models, makes, capacities and machine ages as extra dimensions to plot. These could have been used to investigate whether specific machines lasted longer than others. Unfortunately, that information was not available in the datasets provided.

2. Data

2.1. Descriptive statistics:

The company stored all its data in google cloud, and for this project, two datasets, Transaction data and Error data, were provided. The data was supposed to be queried directly from the cloud without downloading.

The datasets had a shared ID ("serialNumber" in the Transaction data and "gateway_serialNumber" in the Error data) which could be used to join them in R.
Since the data was sent directly to the cloud from the machines, there were no problems related to data authenticity.
The quantity of the data varied due to the period selected for reporting, though data for one day would have an average of 15 000 records and about 400 000 records for a month. Data for the period January 2021 to December 2022 had more than 6.7 million records.
The tables and graphs in this report were created using data for December 2022, which had 396 735 entries for Transaction data and 1950 entries for Error data.

Transaction data

Transaction data comprised thirty-two variables. The variables were a mixture of strings, integers and date-time. Since one of the company's goals was to maximise sales, the variable "total_cost_dollars" was considered the most important.

Error data

The Error data had only eight variables. These were a mixture of strings, integers and date-time, just like Transaction data. Errors were categorised into three levels, namely "error", "critical", and "fatal".

2.2. Missing:

There was high missingness in the Transaction data. The variables “siteId" and “applianceName" had high missingness. Some variables with excessively high missingness were of low importance and were not used in making the EDAs.
There were sites with missing site names, site IDs, and appliance names, a problem caused by machines using an older firmware version that did not support showing all this information. The missing values could not be backfilled programmatically since we did not know which appliance raised how much in sales.

3. Methodology

3.1. Data retrieval and cleaning:

The data was retrieved from BigQuery using an SQL code run in R studio, and EDAs generated using a purpose-built R Shiny app.
A series of cleaning steps were taken to ensure meaningful graphs were constructed.
The variable “siteName" was concatenated with the variable “contractName”, which solved the problem of sites with the same names but different contractors.

BigQuery uses the Coordinated Universal Time "UTC", so time was converted into local New Zealand time in R after retrieval. This was because the SQL code in BigQuery did not convert time from UTC to New Zealand Daylight Time “NZDT”, but it only retrieved data 13 hours ahead of UTC. New columns were created in the dataset to restore the “UTC” datetime to “NZDT”. Then, the Datetime variable was converted to a Date variable used to make all the plots and summaries.
Duplicate rows were removed from the data using the SQL code “distinct”, and the data was ordered by date from the newest to the oldest, and strings were converted to factors.

A date range input was added in the Shiny UI file, prompting the user to enter the required dates for the reporting period.

Transaction data

“Transaction_datetime” was converted to a date variable because the company was interested in daily transactions. The variable “Terminal_transaction_type” was filtered to include "purchase" transaction types only since non-purchase transaction types (“logon”, “restock”, and “status”) were irrelevant in the analysis.
The variable “Terminal_result_string” had strings "DENIED", "DECLINED", Cancelled", "CANCELLED", "TERMINAL BUSY", and "Busy". "DENIED" was renamed "DECLINED" whilst "Cancelled" was renamed "CANCELLED" and "TERMINAL BUSY" was renamed "Busy".
The sales in BigQuery were in cents, so these were transformed into dollars by dividing by 100.

Error data

The variable “Timestamp”, which was a datetime variable, was converted to a date variable. There were no missing observations in the Error data since the variables of importance were “Timestamp”, “Message”, “SerialNumber”, and “MachineId”.

3.2. Methods:

To create data visualisations that could easily be understood and generated anytime, R Shiny and Power-BI were used to perform the data analytics.

Generating charts in Power BI was easy since it did not require coding expertise. The charts were readily available, and assembling visualisations was “a drag-and-drop process” (Becker, 2019), making it a preferred choice for interactive visualisations at first glance. However, only simple charts with little filtering could be created. Some of the visualisations required a lot of filtering and coding, making R Shiny a better option than Power BI. In addition, Power BI did not have an accessible source code, and data transformation was impossible while working in DirectQuery mode.

The company also wanted the visuals and tables to be hyperlinked to the cloud data so the user could be directed to the appliance and the site name in the payments cloud web app when clicked. This requirement could be done easily in a Shiny app but was impossible in Power BI DirectQuery mode.

Though more time was required to code reports in R Shiny, the generated visualisations could be tailored to suit the company's requirements.

4. Results:

4.1. Charts and tables

Thirteen visualisations (nine plots and four tables) were created using Transaction data, whilst seven visualisations (four plots and three tables) were created using the Error data, making a total of twenty visualisations.

Transaction data

Figure 4.1.1. Transactions data Shiny app starter page

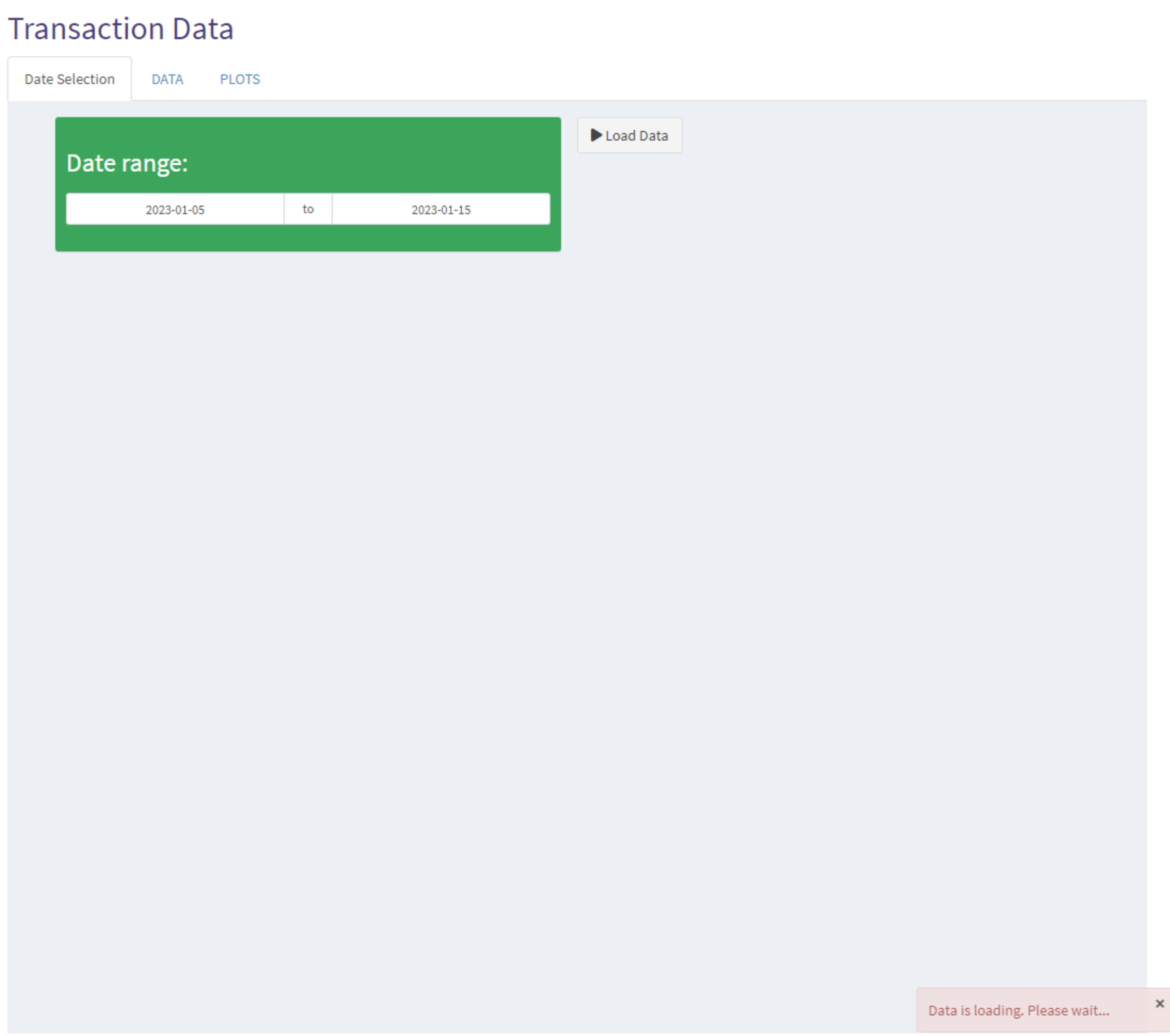


Figure 4.1.1 above shows the Shiny main page where the users entered the dates for the reporting period. When the “Load Data” button was pressed, a red dialogue box would pop up to let the user know that the data was loading, and when loading was complete, a blue dialogue box would pop up to let the user know that loading was complete.

The company must keep an eye on approved and unapproved transactions. Though approved transactions were expected to be higher than unapproved ones, machines and appliances with a high percentage of unapproved transactions had to be checked in the system. The figure below shows transaction results.

Figure 4.1.2. Terminal result string summary and plot

Transaction Data for DATA 601

Date Selection | DATA | PLOTS

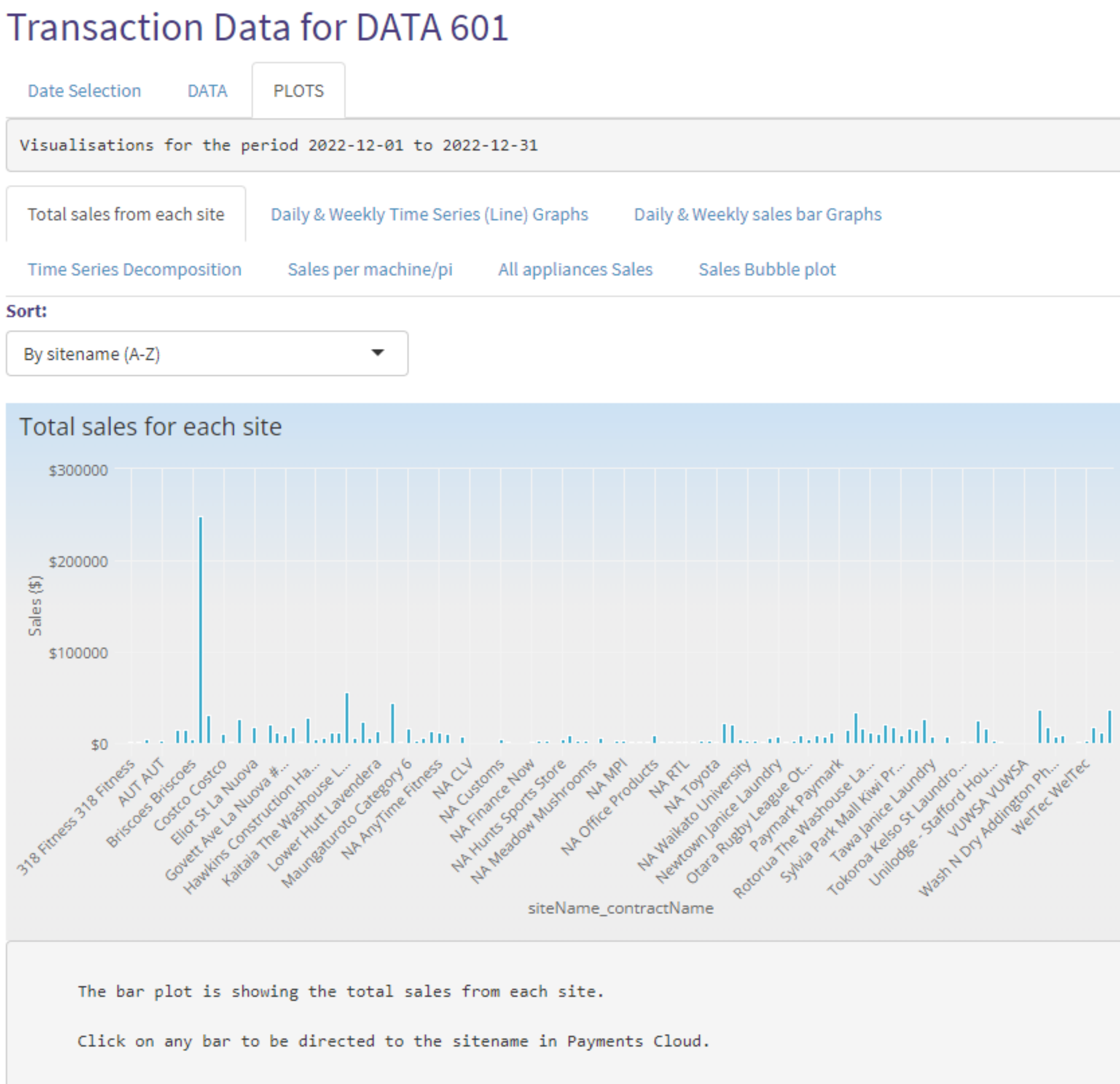
```
Visualisations for the period 2022-12-01 to 2022-12-31
```

Terminal result string summary and plot | Total sales from each site | Daily & Weekly Time Series Graphs of Sales | Time Series Decomposition | All appliances and all machine sales | Sales Bubble plot


```
  terminal_result_string    Amount
1               APPROVED 1308732.8
2              CANCELLED  624877.5
3               DECLINED  109457.8
4                   Busy   13793.6
5       HOST UNAVAILABLE    8690.6
6                 FAILED      93.0
```

```
If APPROVED transactions are less than the sum of the other strings, we
have problems that need to be investigated.

A look into the 'Terminal result string details' table and
'Approved transactions as a percentage' table can give us
some light into the appliances with issues that need to be checked.
```

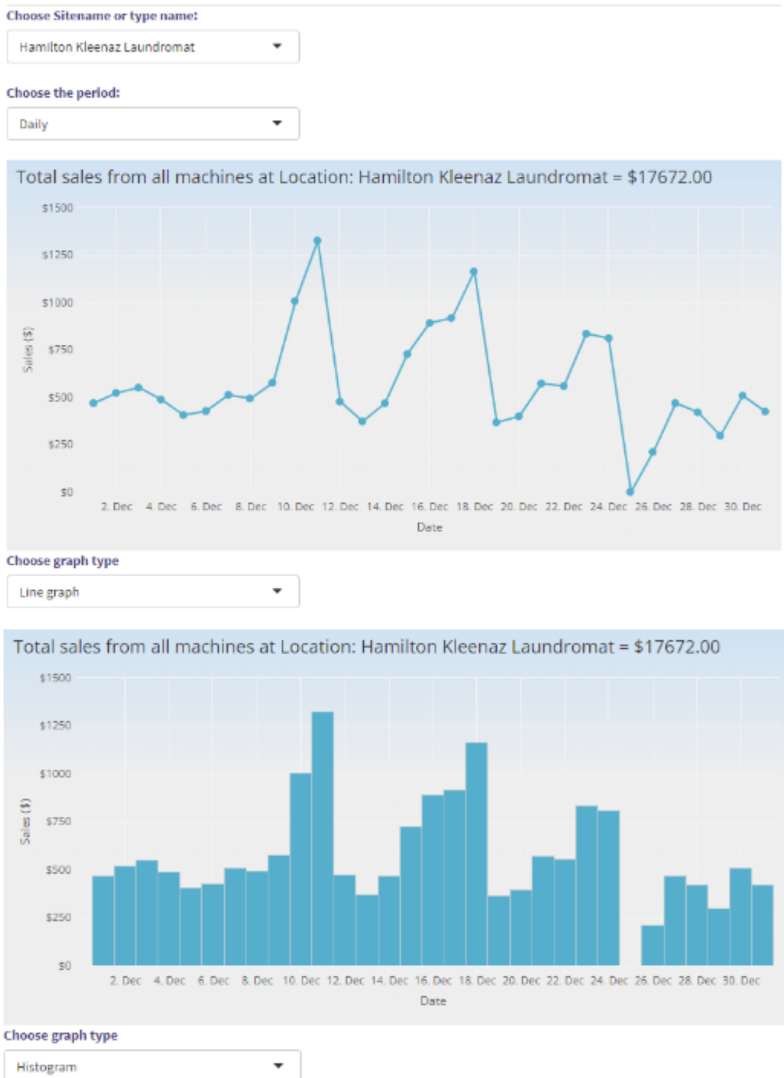
Figure 4.1.3. Transactions data Total sales plot



The plot above shows the total sales from each site for December 2022. The chart could sort by site name or sales, from largest to smallest.

When clicked, each bar would direct the user to the payments cloud web application where the site name would open, showing all the appliances that transacted to give the amount in the bar chart.

Figure 4.1.4. Time series line plot and histogram of daily and weekly sales



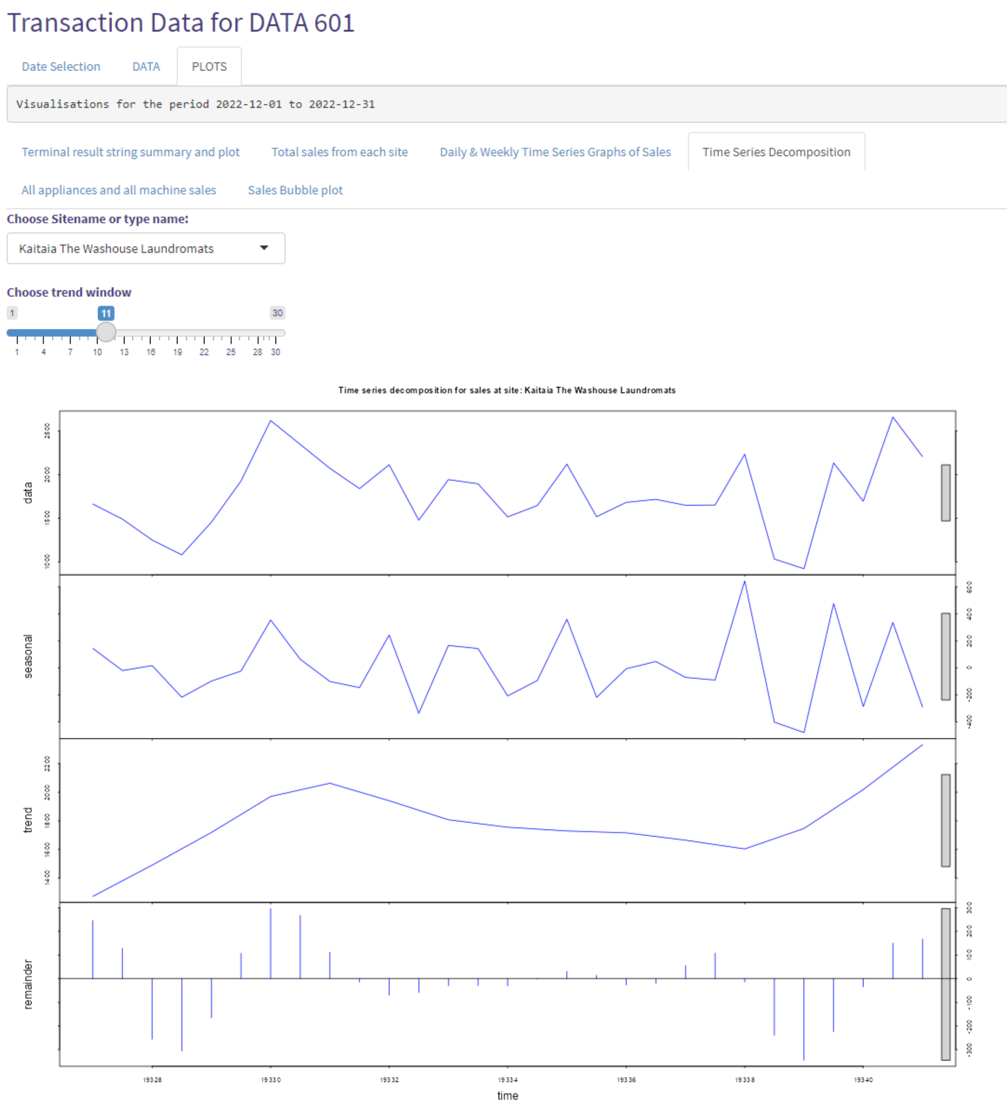

```
This plot shows the daily and weekly sales for Hamilton Kleenaz Laundromat.
Note the peaks and troughs for each site to see if they are following a cycle.
As a company, we need to think of strategies we can use to increase the
sales on those days where sales are usually low and to get more sales on
days where sales are usually high.

Possible strategies would be:
1. Give discounted prices on the days when sales are usually low.
2. Create raffles where a minimum amount is required to enter.
3. Increase the price on those days when sales are high.
The line graph allows us to easily see the trend for sales, especially
over longer periods.
```

The time series plot shows daily sales from the site “Hamilton”. For the graphs that showed cyclic trends, possible actions were discussed to increase sales on those days when they were periodically low and to maintain high sales on the days when peaks were observed.

The chart had options to choose the site name, as well as choosing between a daily and a weekly time series and between a line graph and a histogram, as shown in Figure 4.1.4 above. The line graph and histogram for the weekly sales are shown in Appendix 4 and 5.

The plot also showed the name of the selected site and the total sales collected. The trend and cyclic behaviours portrayed in the line graphs were analysed further using time series decomposition.

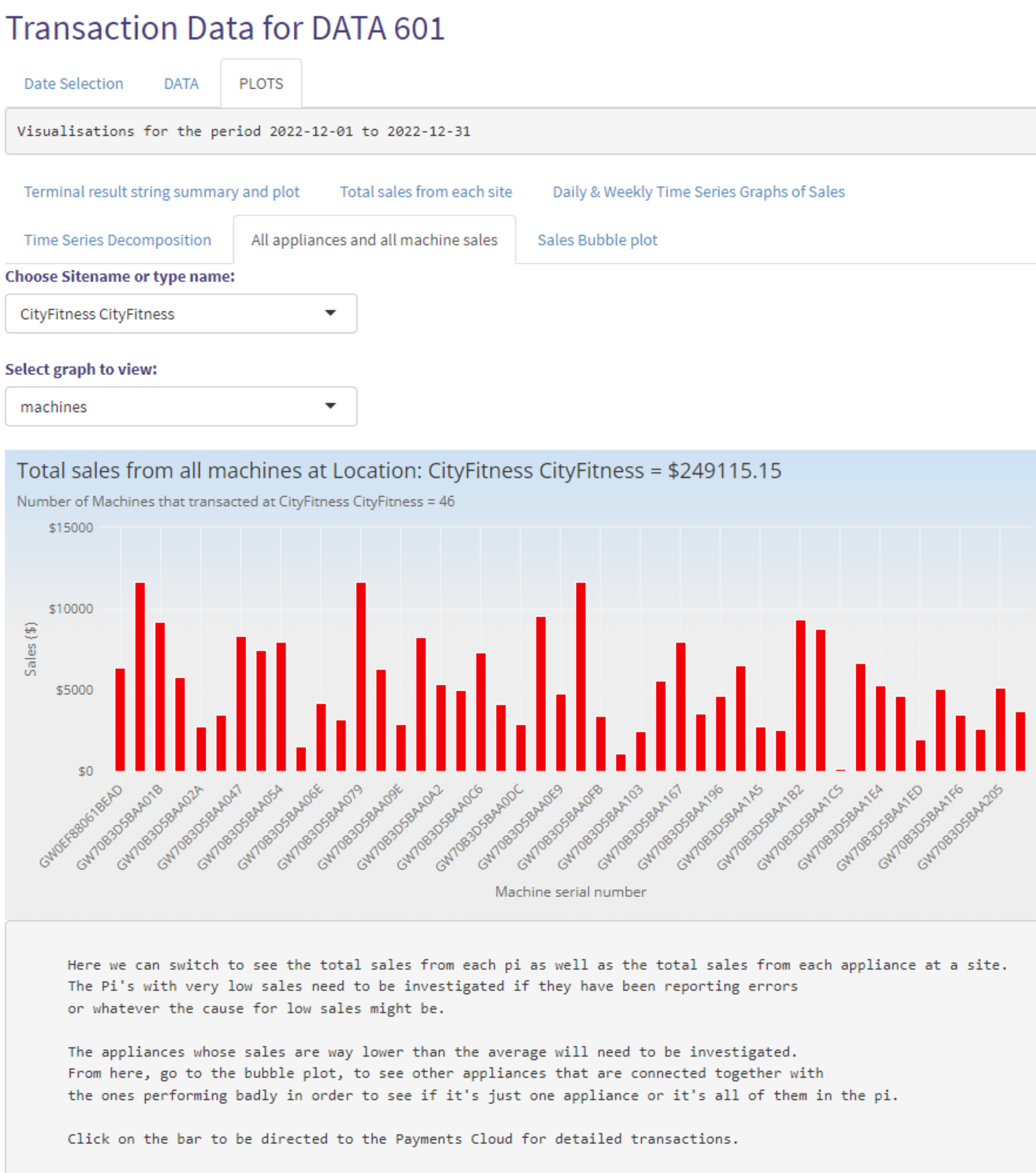
Figure 4.1.5. Time Series decomposition of the daily transactions



The daily sales time series was decomposed into the seasonal, the trend, and the remainder components, with a slider to choose the values of t.window. STL decomposition was used instead of SEATS and X11 due to its ability to handle any seasonality, not only monthly and quarterly data (Hyndman and Athanasopoulos, 2018).

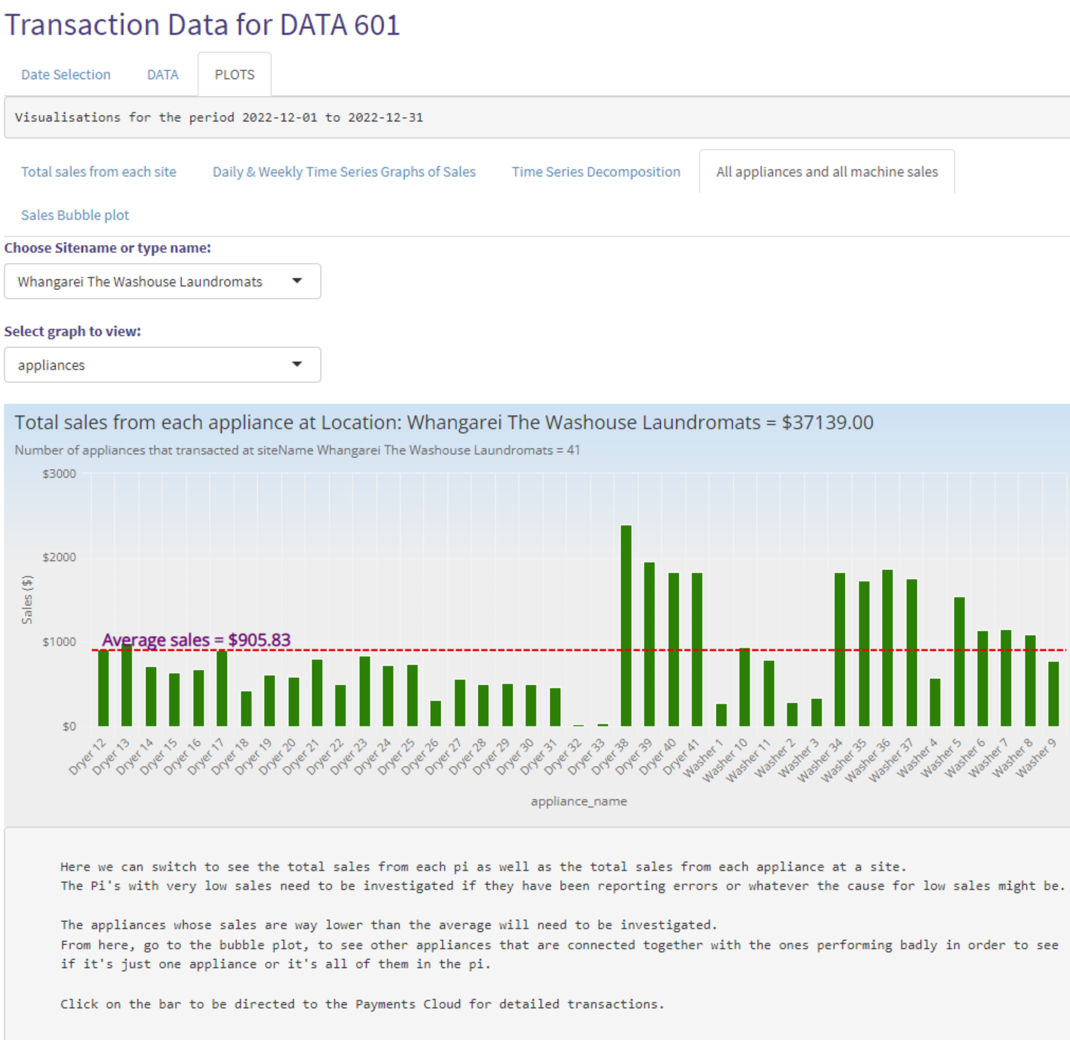
The seasonal chart shows weekly cycles in the Transaction data. This was because the data showed recurring patterns as most sites had high sales over the weekends and low sales in the midweek, on days like Wednesday and Thursday.
The time series decomposition for the Error data is illustrated in Appendix 2.

Figure 4.1.6. Bar plot for machine sales



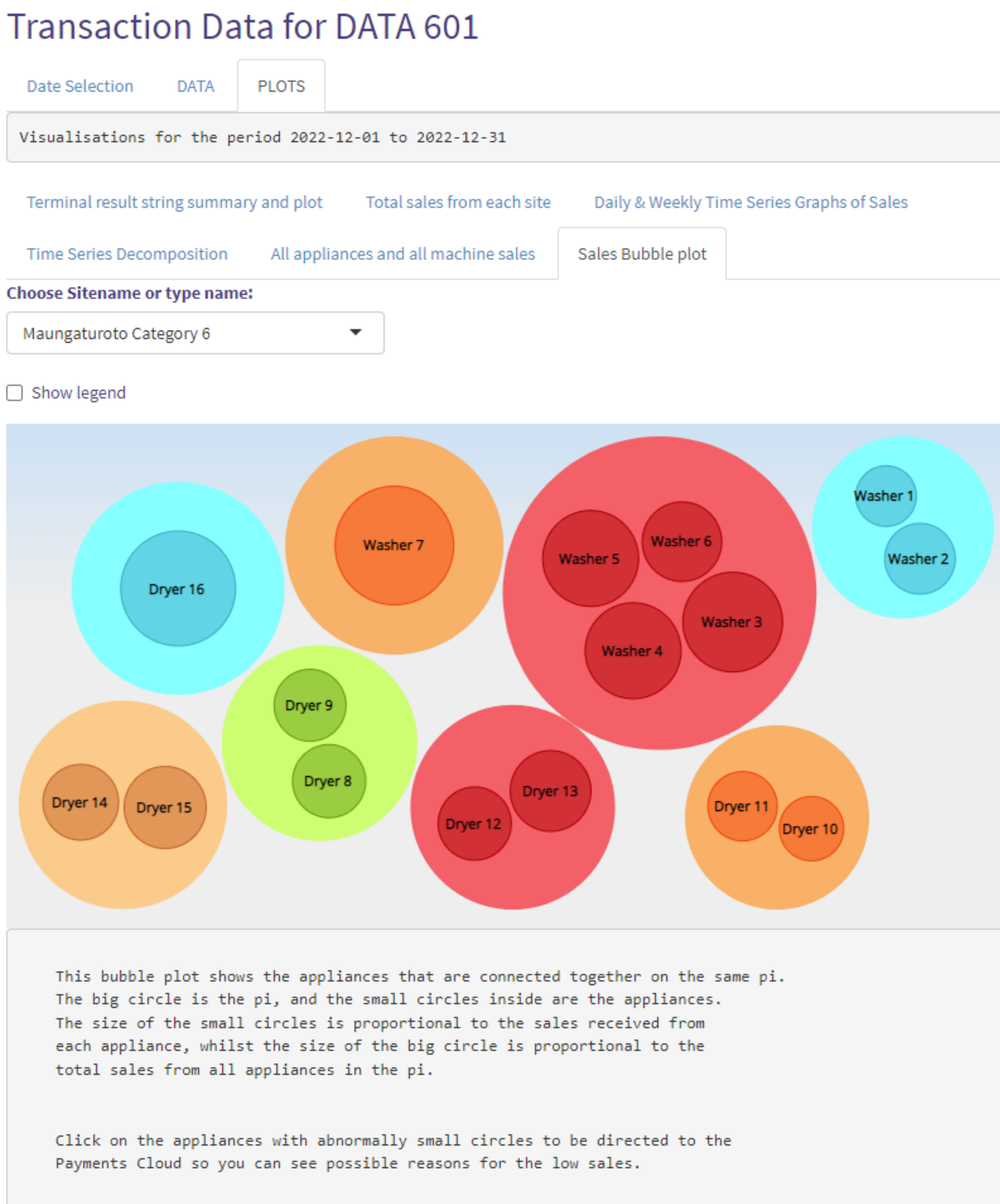
The bar plot shows all the machines and the sales they raised from each site. The bars were hyperlinked to the company web app, where detailed transactions for the machines could be found. The machines with significantly low sales had to be checked to see if they were operating well. Usually, those machines would have a high number of unapproved transactions.
The chart enabled the users to choose between the bar plot for the machine sales or the appliance sales.

Figure 4.1.7. Bar plot for appliance sales



The plot above shows all the appliances and their sales on each site. The red horizontal line is the average sales for the site for the period chosen. This plot was used to see which appliances were underperforming (had significantly low sales compared to the average sales).

Figure 4.1.8. Bubble plot for sales from each appliance and machine



The bubble plot showed the appliances connected to a machine (pi). The sizes of the inner circles were proportional to the sales from each appliance, whilst the sizes of the outer circles (pis) were proportional to the total sales from the appliances inside those circles.
The bubbles were also hyperlinked to the web app, so clicking on each of them would open a link to the payments cloud web where the clicked appliances would be revealed.

Table 4.1.1. Terminal result string table

Transaction Data

DATA PLOTS

Summaries for the period 2022-12-01 to 2022-12-31

Terminal result string summary | Terminal result string Details | Appliances with less than 50% Approved Transactions | Raw Data | Data Summary

```
Average Percentage of Approved Transcations in Whangarei The Washouse Laundromats = 84.15%
```

Choose Sitename or type name:

Whangarei The Washouse Laundromats

Copy CSV Excel Print PDF

Search:

	appliance_name	total_transactions	approved_transactions	approved_percentage	cancelled	cancelled_percentage	declined	declined_percentage	failed	failed_percentage	busy	busy_percentage	unavailable	unavailable_percentage	machine_url
1	Dryer 12	174	162	93.10%	2	1.10%	10	5.70%	0	0.00%	0	0.00%	0	0.00%	link
2	Dryer 13	193	169	87.60%	4	2.10%	20	10.40%	0	0.00%	0	0.00%	0	0.00%	link
3	Dryer 14	147	122	83.00%	11	7.50%	14	9.50%	0	0.00%	0	0.00%	0	0.00%	link
4	Dryer 15	118	104	88.10%	4	3.40%	10	8.50%	0	0.00%	0	0.00%	0	0.00%	link
5	Dryer 16	135	120	88.90%	8	5.90%	7	5.20%	0	0.00%	0	0.00%	0	0.00%	link
6	Dryer 17	170	156	91.80%	4	2.40%	10	5.90%	0	0.00%	0	0.00%	0	0.00%	link
7	Dryer 18	88	77	87.50%	3	3.40%	8	9.10%	0	0.00%	0	0.00%	0	0.00%	link
8	Dryer 19	115	100	87.00%	6	5.20%	9	7.80%	0	0.00%	0	0.00%	0	0.00%	link
9	Dryer 20	113	105	92.90%	3	2.70%	5	4.40%	0	0.00%	0	0.00%	0	0.00%	link
10	Dryer 21	157	136	86.60%	8	5.10%	13	8.30%	0	0.00%	0	0.00%	0	0.00%	link

Showing 1 to 10 of 41 entries — Previous 1 2 3 4 5 Next

```
This is a breakdown of all the terminal result strings.
NOTE: All percentages should add to 100%.
A 'good' appliance should have more than 50% of its transactions approved.
A 'very good' appliance should have more than 90% of its transactions approved.

ALL sites with no appliance names will give an ERROR.
ALL sites with NAs (no sitename) will give an ERROR.

When clicking on the link, choose 'Open in new window' so you can continue working on this page.
```

As shown in Table 4.1.1 above, different terminal results were summarised for each appliance at a site. Information from this table was used to give an overview of how approved transactions compared to unapproved transactions per appliance. For different approved transaction percentages, table 4.1.2 below was used.

Table 4.1.2. Approved transactions as a percentage

Transaction Data for DATA 601

Date Selection DATA PLOTS

```
Summaries for the period 2022-12-01 to 2022-12-31
```

Terminal result string summary | Terminal result string Details | Approved transactions as a percentage | Raw Data | Data Summary

Maximum percentage approved (0.5 = 50%)

0 — 0.8 — 1

0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1

Copy CSV Excel Print PDF

Search:

	appliance_name	gateway_serial_number	siteName_contractName	gateway_software_version	total_transactions	approved_transactions	percentage_approved	cancelled_transactions	percentage_cancelled	declined	percentage_declined	machine_url
1	10 min = $10	GWB827EB568B2C	Dogwash The Hub Laundromat	4.2.0.7	2	1	50.00%	1	50.00%	0	0.00%	link
2	Dryer 10	GWB827EB0DA8AA	Hillcrest Maxaclean Laundromat	4.2.0.11	91	53	58.00%	23	25.00%	4	4.00%	link
3	Dryer 10	GWB827EB157DE3	Hamilton Kleenaz Laundromat	4.2.0.11	99	76	77.00%	10	10.00%	11	11.00%	link
4	Dryer 10	GWB827EB59D5E2	Newtown Janice Laundry	4.2.0.11	28	22	79.00%	3	11.00%	2	7.00%	link
5	Dryer 10	GWB827EB690665	Kaitaia The Washouse Laundromats	4.2.0.11	270	213	79.00%	24	9.00%	29	11.00%	link
6	Dryer 10	GWB827EB8556DB	Nurses Quarter Nurses Quarter	4.2.0.11	16	12	75.00%	3	19.00%	1	6.00%	link
7	Dryer 10	GWB827EB90B512	Eliot St La Nuova	4.2.0.11	176	141	80.00%	23	13.00%	12	7.00%	link
8	Dryer 10	GWB827EBAEA882	Tokoroa Kelso St Laundromat - Tokoroa	4.2.0.11	21	14	67.00%	3	14.00%	3	14.00%	link
9	Dryer 10	GWB827EBEF6F20	Otahuhu Laundrocuts	4.2.0.11	78	50	64.00%	12	15.00%	9	12.00%	link
10	Dryer 11	GWB827EB0DA8AA	Hillcrest Maxaclean Laundromat	4.2.0.11	92	66	72.00%	15	16.00%	4	4.00%	link

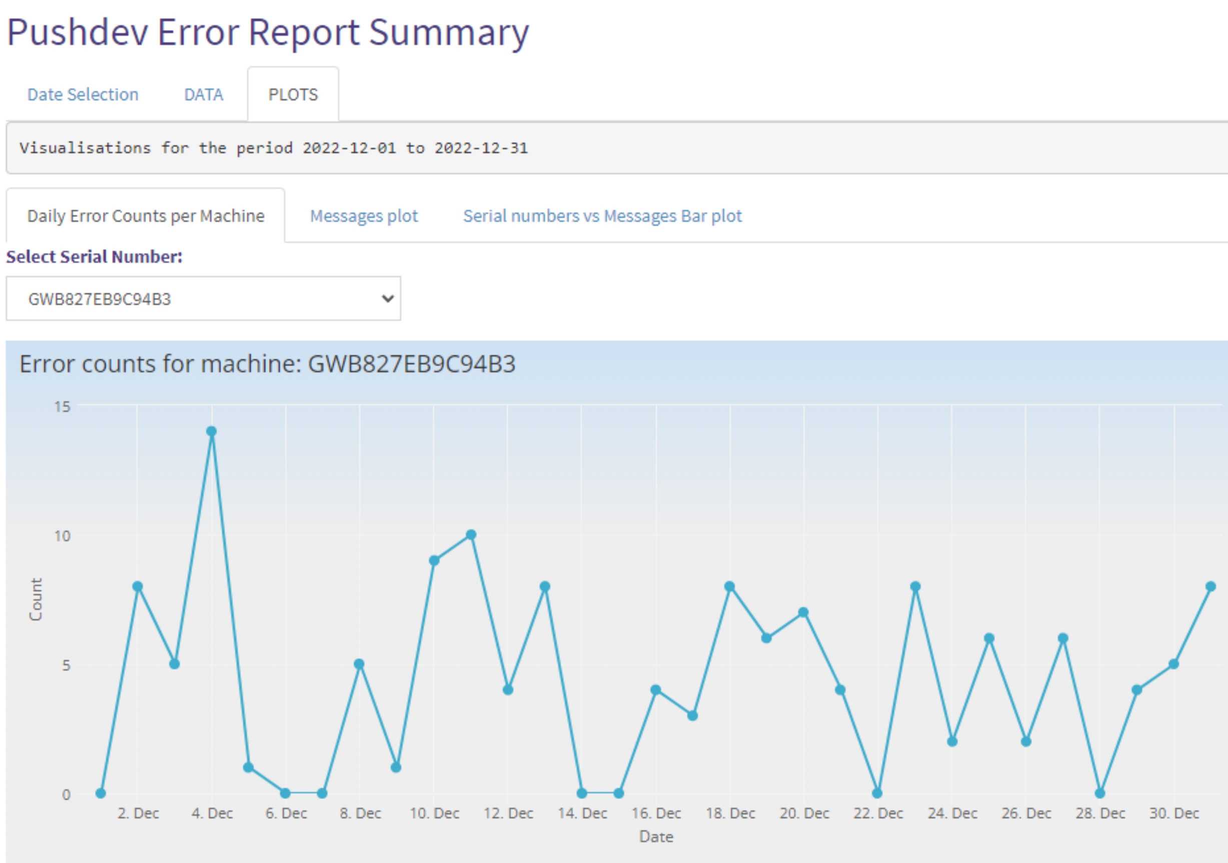
Showing 1 to 10 of 357 entries — Previous 1 2 3 4 5 … 36 Next

```
These appliances need to be reported to the technical team (Simon) so they look to see why they have a high percentage of unapproved transactions.
When clicking on the link, choose 'Open in new window' so you can continue working on this page.
```

Table 4.1.2 shows a list of appliances with a selected percentage of approved transactions. The table above shows appliances with a maximum of 80% approved transactions. Usually, appliances with less than 50% approved transactions would be checked in the payments cloud web app to see if there were any issues with their functionality. If they were reported to be working correctly in the web app, someone would ask the technical team to check these appliances physically.

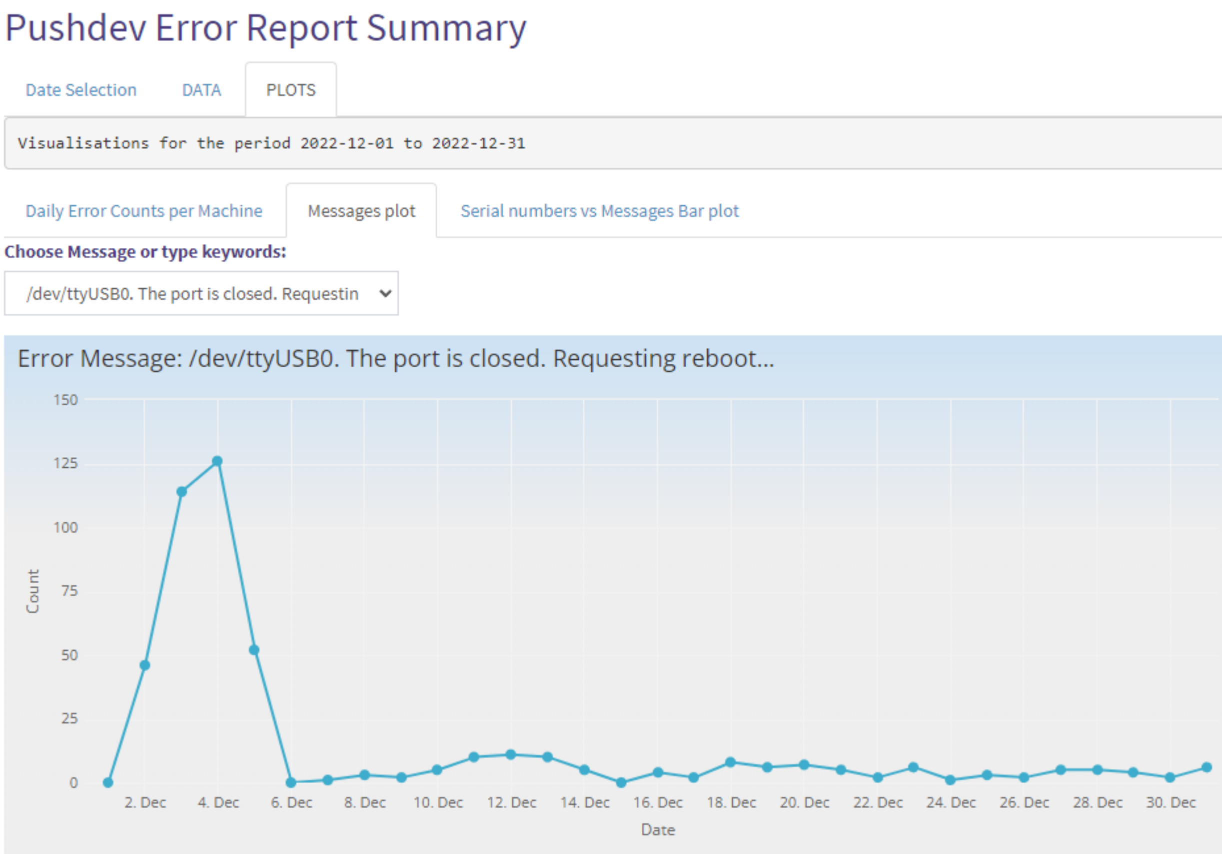
Error data

Figure 4.1.8. Time series plot for machine error reports



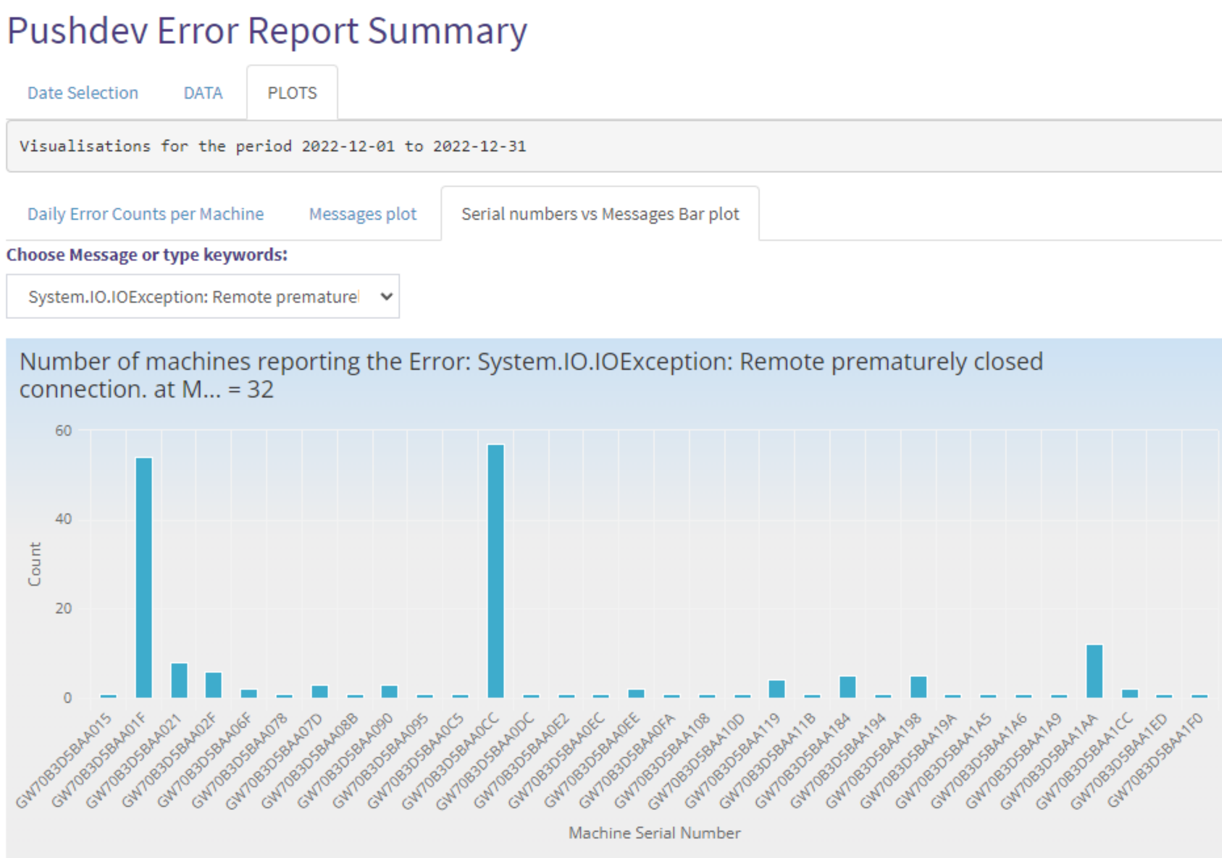
The plot shows the number of errors reported by machine GWB827EB9C94B3 between the 1st of December and the 31st of December 2022. The highest number of errors was seen on the 4th of December, when 14 error messages were recorded.

Figure 4.1.9. Time series plot for daily error reports



The time series plot in figure 4.1.9 shows the number of times the Port-0 error was reported each day in December. About 125 Port-0 error messages were reported on the 4th of December.

Figure 4.1.10. Bar plot for machines that reported the same error



The chart above shows the machines that reported the same error and the number of times each machine reported that error. In this figure, a total of 32 machines reported the error message "Remote prematurely closed connection", and machines GW70B3D5BAA01F and GW70B3D5BAA0CC reported this error more than 50 times each. This chart made it easy for the company to see the top machines reporting a particular error.

Table 4.1.4. Machine, message and error count table

Pushdev Error Report Summary

DATA PLOTS

Summaries for the period 2022-12-01 to 2022-12-31

Message counts | Serial No, Message, Counts | Daily error counts | Raw Data | Data Summary

Copy CSV Excel Print PDF

Search:

	serialNumber	message	Count	machine_url
1	GWB827EB1C76ED	/dev/ttyUSB1. The port is closed. Requesting reboot...	330	link
2	GWB827EB1C76ED	/dev/ttyUSB0. The port is closed. Requesting reboot...	320	link
3	GWB827EB568B2C	An error occurred while reading the key ring.	142	link
4	GWB827EB1C76ED	/dev/ttyUSB0 saw exception during write. System.IO.IOException: No such device at System.IO.Ports.SerialStream.Write(Byte[] array, Int32 offset, Int32 count, Int32 timeout) at System.IO.Ports.SerialPort.Write(Byte[] buffer, Int32 offset, Int32 count) at IoProvider.SerialIo.SendData(List`1 dataToSend) in /builds/payments-group/VendingIoT/IoProvider/SerialIo.cs:line 311	80	link
5	GWB827EB1C76ED	/dev/ttyUSB1 saw exception during write. System.IO.IOException: No such device at System.IO.Ports.SerialStream.Write(Byte[] array, Int32 offset, Int32 count, Int32 timeout) at System.IO.Ports.SerialPort.Write(Byte[] buffer, Int32 offset, Int32 count) at IoProvider.SerialIo.SendData(List`1 dataToSend) in /builds/payments-group/VendingIoT/IoProvider/SerialIo.cs:line 311	78	link
6	GWB827EB9C94B3	/dev/ttyUSB1. The port is closed. Requesting reboot...	64	link
7	GWB827EB9C94B3	/dev/ttyUSB0. The port is closed. Requesting reboot...	63	link
8	GW70B3D5BAA0CC	System.IO.IOException: Remote prematurely closed connection. at Mono.Net.Security.AsyncProtocolRequest+d__24.MoveNext () [0x000e7] in <954857ea8d554be3854ef76a54d3dce2>:0 --- End of stack trace from previous location where exception was thrown --- at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw () [0x0000c] in <063042d84f0c4a14831bb75028a24f76>:0 at System.Runtime.CompilerServices.TaskAwaiter.ThrowForNonSuccess (System.Threading.Tasks.Task task) [0x0003e] in <063042d84f0c4a14831bb75028a24f76>:0 at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification (System.Threading.Tasks.Task task) [0x00028] in <063042d84f0c4a14831bb75028a24f76>:0 at System.Runtime.CompilerServices.TaskAwaiter.ValidateEnd (System.Threading.Tasks.Task task) [0x00008] in <063042d84f0c4a14831bb75028a24f76>:0 at System.Runtime.CompilerServices.ConfiguredTaskAwaitable+ConfiguredTaskAwaiter.GetResult () [0x00000] in <063042d84f0c4a14831bb75028a24f76>:0 at Mono.Net.Security.AsyncProtocolRequest+d__23.MoveNext () [0x0008b] in <954857ea8d554be3854ef76a54d3dce2>:0 --- End of stack trace from previous location where exception was thrown --- at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw () [0x0000c] in <063042d84f0c4a14831bb75028a24f76>:0 at Mono.Net.Security.MobileAuthenticatedStream+	57	link
9	GW70B3D5BAA01F	System.IO.IOException: Remote prematurely closed connection. at Mono.Net.Security.AsyncProtocolRequest+d__24.MoveNext () [0x000e7] in <954857ea8d554be3854ef76a54d3dce2>:0 --- End of stack trace from previous location where exception was thrown --- at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw () [0x0000c] in <063042d84f0c4a14831bb75028a24f76>:0 at System.Runtime.CompilerServices.TaskAwaiter.ThrowForNonSuccess (System.Threading.Tasks.Task task) [0x0003e] in <063042d84f0c4a14831bb75028a24f76>:0 at System.Runtime.CompilerServices.TaskAwaiter.HandleNonSuccessAndDebuggerNotification (System.Threading.Tasks.Task task) [0x00028] in <063042d84f0c4a14831bb75028a24f76>:0 at System.Runtime.CompilerServices.TaskAwaiter.ValidateEnd (System.Threading.Tasks.Task task) [0x00008] in <063042d84f0c4a14831bb75028a24f76>:0 at System.Runtime.CompilerServices.ConfiguredTaskAwaitable+ConfiguredTaskAwaiter.GetResult () [0x00000] in <063042d84f0c4a14831bb75028a24f76>:0 at Mono.Net.Security.AsyncProtocolRequest+d__23.MoveNext () [0x0008b] in <954857ea8d554be3854ef76a54d3dce2>:0 --- End of stack trace from previous location where exception was thrown --- at System.Runtime.ExceptionServices.ExceptionDispatchInfo.Throw () [0x0000c] in <063042d84f0c4a14831bb75028a24f76>:0 at Mono.Net.Security.MobileAuthenticatedStream+	54	link
10	GWB827EBEF07B5	/dev/ttyUSB2. The port is closed. Requesting reboot...	47	link

Showing 1 to 10 of 224 entries

Previous 1 2 3 4 5 ... 23 Next

```
The table shows a summary of the 'Serial numbers vs Messages Bar Plot'.
Here we can sort the messages and select the machines with the highest counts.
If the table fails to load, go to the server file and comment out the code in line 53.
```

Table 4.1.4 is the table version of Figure 4.1.10. It shows a summary of the error messages reported by each machine and the number of times the errors were reported. The link, when clicked, would open the company's web application where the machine will be displayed.

Table 4.1.5. Error message counts

Pushdev Error Report Summary

Date Selection | DATA | PLOTS

```
Summaries for the period 2022-12-01 to 2022-12-31
```

Message counts | Serial No, Message, Counts | Daily error counts | Raw Data | Data Summary

```
# A tibble: 41 × 2
   message                                                                    Count
   <fct>                                                                      <int>
 1 "/dev/ttyUSB0. The port is closed. Requesting reboot..."                     453
 2 "/dev/ttyUSB1. The port is closed. Requesting reboot..."                     450
 3 "System.IO.IOException: Remote prematurely closed connection.\n  at Mono.Net.…   182
 4 "An error occurred while reading the key ring."                              158
 5 "System.IO.IOException: Unable to read data from the transport connection: Co…   127
 6 "/dev/ttyUSB1 saw exception during write. System.IO.IOException: No such devi…    96
 7 "/dev/ttyUSB2. The port is closed. Requesting reboot..."                      94
 8 "/dev/ttyUSB0 saw exception during write. System.IO.IOException: No such devi…    91
 9 "/dev/ttyUSB3. The port is closed. Requesting reboot..."                      81
10 "Cannot open modem port. error code:-1"                                       72
11 "System.IO.IOException: Authentication failed because the remote party has cl…    35
12 "MDB log error: mdb poll response new: JUST_RESET (0)"                        17
13 "MDB log error: mdb poll response lost: BEGIN_SESSION (3)"                    13
14 "/dev/ttyUSB3 saw exception during write. System.IO.IOException: No such devi…     8
15 "Unknown IoT command get-config"                                               8
16 "Scheduled job error: System.InvalidOperationException: Collection was modifi…     7
17 "Eftpos terminal not found at 10.0.0.2:20099"                                  6
18 "System.IO.IOException: Remote prematurely closed connection.\n  at Mono.Net.…     6
19 "/dev/ttyUSB2 saw exception during write. System.IO.IOException: No such devi…     5
20 "Error opening serial port '/dev/ttyUSB0'. System.UnauthorizedAccessException…     5
21 "Error opening serial port '/dev/ttyUSB1'. System.UnauthorizedAccessException…     5
22 "Scheduled job error: System.InvalidOperationException: Collection was modifi…     4
23 "MDB log error: mdb poll response lost: VEND_DENIED (6)"                       3
24 "MDB log error: mdb poll response new: END_SESSION (7)"                        3
25 "System.IO.IOException: Unable to write data to the transport connection: Con…     3
26 "*** Transaction Service unhandled exception.  See debug log for details"      2
27 "MDB log error: mdb poll response lost: END_SESSION (7)"                       2
28 "/dev/ttyUSB0 saw exception during write. System.ObjectDisposedException: The…     1
29 "/dev/ttyUSB1 saw exception during write. System.ObjectDisposedException: The…     1
30 "/dev/ttyUSB2 saw exception during write. System.ObjectDisposedException: The…     1
31 "/dev/ttyUSB3 saw exception during write. System.ObjectDisposedException: The…     1
32 "AppDomain unhandled exception: System.NullReferenceException: Object referen…     1
33 "MDB log error: mdb poll response lost: JUST_RESET (0)"                        1
34 "MDB log error: mdb poll response lost: MDB_RESPONSE_VEND_DENIED"              1
35 "MDB log error: mdb poll response lost: VEND_APPROVED (5)"                     1
36 "MDB log error: mdb poll response new: MDB_RESPONSE_END_SESSION"               1
37 "Send to terminal failed: Unable to write data to the transport connection: C…     1
38 "System.IO.IOException: Authentication failed because the remote party has cl…     1
39 "System.IO.IOException: Unable to read data from the transport connection: Co…     1
40 "Total computed payment was 300, but RequiredRemainingVend was 400. Sending R…     1
41 "Unable to connect to payment terminal"                                        1
```

The table above, which shows the frequency of errors reported from 1 December to 31 December, was used to decide on the actions needed to solve the problems caused by the errors. Some errors required machine system updates, and others required machine replacements, among other reasons.

Table 4.1.6. Error counts by date

Pushdev Error Report Summary

DATA | PLOTS

Summaries for the period 2022-12-01 to 2022-12-31

Message counts | Serial No, Message, Counts | Daily error counts | Raw Data | Data Summary

Copy | CSV | Excel | Print | PDF

Search:

	DATE	Number of Errors
1	2022-12-31	36
2	2022-12-30	28
3	2022-12-29	23
4	2022-12-28	33
5	2022-12-27	36
6	2022-12-26	30
7	2022-12-25	18
8	2022-12-24	29
9	2022-12-23	26
10	2022-12-22	43

Showing 1 to 10 of 31 entries

Previous 1 2 3 4 Next

This table shows the total number of error messages recorded each day from the 1st to the 31st of December 2022. The users would know when most errors were reported.

Figure 4.1.11. Monthly sales visualisations using Power BI

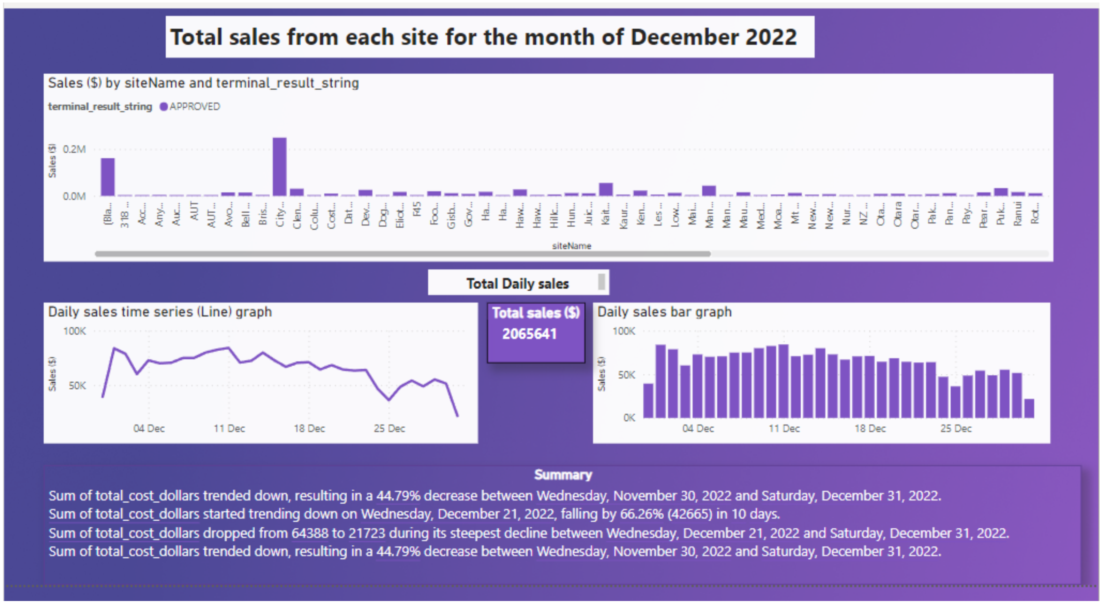


Figure 4.1.11 shows three plots and a summary. The plots were interactive, so choosing a site in the top bar graph resulted in the line and bar charts underneath to give the plots for the selected site. Likewise, the summary would also give the transaction summary for the selected site, and the total sales changed to give the amount collected at the selected site.
The visualisations above are similar to those in Figure 4.1.2, Figure 4.1.3 and Figure 4.1.4

4.2. Preferred reporting platform:

Shiny was better than Power BI for the company's data visualisations and presentations as it housed all the reports together and could be used to generate reports for any period at any time. Graphs and tables that required a lot of filtering and coding were more straightforward to create in Shiny than in Power BI.
It was easy to clean the data in Shiny, whereas in Power BI, transforming the data while working in DirectQuery mode was impossible.

5. Conclusions/Discussion:

5.1. Authenticity of the summaries and visualisations:

The summaries and visualisations generated in the Shiny app were cross-checked against those in the company's payments cloud web app, and they all proved authentic and accurate.

When transactions were only recorded in a single day, the Shiny bar plots were better than those generated in the company web app. An example is shown in the two plots in Figure 5.1.1 (plot created in the company web app) and Figure 5.1.2 (plot created in Shiny app) below.

Figure 5.1.1. Sales for December 2022 at the site Dogwash from the company web app.

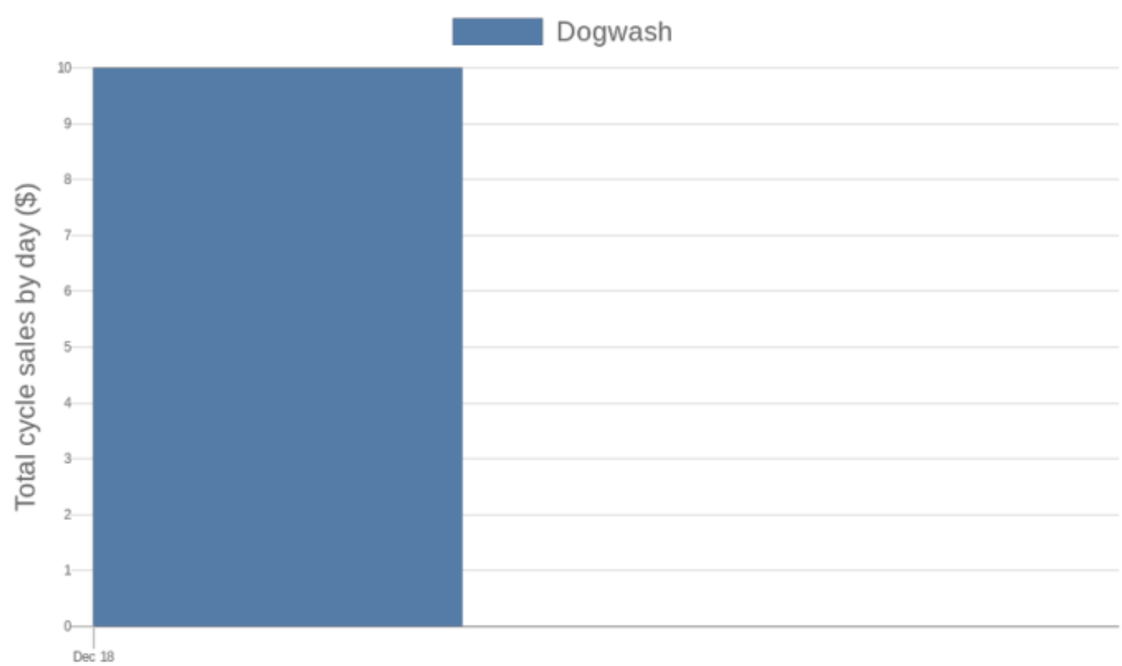
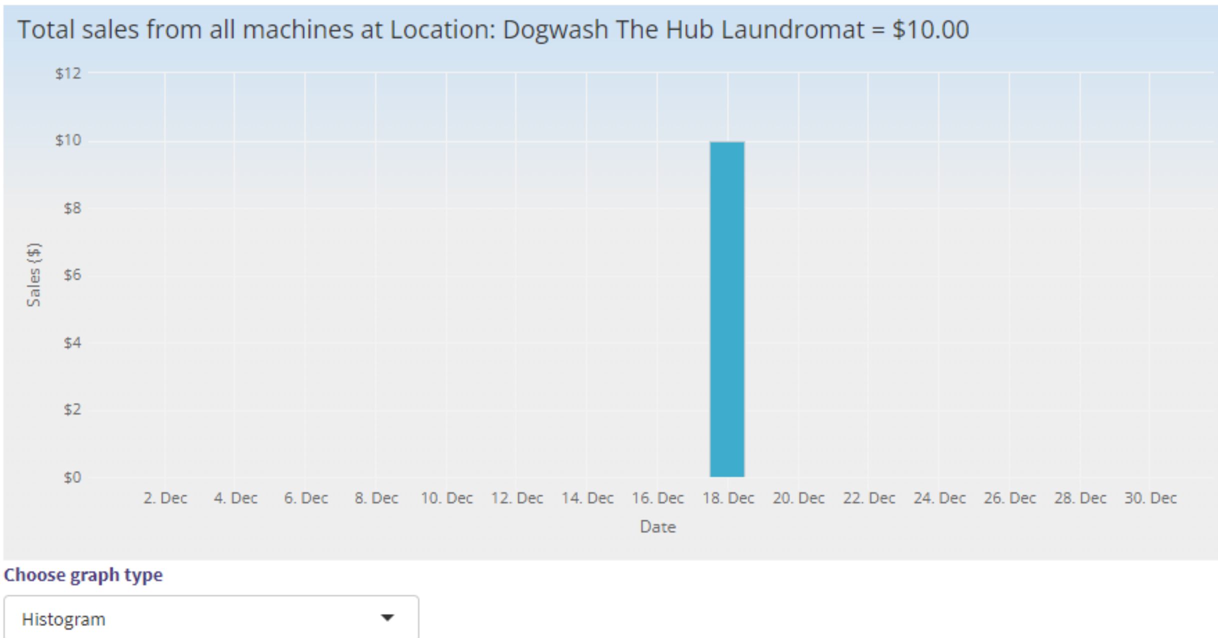


Figure 5.1.2. Sales for December 2022 at site Dogwash from the Shiny app.



5.2. Future work:

The two Shiny applications should be merged into one app, giving users choices of data they want to use.
Combining the two applications will make it easy if more datasets are added, as everything will be under one application.

The app can be embellished with tab icons and company logos.

The deployment of the app needs to be resolved. Currently, users will have the app and RStudio installed on their computers. This should be hosted on a Shiny server (such as shinyapps.io) so that no local software installation is necessary.

Time series analysis can be used to predict machine errors and sales.

5.3. Problems:

Using date ranges in the data filter proved difficult because Dates are not UTC-corrected. In March and September, an assessment of whether the application handles daylight saving time changes should be made.

6. References:

1. **Chang, Winston, et al. "Package ‘shiny’."** ***See http://citeseerx. ist. psu. edu/viewdoc/download*** **(2015).**
2. **Becker, Louis T., and Elyssa M. Gould. "Microsoft Power BI: extending excel to manipulate, analyze, and visualize diverse data."** ***Serials Review*** **45.3 (2019): 184-188.**
3. **Negrut, Viorel. "Power bi: Effective data aggregation."** ***Quaestus*** **13 (2018): 146-152.**
4. **Hyndman, Rob J., and George Athanasopoulos.** ***Forecasting: principles and practice*****. OTexts, 2018.**
5. **http://shiny.rstudio.com/tutorial/**
6. **https://www.highcharts.com/blog/tutorials/highcharts-for-r-users/**
7. **https://shiny.rstudio.com/reference/shiny/1.7.4/**

7. Appendices:

Appendix 1. Time series decomposition for the error messages

Pushdev Error Report Summary for DATA 601

Date Selection | DATA | PLOTS

```
Visualisations for the period 2022-12-01 to 2022-12-31
```

Daily Error Counts per Machine | Messages plot | Serial numbers vs Messages Bar plot | Time Series Decomposition

Choose the decomposition:

messages

Choose Error message:

/dev/ttyUSB3. The port is closed. Requestin

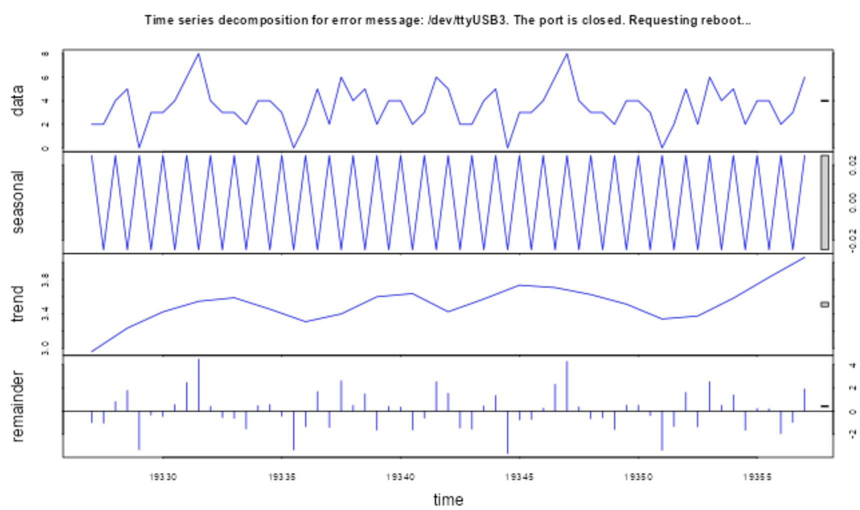
Select Serial Number:

GWB827EB9C94B3

Choose trend window

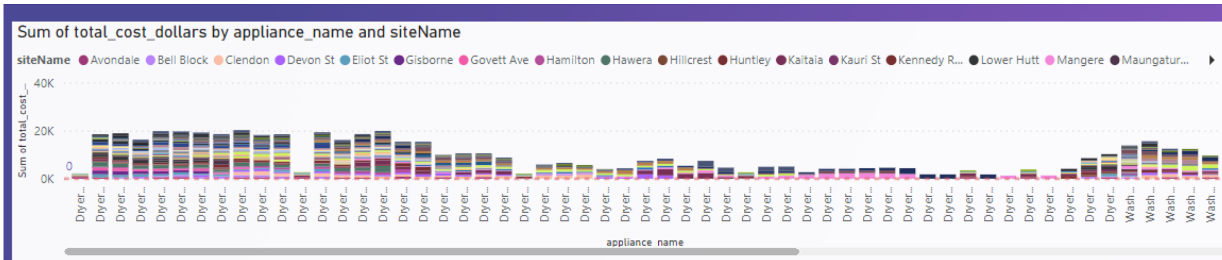
1 22 30

1 4 7 10 13 16 19 22 25 28 30




```
The daily sales for the error have been decomposed to show
the actual plot (data),
the seasonal component,
the trend component and
the remainder (residuals) component.

Use the slider above to smoothen the trend.
```

Appendix 2. Sales from individual appliances using Power BI



siteName	appliance_name	Sum of total_cost_dollars
Whangarei	Dryer 12	913
Whangarei	Dryer 13	990
Whangarei	Dryer 14	712
Whangarei	Dryer 15	642
Whangarei	Dryer 16	681
Whangarei	Dryer 17	905
Whangarei	Dryer 18	433
Whangarei	Dryer 19	610
Whangarei	Dryer 20	584
Total		**1308723**

Appendix 3. Line graph of the weekly sales

Transaction Data for DATA 601

Date Selection | DATA | PLOTS

```
Visualisations for the period 2022-12-01 to 2022-12-31
```

Terminal result string summary and plot | Total sales from each site | Daily & Weekly Time Series Graphs of Sales

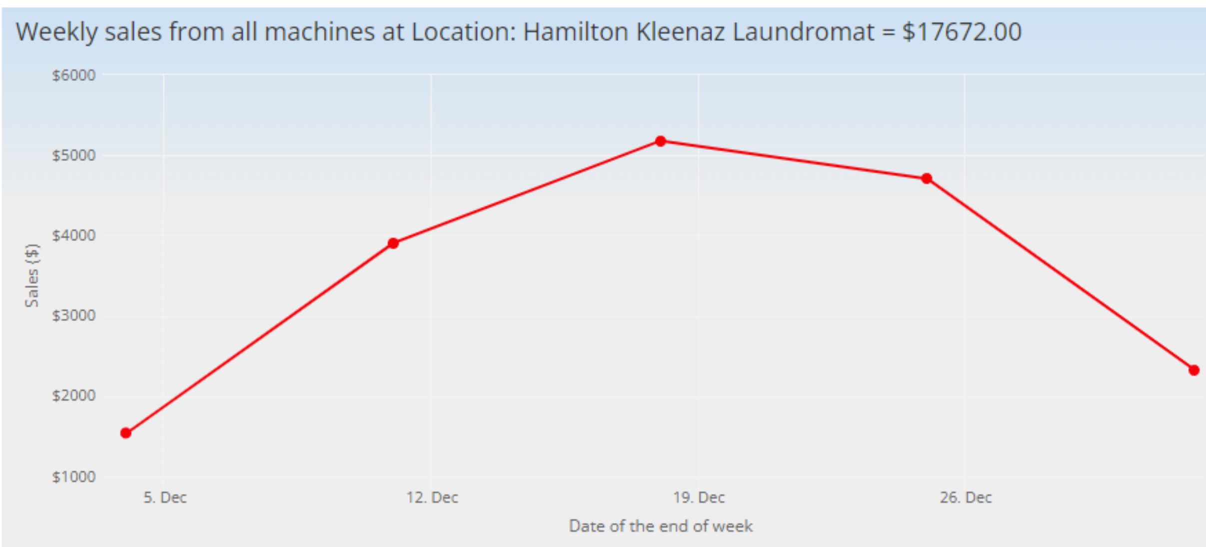
Time Series Decomposition | All appliances and all machine sales | Sales Bubble plot

Choose Sitename or type name:

Hamilton Kleenaz Laundromat

Choose the period:

Weekly



Choose graph type

Line graph

```
This plot shows the daily and weekly sales for Hamilton Kleenaz Laundromat.
Note the peaks and troughs for each site to see if they are following a cycle.
As a company, we need to think of strategies we can use to increase the
sales on those days where sales are usually low and to get more sales on
days where sales are usually high.

Possible strategies would be:
1. Give discounted prices on the days where sales are usually low.
2. Create raffles where a minimum amount is required to enter.
3. Increase the price on those days where sales are high.
The line graph allows us to easily see the trend for the sales, especially
over longer periods.
```

Appendix 4. Histogram of the weekly sales

Transaction Data for DATA 601

Date Selection | DATA | PLOTS

```
Visualisations for the period 2022-12-01 to 2022-12-31
```

Terminal result string summary and plot | Total sales from each site | Daily & Weekly Time Series Graphs of Sales

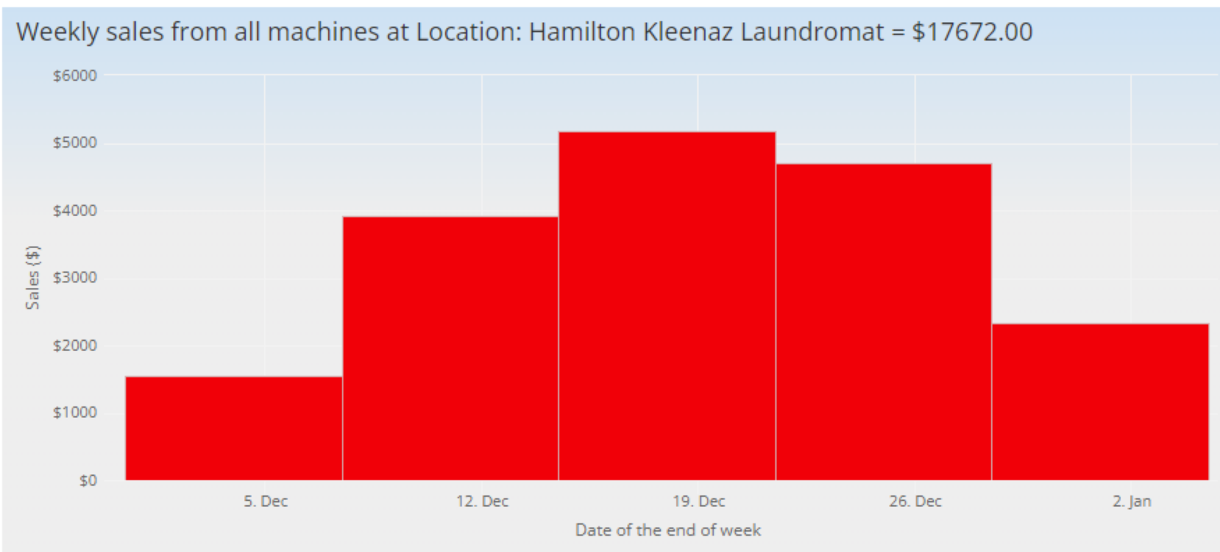
Time Series Decomposition | All appliances and all machine sales | Sales Bubble plot

Choose Sitename or type name:

Hamilton Kleenaz Laundromat

Choose the period:

Weekly



Choose graph type

Histogram

```
This plot shows the daily and weekly sales for Hamilton Kleenaz Laundromat.
Note the peaks and troughs for each site to see if they are following a cycle.
As a company, we need to think of strategies we can use to increase the
sales on those days where sales are usually low and to get more sales on
days where sales are usually high.

Possible strategies would be:
1. Give discounted prices on the days where sales are usually low.
2. Create raffles where a minimum amount is required to enter.
3. Increase the price on those days where sales are high.
The line graph allows us to easily see the trend for the sales, especially
over longer periods.
```

Appendix 5. Effects of Covid-19 on transaction data sales

The two charts below show how Covid-19 affected sales at the site Juice Placeholder when the country went into a lockdown. This dataset had 6 788 798 records of data. The data was for the period from January 2021 up to December 2022.

Transaction Data for DATA 601

Date Selection | DATA | PLOTS

```
Visualisations for the period 2021-01-01 to 2022-12-31
```

Terminal result string summary and plot | Total sales from each site | Daily & Weekly Time Series Graphs of Sales

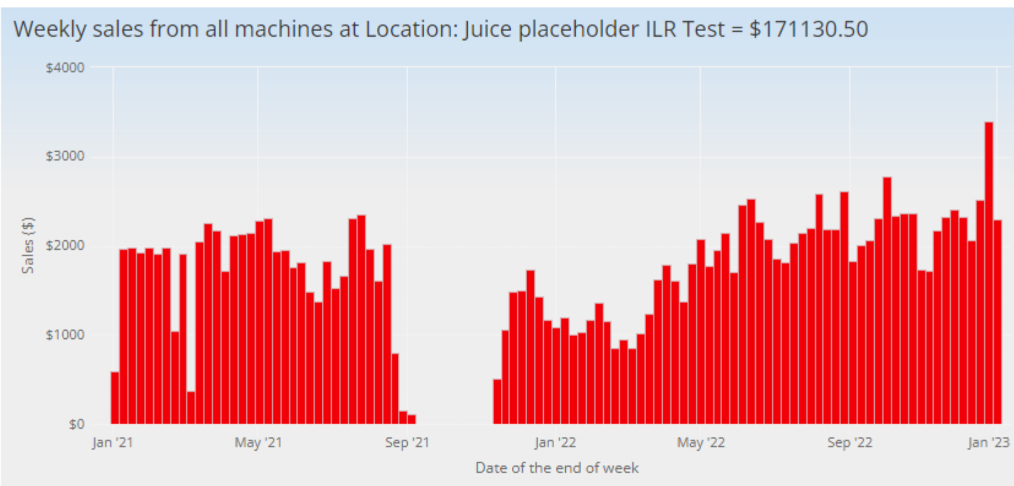
Time Series Decomposition | All appliances and all machine sales | Sales Bubble plot

Choose Sitename or type name:

Juice placeholder ILR Test

Choose the period:

Weekly



Choose graph type

Histogram

```
This plot shows the daily and weekly sales for Juice placeholder ILR Test.
Note the peaks and troughs for each site to see if they are following a cycle.
As a company, we need to think of strategies we can use to increase the
sales on those days where sales are usually low and to get more sales on
days where sales are usually high.

Possible strategies would be:
1. Give discounted prices on the days where sales are usually low.
2. Create raffles where a minimum amount is required to enter.
3. Increase the price on those days where sales are high.
The line graph allows us to easily see the trend for the sales, especially
over longer periods.
```

Transaction Data for DATA 601

Date Selection | DATA | PLOTS

```
Visualisations for the period 2021-01-01 to 2022-12-31
```

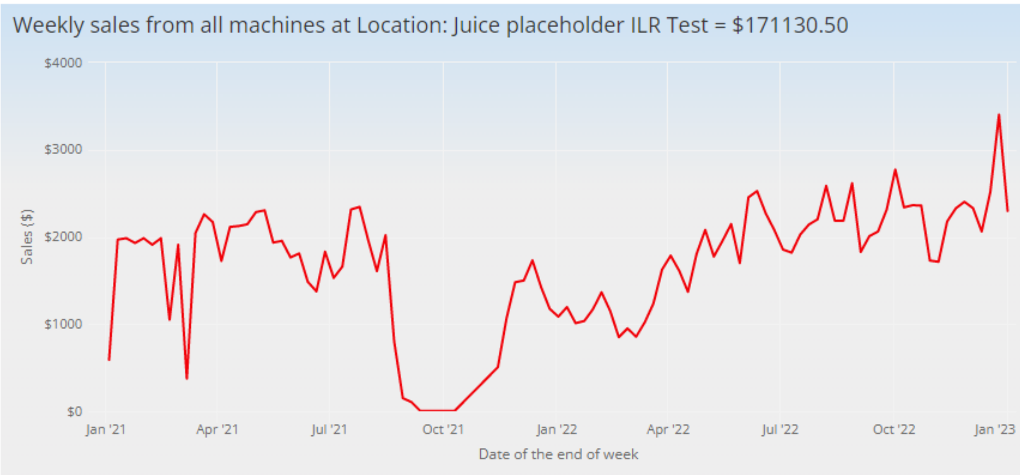
Terminal result string summary and plot | Total sales from each site | Daily & Weekly Time Series Graphs of Sales | Time Series Decomposition | All appliances and all machine sales | Sales Bubble plot

Choose Sitename or type name:

Juice placeholder ILR Test

Choose the period:

Weekly



Choose graph type

Line graph

```
This plot shows the daily and weekly sales for Juice placeholder ILR Test.
Note the peaks and troughs for each site to see if they are following a cycle.
As a company, we need to think of strategies we can use to increase the
sales on those days where sales are usually low and to get more sales on
days where sales are usually high.

Possible strategies would be:
1. Give discounted prices on the days where sales are usually low.
2. Create raffles where a minimum amount is required to enter.
3. Increase the price on those days where sales are high.
The line graph allows us to easily see the trend for the sales, especially
over longer periods.
```