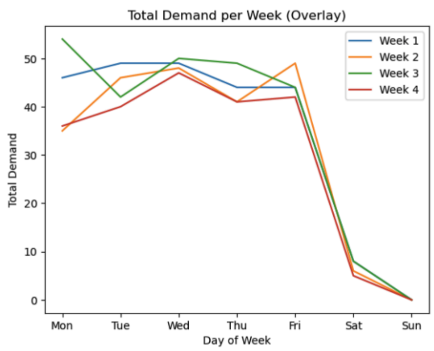
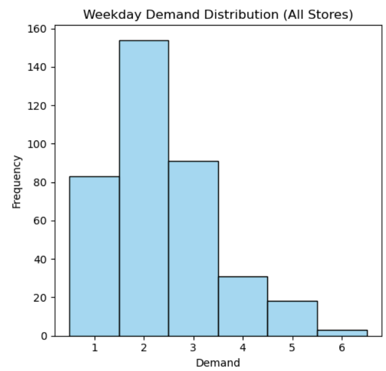
Group Model Report

Demand Estimates

When creating sufficient estimates for our routes, several factors needed to be considered. From the initial visualisation of the data, two key aspects were evident: the demands of Woolworths stores were stochastic, and a split existed between weekday, Saturday, and Sunday demands.





As this is an initial, rough mock-up of the final product, we were allowed to make a few assumptions and simplifications to aid the estimation process. The first was assuming that the demand can be modelled using a normal distribution. From the two plots on the right, we can see that they're closer to gamma/poisson distributions, but a normal distribution allows for easy calculation of percentiles. This assumption also has sub-assumptions, such as individual store daily demands being independent not only from each other, but from past days too.

The second assumption is that the demand distribution on each weekday is the same; there is no change in either the mean or the standard deviation, whether it is a Tuesday or a Friday. There is one weekend assumption: if a store has no demand on all 4 weeks of data, the store is assumed to be closed, and thus has zero demand.

As a simplification, we are using a single value for each store's demand. Using the mean/median would lead to our vans being understocked 50% of the time. To counteract this, we used the 75th percentile value, skewing our van to be overstocked more than understocked, a less-penalised position. However, this value can be easily adjusted, depending on how risk-averse we want to be.

We kept our demand values unrounded to better represent the underlying stochastic process. This approach ensures that when demands are no longer fixed later in the project, an estimate of 8.75 reflects a realistic distribution, where approximately 75% of the time the demand will be 9, and 25% of the time it will be 8. Once demands become fixed, they will be integer values, so fractional boxes will no longer need to be considered.

Route Generation

To generate the routes and eventually the linear program, I implemented object-oriented programming by creating a route class. This object holds attributes such as the stores in the route, the travel time, the day of the week, and the total demand. These attributes are all generated using self-defined methods that update automatically when a store is added.

Using these objects, I created a recursive function that finds all feasible routes. On every iteration, it completes the route provided to the function (returns to CentrePort) and appends it to our list of routes. It then loops through every feasible, unvisited store and calls itself again for each.

Feasibility was defined as having a total demand that falls within our van capacity of 9 boxes and having a travel time of under 3 hours, excluding unloading times.

With this, I generated approximately 3,000 routes with weekday demands and 20,000 routes with Saturday demands. This took too long to compute, which would later affect the linear program's performance. To mitigate this effect, I introduced a check when completing the route. If the route being added visits the same stores as an already existing route, compare the two and take the route with the shorter travel time. This would result in a final set of 598 weekday routes and 3040 weekend routes.

For our weekday routes, the limiting factor was the van capacity. Routes would commonly hit 9 boxes before reaching 3 hours. This differed from our Saturday routes, where the travel time was by far the limiting factor. The maximum travel time of any Saturday route was 10799 seconds (2.99 hours), and he maximum demand of any Saturday route was only 6.4 boxes.

Formulation

Discussion:

After generating the routes, we formulated the route selection as a mixed-integer linear program. This model calculates overtime, maximum demand, fleet size, and which entity will carry out each route. These components are then used to determine and minimise the objective function, which represents the total cost over a single weekday and a Saturday combined. In the objective function, time (in seconds) and overtime are used to calculate the cost when a van is selected. If a van is not selected, the cost is instead derived from the subcontractor, who charges in four-hour blocks. The fleet size also influences the proportion of annual maintenance costs that must be budgeted for the two days.

Definitions:

Switch Variables

x_r - 1 if route r selected, 0 otherwise.

w_r - linearising variable - Subcontracting cost of a selected route.

z_r - binary switch variable - 1 if Van used AND route selected, 0 otherwise.

Decision Variables

van_r - 1 if using a van on route r, 0 otherwise.

Sub_r - The amount of 4-hour blocks route r needs to be charged for.

$overtime_r$ - The amount of overtime route r needs to be charged for.

$Fleet$ - Number of vans in the fleet

Route Data

$Demand_r$ - Demand estimate over the course of route r.

$time_r$ - the time route r takes, in seconds.

$Visited_{s,r}$ - 1 if route r visits store s, 0 otherwise

M - Large number

Sets:

{Routes} - All possible routes

{Week} - All weekday routes

{Sat} - All Saturday routes

Objective Function:

$$MIN: \sum_{r \in \{Routes\}} [z_r \times \frac{200}{3600} time_r + \frac{75}{3600} overtime_r + 1000 w_r] + \frac{2 \times 50000}{\frac{6}{7} \times 365} \text{Fleet}$$

Constraints:

Overtime:

$\forall r \in \{Routes\}$: $overtime_r \leq Mx_r$ - *Large UB if selected, 0 otherwise.*

$overtime_r \geq t_r - 10800 - M(1 - x_r)$ - *LB of t-3 if selected*

Sub60:

$\forall r \in \{Routes\}$: $Sub_r \geq \frac{time_r}{4*3600} - M * van_r$ - if van not selected, subcontractor charged, in 4-hour blocks

Demand:

$\forall r \in \{Routes\}$: $demand_r \leq 4 + 5van_r$ - If van is selected, demand can be 9; otherwise, it has to be below 4

Vans:

Only 2 shifts can be run a day. On each shift, the number of vans used must be less than fleet size.

$$\sum_{r \in \{Week\}} z_r \leq 2 \times Fleet$$

$$\sum_{r \in \{Sat\}} z_r \leq 2 \times Fleet$$

Linearise Subcontractor Costs:

$\forall r \in \{Routes\}$: $w_r \leq Sub_r$

$w_r \leq Mx_r$

$w_r \geq Sub_r - M(1 - x_r)$

Linearise Vans:

$\forall r \in \{Routes\}$: $z_r \leq x_r$

$z_r \leq van_r$

$z_r \geq x_r + van_r - 1$

One visit per store:

For all stores, the sum of routes visiting on a certain day must be 1

$$\forall s \in \{Stores\}: \sum_{r \in \{Week\}} visited_{s,r} \times x_r = 1$$

$$\sum_{r \in \{Sat\}} visited_{s,r} \times x_r = 1$$

Non-negativity/Others

$x_r,\ z_r, van_r,\ visited_{s,r} \in \{0, 1\},\ binary$

$w_r,\ sub_r,\ overtime_r,\ Fleet,\ Demand_r,\ time_r \geq 0,\ integer$

$Fleet \leq 8$

$M = 9999999$

Estimation

When solving the linear programming problem on Python, our initial weekday-only run took only 0.15 seconds to find a solution. However, when Saturdays were introduced into the mix, the solver ran for over 3 hours. I eventually introduced a 30-minute time limit, taking the best answer at that point. This was later checked with GUROBI as the optimal answer. The returned answer is provided below. CentrePort Wellington has been manually removed from each route for space-saving purposes; however, it was included in the calculations.

Status: Optimal
Total Cost = 6837.838152431071 ≈ $6837.84
Fleet = 4

Selected Weekday routes:

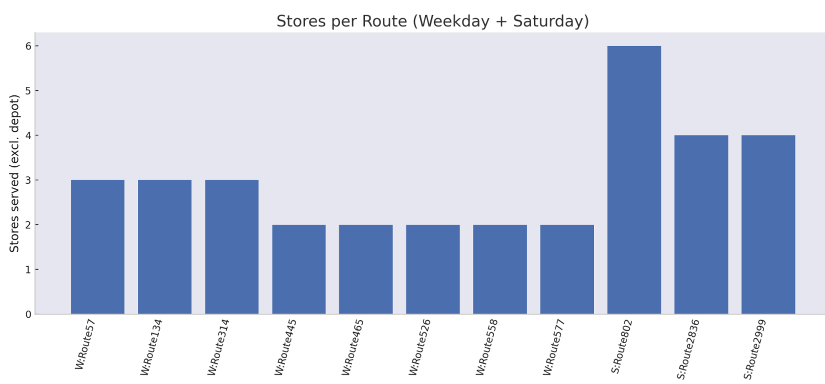
Route57 -> ['Woolworths Aotea', 'FreshChoice Cannons Creek', 'Woolworths Tawa'],	Demand: 8.272, Time: 10,497.46s
Route134 -> ['Metro Cable Car Lane', 'FreshChoice Cuba Street', 'Woolworths Karori'],	Demand: 7.74, Time: 8767.69s
Route314 -> ['FreshChoice Woburn', 'Woolworths Queensgate', 'Woolworths Lower Hutt'],	Demand: 8.1, Time: 9704.31s
Route445 -> ['Woolworths Johnsonville Mall', 'Woolworths Crofton Downs'],	Demand: 6.71, Time: 8043.89s
Route465 -> ['Woolworths Porirua', 'Woolworths Johnsonville'],	Demand: 7.47, Time: 9924.81s
Route526 -> ['Woolworths Kilbirnie', 'Woolworths Newtown'],	Demand: 6.64, Time: 7687.26s
Route558 -> ['Woolworths Upper Hutt', 'Woolworths Maidstone'],	Demand: 6.68, Time: 10534.15s
Route577 -> ['Woolworths Petone', 'Woolworths Wainuiomata'],	Demand: 6.61, Time: 9460.71s

Selected Saturday routes:

Route802 -> ['Woolworths Kilbirnie', 'Woolworths Newtown', 'Woolworths Karori', 'Woolworths Crofton Downs', 'Woolworths Johnsonville Mall', 'Woolworths Johnsonville'], Demand: 4.33, Time: 7952.89s

Route2836 -> ['Woolworths Petone', 'Woolworths Wainuiomata', 'Woolworths Queensgate', 'Woolworths Lower Hutt'], Demand: 4.34, Time: 7582.94s

Route2999 -> ['Woolworths Upper Hutt', 'Woolworths Aotea', 'Woolworths Porirua', 'Woolworths Tawa'], Demand: 4.2, Time: 9911.24s



Weekdays:
8 routes, 19 total store visits

Saturdays:
3 routes, 14 total store visits
(5 Stores closed)

On Saturdays, we are running fewer routes compared to weekdays. This is due to the lower demands, which shift the binding constraint from demand to travel time. This means, as seen above, that each route can service more stores before hitting any limit. We found that zero routes utilised subcontractors, which means that, on weekdays, it is necessary to maintain a fleet size of 4 vans and use all of them. This indicates that the added maintenance costs of a van outweigh the additional cost of subcontractors.