Benchmarking Using Data Envelopment Analysis: Application to Stores of a Post and Banking Business

Andrea Raith[a], Paul Rouse[b], Lawrence M. Seiford[c]

1 INTRODUCTION

Data Envelopment Analysis (DEA) is a non-parametric, optimisation-based benchmarking technique first introduced by (Charnes, et al., 1978), later extended by (Banker, et al., 1984), with many variations of DEA models proposed since. DEA measures the production efficiency of a so-called Decision Making Unit (DMU) which consumes inputs to produce outputs. DEA can be a particularly useful tool of analysis when there is an abundance of measures to be analysed in terms of DMU performance, allowing it to benchmark and identify comparable peers. DEA is capable of capturing multi-dimensional activities of complex DMUs (or organisations) and is particularly useful when investigating the effects of contextual or environmental factors on organisations' performance. DEA has been applied in numerous areas including banking, education, health, transport, justice, retail stores, auditing, fighter jet design, research and development to name a few. Growth in the use of DEA since its inception in 1978 has been rapid with the total number of journal articles reaching 10,300 and distinct authors 11,975 by the end of 2016 (Emrouznejad & Yang, 2017).

DEA is based around a production model which assesses the efficiency of DMUs in turning inputs into outputs. This is done by comparing units with each other to identify a frontier of best performance defined by so-called efficient DMUs, which non-efficient DMUs are benchmarked against. This efficient frontier represents "achieved best performance" based on actual outputs produced and inputs consumed. The efficient frontier thus provides a useful reference set for benchmarking and performance improvement. Most DEA models assume convexity but there are non-convex variations such as the Free Disposal Hull models (Leleu, 2006) which do not. There are very few assumptions required in DEA and its non-parametric form avoids the need to consider alternative distribution properties. Although often described as deterministic, there has been considerable work in recent years in investigating the stochastic properties of DEA estimates (Olesen & Petersen, 2016).

[a] Corresponding Author, Department of Engineering Science, University of Auckland, New Zealand. Phone +64 9 3737599 ext 81977, email: a.raith@auckland.ac.nz
[b] Department of Accounting and Finance, Faculty of Business and Economics, University of Auckland, New Zealand.
[c] Industrial & Operations Engineering, University of Michigan, USA.

Learning Outcomes of this Chapter

- Develop an intuitive understanding of DEA
- Understand basic linear programming models for DEA
- Be aware of common DEA modelling techniques
- Be able to conduct a DEA analysis supported by `pyDEA` software
- Be able to interpret the DEA results and explain them to a non-technical audience

In the following we will first describe the case of a Post and Banking Business, and then introduce DEA in the context of our case. Different DEA models and additional features are discussed. We give a brief outline of an open-source software tool for DEA and finally apply three different DEA models to the case study and discuss the results.

2 CASE STUDY

We consider the case of a postal service which also owns and operates a banking business. New Zealand Post and its partly-owned subsidiary Kiwibank Limited are only one example of such a mode of operation, and Deutsche Postbank used to be an example of this in the past (although they are no longer owned by the associated postal service)[d]. We focus on the operation of retail stores which all provide postal services, and some of which also offer banking services (Priddey & Harton, 2010).

Postal retail stores provide a large range of services such as postal services, bill payments, banking services, travel bookings and insurance. Stores operate throughout the country under varying environmental circumstances, such as varying population both in size and socio-economic background, different levels of competition and location, which can be rural, satellite urban (outer suburbs of a large urban centre) or urban. Stores are split into three groups depending on what types of banking functions they provide. Banking functions include everyday banking, credits cards, loans, home loans, insurance, etc[d]. In our dataset there are 48 stores with full banking functions, four with limited banking functions, and 91 that do not provide extended banking functions.

Managing this large range of stores, often under unique circumstances, means it is challenging to accurately and fairly capture the performance of each individual store, and to set fair performance targets for stores.

To be able to assess store performance, metrics that capture the operations of each store need to be defined. NZ Post had an established performance measurement system and metrics that they regarded

[d] www.kiwibank.co.nz and https://en.wikipedia.org/wiki/Kiwibank; www.postbank.com and https://en.wikipedia.org/wiki/Deutsche_Postbank

as important. These metrics were collected and reported internally by their stores and they formed the database that was provided to us. Possible metrics for postal stores are:

- Staffing: Number of employees (full time equivalent (FTE)), staff engagement
- Facilities: store floor area and rent.
- Operating environment: median income in the area, competition, location. Here, we classify all stores according to their location (main urban, satellite urban, and rural); the number of other post stores within a radius of 5km.
- Customer satisfaction: a score between 0 and 1 derived from the experience of mystery shoppers at each store.
- Expenditure: different expenditure streams (personnel cost and other expenditure).
- Revenue: different revenue streams (distinguishing banking revenue, retail, post and bill pay revenue), and the number of sales lines per customer. For instance, different sales lines could be the sale of stamps to a customer, paying for a car's annual registration, or signing up for a loan or a mortgage.

In the following we will derive a DEA model to be used in the assessment of store performance and to determine fair annual performance targets for the management of individual stores. Sample data is shown in Table 1. We note that throughout this chapter units and timeframes are deliberately omitted.

Table 1: Input data

	Expenditure			Revenue							
Store	Total	Personnel	Other	Total	Banking	Postal	Billpay	Retail	SL	CS	Loc
A	364,814	260,142	104,672	209,088	31,106	60,968	45,225	71,789	1.45	0.85	urban
B	451,598	354,155	97,443	275,924	125,342	47,286	32,071	71,226	1.41	0.79	urban
C	394,370	279,537	114,833	296,295	63,645	52,679	104,955	75,016	1.39	0.83	urban
D	568,372	488,526	79,846	437,407	127,474	100,725	71,650	137,558	1.45	0.77	urban
E	880,434	614,098	266,336	597,780	41,387	152,145	155,354	248,894	1.42	0.90	rural
F	800,319	580,788	219,531	697,707	83,067	183,140	199,802	231,698	1.40	0.87	urban
G	1,041,821	780,228	261,593	724,544	129,514	159,586	197,259	238,185	1.37	0.81	urban

SL: Average number of Sales Lines per customer; CS: Customer Satisfaction Score; Loc: Location

Expenditure is disaggregated into two types: personnel and other where the latter includes occupancy costs, utilities and depreciation. There are four revenue streams corresponding to services for banking, postal, bill payments and retail sales. Table 1 also shows the average number of sales lines per customer (SL), customer satisfaction score (CS) and the type of location (Loc).

3 INTRODUCTION TO DEA

DEA measures the production efficiency of a so-called Decision Making Unit (DMU) which consumes inputs to produce outputs. In the context of our case, a DMU is a post store (with or without banking functions). DEA estimates a non-parametric production frontier which determines the relative efficiency of production of the individual DMUs based on linear programming. Figure 9 later in the chapter illustrates the production process for a potential set of two inputs and five outputs where each post store is a DMU, and so-called environmental factors are also considered (more on environmental factors later).

For simplicity of presentation we will measure store efficiency initially with a single input and a single output. Efficiency will be assessed in terms of converting total expenditure (input) into total revenue (output), both shown in Table 1 and Figure 1.

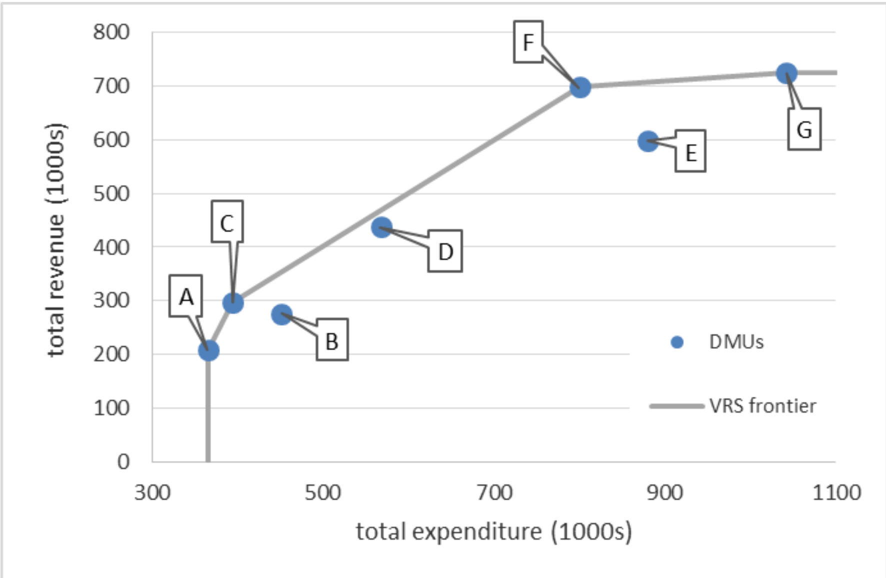


Figure 1: Plot of input and output

Figure 1 depicts seven DMUs consuming a single input (total expenditure - horizontal axis) to produce a single output (total revenue - vertical axis). The expectation is that higher levels of input lead to higher levels of output. Considering DMU B consumes a higher level of input than DMU C, but produces a lower level of output than DMU C, B is considered to be *dominated* by C. Likewise, DMU E is dominated by DMU F. DMUs A, C, F and G are efficient DMUs which determine the efficient frontier. The other DMUs are inefficient. We note that in principle the concept of dominance and (in-) efficiency is analogous to that in multi-criteria optimisation, where a point is called efficient (or non-dominated) if there does not exist another feasible point which dominates it. However, in conventional DEA models a DMU is also considered inefficient if it is dominated by the convex combination of efficient DMUs. The convex combination of two points z^1, z^2 is any point $z = \rho z^1 +$

$(1-\rho)z^2$ with $\rho \in [0,1]$. An example is DMU D which is not dominated by another DMU, but it is dominated by a convex combination of DMUs C and F.

In an output-oriented assessment of performance in-efficient DMU E should be able to increase its output while maintaining the same level of input. The maximum amount by which output could be increased is limited by the production frontier, determined by the convex hull of the efficient DMUs. Efficient DMUs F and G form the segment to which DMU E is projecting and thus are appropriate examples of best practice for E to benchmark against. E could discuss its results with these peer units to identify areas where it could improve its operations. Figure 2 illustrates this so-called variable returns to scale (VRS) frontier, as explained later, and indicates the output level DMU E has to achieve to become efficient in an output orientation (solid arrow).

In an input-orientation, performance assessment aims to reduce the input while maintaining output at its current level (for inefficient DMU E this is shown as the dotted arrow in Figure 2). Inefficient DMUs B and D can similarly improve their input or output levels depending on the orientation. DMU D would be considered more efficient than B and E in Figure 2 as it is closer to the frontier than the other two inefficient DMUs. In a managerial context DMU efficiency can help identify benchmarks and best practice. For instance, efficient DMUs C and F form the segment to which DMU E is projecting and thus are appropriate examples of best practice for E to benchmark against. E could discuss its results with these peer units to identify areas where it could improve its operations.

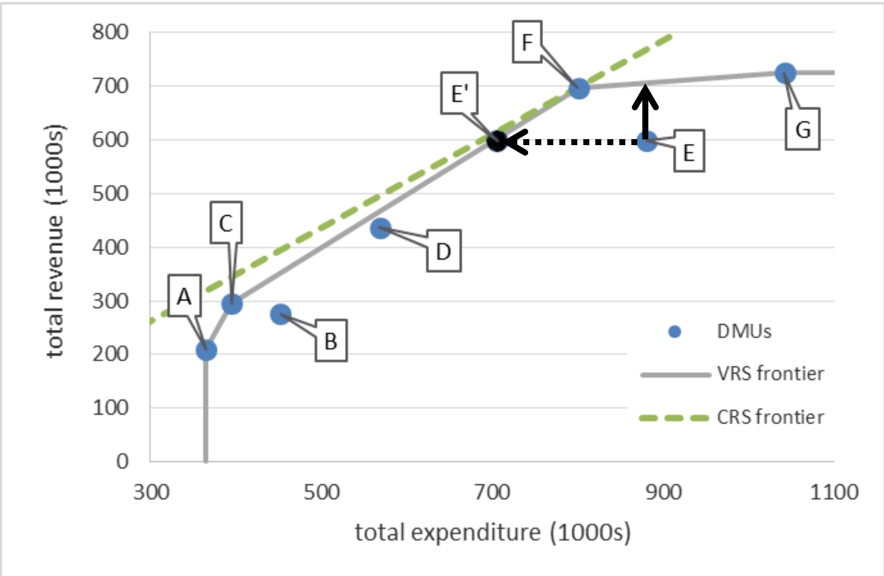


Figure 2: DEA Frontier illustration

Different economies of scale can also be modelled in DEA. Constant returns to scale (CRS) implies that if a DMU increases expenditure (input) by say 10 percent, a corresponding 10 percent increase of revenue (output) is observed. It also refers to the highest ratio of output to input, i.e. average productivity. If the increase in revenue is more than the 10 percent increase in input, this is called increasing returns to scale (IRS) and if the increase in revenue is less than 10 percent, it is known as decreasing returns to scale (DRS). In our simple example, the most scale efficient DMU(s) are those with a maximum ratio of input over output, i.e. DMU F. The CRS DEA frontier is shown as the dotted line in Figure 2, a ray passing through the origin. The assumption of variable returns to scale (VRS) allows for all three types of scale, increasing, constant and decreasing, with varying proportional increase of output at different input levels. The VRS frontier is also shown in Figure 2 and comprises DMUs A and C (IRS), F (CRS) and G (DRS).

4 LINEAR PROGRAMMING MODELS FOR DEA

An intuitive initial DEA model begins with the ratio form of a DEA problem, see for instance Chapter 6 in (Coelli, et al., 2005) or Chapter 2 in (Cooper, et al., 2007). This is also known as the CCR model after the authors (Charnes, et al., 1978) who first proposed it. We assume there are DMUs $i = 1,2, \dots, d$ each with n inputs and m outputs. The input vector of DMU i is $x_i \in \mathbb{R}^n$ and its output vector is $y_i \in \mathbb{R}^m$. The ratio form of an input-oriented CRS DEA model maximises the ratio of the weighted outputs over the weighted inputs for each DMU i while requiring that all other DMUs' ratios are at most 1 using the same weights. Weight vectors $u \in \mathbb{R}_+^n$ and $w \in \mathbb{R}_+^m$ of inputs and outputs are to be determined. This ensures that each DMU is evaluated under its most favourable weights and then compared to other DMUs' performance under this same set of weights. The ratio problem for DMU i is:

$$
\begin{aligned}
max \quad & \frac{w^T y_i}{u^T x_i} && (1) \\
s.t. \quad & \frac{w^T y_j}{u^T x_j} \leq 1 \qquad \forall\, j = 1,2, \dots, d \\
& u, w \geq 0.
\end{aligned}
$$

By setting the denominator of (1) to 1 and introducing some new notation (Charnes & Cooper, 1962), the following DEA linear programming problem (LP) with unique optimal objective function value is obtained, with input and output weights now denoted $\mu \in \mathbb{R}_+^n$ and $\nu \in \mathbb{R}_+^m$.

$$
\begin{aligned}
max \quad & \nu^T y_i && (2) \\
s.t. \quad & \mu^T x_i = 1 \\
& \nu^T y_j - \mu^T x_j \leq 0 \qquad \forall\, j = 1,2, \dots, d \\
& \mu, \nu \geq 0.
\end{aligned}
$$

LP (2) is known as the *multiplier form* of the CRS DEA model. Linear programming theory associates a so-called dual problem with each (primal) linear programme. The dual form of LP (2) is known as the *envelopment form* of the CRS DEA model for DMU i:

$$\begin{aligned} \min \quad & \theta \qquad\qquad (3) \\ s.t. \quad & -y_i + Y\lambda \geq 0 \\ & \theta x_i - X\lambda \geq 0 \\ & \theta, \lambda \geq 0. \end{aligned}$$

Here, decision variables are θ, the scalar representing efficiency of DMU i, and $\lambda \in \mathbb{R}^d$ which is a vector of peer weights attached to each DMU. $X \in \mathbb{R}^{n \times d}$ is the matrix of inputs, and similarly, $Y \in \mathbb{R}^{m \times d}$ is the matrix of outputs. This form of the model aims to reduce the input of DMU i radially by contracting the input to θx_i to obtain a feasible point that lies on the production frontier $(X\lambda, Y\lambda)$. This corresponds to the projection of inefficient point E to a point on the CRS frontier in Figure 2. We call the computed point $(X\lambda, Y\lambda)$ target point for each DMU i, noting that point $(X\lambda, Y\lambda)$ is not a real DMU but a composite DMU made up of a linear combination of DMUs that define best practice for DMU i.

The given models can easily be converted to VRS models, also known as the BCC model (Banker, et al., 1984). In the envelopment form of the model we add a convexity constraint $\sum_{j=1}^{d} \lambda_j = 1$:

$$\begin{aligned} \min \quad & \theta \qquad\qquad (4) \\ s.t. \quad & -y_i + Y\lambda \geq 0 \\ & \theta x_i - X\lambda \geq 0 \\ & \sum_{j=1}^{d} \lambda_j = 1 \\ & \theta, \lambda \geq 0. \end{aligned}$$

Inclusion of the VRS constraint in the multiplier form requires an additional variable, $\nu_0 \in \mathbb{R}$, which is unrestricted, i.e. has no bounds restricting its value, as follows:

$$\begin{aligned} \max \quad & \nu^T y_i - \nu_0 \qquad\qquad (5) \\ s.t. \quad & \mu x_i = 1 \\ & \nu^T y_j - \mu^T x_j - \nu_0 \leq 0 \qquad \forall\, j = 1,2, \dots, d \\ & \mu, \nu \geq 0.;\ \nu_0 \text{ unrestricted} \end{aligned}$$

Output-oriented models are formulated analogously to the above models. Other modifications to the above DEA models allow for situations where the assessment of efficiency should take into account more nuanced approaches. These include, for instance, weak efficiency, weight restrictions, non-discretionary variables, or categorical variables. References for these approaches are provided in (Seiford, 1990), and a brief explanation of these concepts is provided in the following.

Weak efficiency. Consider Figure 3 which is similar to Figure 1 but with the addition of DMU H.

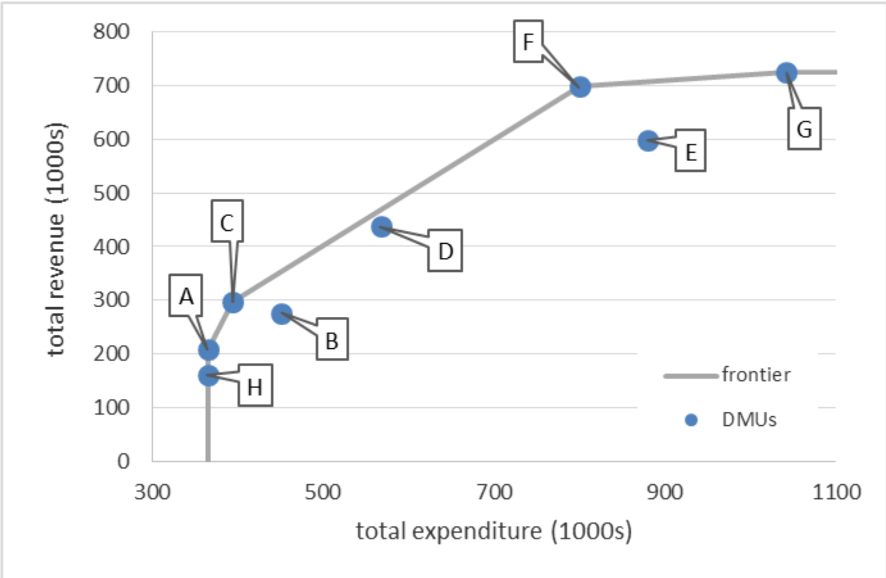


Figure 3: Weak efficiency

DMU H uses the same amount of input as DMU A but only produces 160 units of output compared with A's output of 209 units. H is described as "weakly efficient" because, although it is on the frontier, it is still dominated by A. However, the VRS models above will assign a score of 100% efficiency to H [e]. Attempts to overcome this issue use either a non-Archimedean infinitesimal as a lower bound in models (1), (2), or (5) (Ali & Seiford, 1993) or solve (3) or (4) in a two stage process (Cooper, et al., 2007). Using model (4) as an example, the evaluation of H will result in $\theta = 1$ [e]. A second stage model is then solved similar in form to (4) but where the objective function maximises the slacks with all other constraints being the same (since $\theta = 1$ there is no change in the output for H). DMUs that are strongly efficient will have zero slack in the optimal solution but weakly efficient DMUs will have positive slack on the output constraint since A will form the reference DMU for H. In the `pyDEA` software (Section 5), maximising slacks is provided as an option. However, although

[e] For instance, model (4) for DMU $i = H$ minimises θ while satisfying $y_H \leq \sum_{j=A}^{H} y_j \lambda_j$ and $\theta x_H \geq \sum_{j=A}^{H} x_j \lambda_j$ with $\sum_{j=A}^{H} \lambda_j = 1$. Since H and A are the DMUs which consume the least input ($x_H = x_A \leq x_j$ for $j = B, \dots, G$), only λ_H, λ_A can take non-zero values and $\theta = 1$ (corresponding to 100% efficiency). H is considered efficient as its input cannot be reduced.

weakly efficient DMUs appear to be a problem in theoretical examples, it does not occur that regularly in practice.

Weight restrictions or weight constraints (Allen, et al., 1997), (Dyson, et al., 2001) allow the modeller to limit the freedom of choice of weights in the models. For instance, a decision maker may not want the weight of an important input or output to be zero, or there may be relationships between inputs or outputs or both to be incorporated. Simple weight constraints, called absolute weight restrictions, in the multiplier form could be, for instance, $\alpha \leq \mu_r \leq \beta$ or $\gamma \leq \nu_s \leq \delta$ for some $r \in \{1,2,\dots,n\}$ or $s \in \{1,2,\dots,m\}$ and parameters $\alpha, \beta, \gamma, \delta$. Virtual weight restrictions constrain the weighted input or output, for instance $\bar{\alpha} \leq \mu_r x_{j,r} \leq \bar{\beta}$ or $\bar{\gamma} \leq \nu_s y_{j,s} \leq \bar{\delta}$, for parameters $\bar{\alpha}, \bar{\beta}, \bar{\gamma}, \bar{\delta}$ with $x_{i,r}$ and $y_{i,s}$ being input r and output s of any DMU j. Constraints that limit the ratio of weights such as $\frac{\nu_s}{\nu_t} \leq \epsilon$ can also be included in the LP models.

Non-discretionary environmental variables (Banker & Morey, 1986) are those that are out of the control of a DMU's management. For example, the floor area of a retail store cannot usually be changed easily in the short term. In an input-oriented model non-discretionary inputs are modelled by removing these corresponding inputs from the set of shrinkable inputs, i.e., constraint $\theta x_i - X\lambda \geq 0$ is now only considered for those rows (inputs) that are discretionary.

Categorical variables (Banker & Morey, 1986) can also be used to capture environmental constraints under which DMUs operate. An example in the context of our case study is the location of post stores. We expect a smaller market of potential customers due to lower population for each store in a rural environment, a larger market on the outskirts of urban centres (satellite urban areas), and the largest market, or most potential customers, in large urban centres. We would therefore expect a store to do better with respect to many performance metrics (outputs) in an urban area. This can be captured by assigning each DMU to a category that reflects, say, its market (potential customers). When assessing DMU i's efficiency based on the above LPs it would only be compared to other DMUs which operate under the same circumstances, or circumstances that are considered a more challenging environment, therefore ensuring fairer benchmarking for efficiency comparisons. For example, with 3 categories and category 1 being the most challenging, DMUs in category 1 would only be compared with themselves, DMUs in category 2 would be compared against both category 1 and category 2 DMUs, while category 3 DMUs would be compared against all DMUs.

In summary, DEA is a popular non-parametric benchmarking tool. It is based on linear programming models that can be adjusted for problem-specific circumstances in order to more fairly compare DMU performance. A few popular additions to DEA models have been discussed here, but there are many more that have been proposed by researchers over the years. To assess the efficiency of all DMUs, an LP has to be solved for each DMU. The DEA models presented here can be solved using any linear programming solver, and specific DEA software packages are also available. In the following we introduce an open-source DEA package which has all the required features above, and can also be extended to include other DEA models as it is open-source.

It should be noted that there are analogies between DEA and multiobjective optimisation (Yougbaré & Teghem, 2007), and that DEA problems have been formulated as multiobjective linear programmes (Yu, et al., 1996), (Hosseinzadeh Lotfi, et al., 2008). The reader is also referred to our comments on the relationship of efficient solutions in multi-criteria optimisation problems and efficient DMUs at the beginning of Section 3.

5 SOLVING DEA USING PYDEA SOFTWARE [f]

`pyDEA` is an open-source software package developed in Python [f], shown in the screenshot in Figure 4. The software enables data to be easily imported from an Excel sheet. After selecting the Excel file to read in, the user can select from which worksheet to read the data. The first column of the sheet is reserved for the DMU names, and consecutive columns can be used as DEA inputs and outputs by selecting the corresponding option. The input data is shown on the "Data" tab in the main window. It should be noted that incomplete data columns, or those with invalid data (negative or zero inputs or outputs) cannot be chosen [g]. `pyDEA` also allows the user to manipulate and save data.

[f] `pyDEA` is being developed at the Department of Engineering Science, University of Auckland, New Zealand. It is available online as an open-source tool in python 3. It can be installed via pypi distribution https://pypi.python.org/pypi/pyDEA and source code is available on github https://github.com/araith/pyDEA

[g] Most DEA software requires strict positivity for inputs and outputs. Thus negative values need to be adjusted. For the VRS models a negative input or output can be adjusted by adding a number to all DMUs' values for that input or output to convert the minimum value to a positive one. Note that this can only be done for VRS models and not the CRS model. See also (Cooper, et al., 2007) Chapter 4.

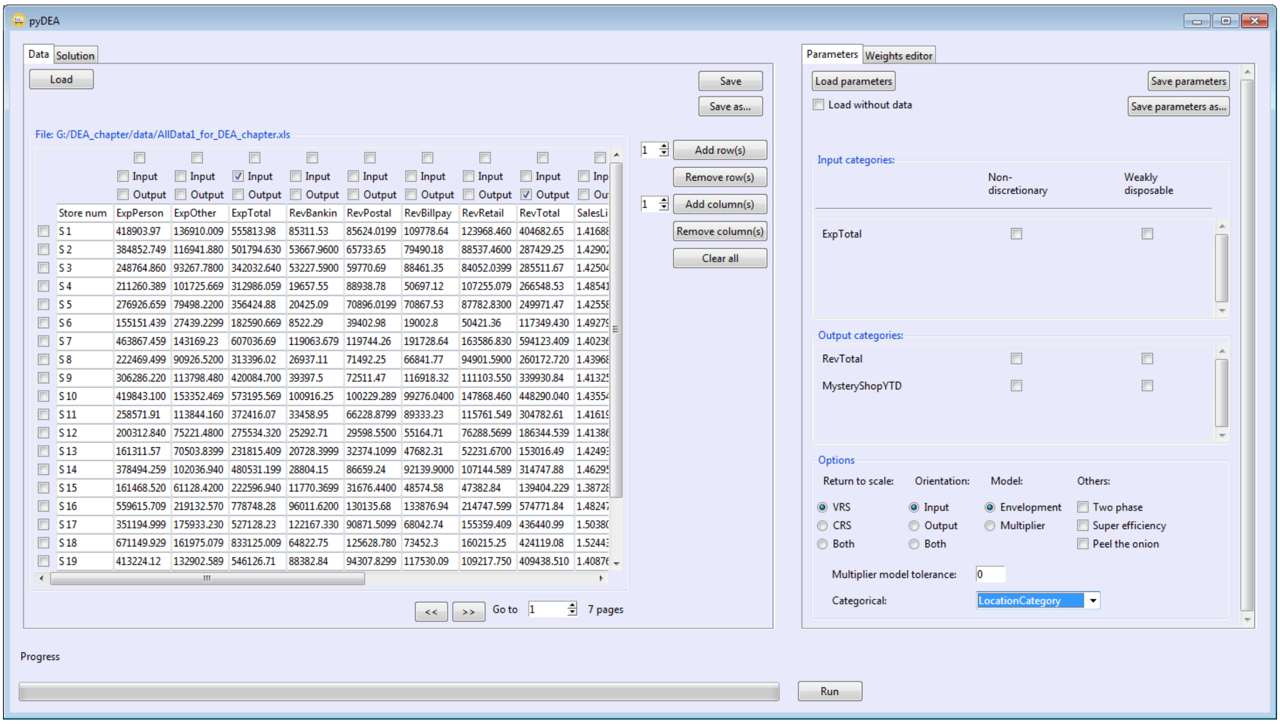


Figure 4: pyDEA screenshot showing Data and Parameters

As shown in Figure 5, each column in the dataset can be designated as an input or output.

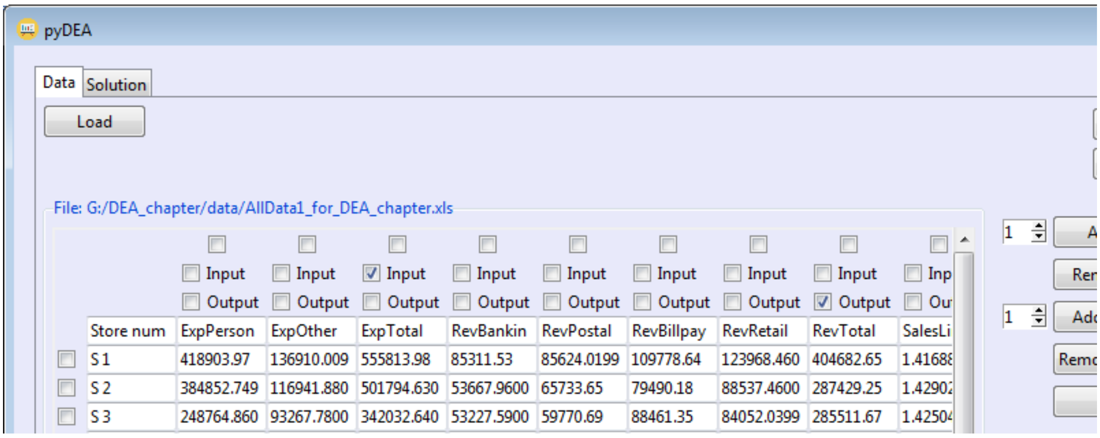


Figure 5: pyDEA screenshot showing choice of input and output

Additional DEA settings, such as non-discretionary inputs or outputs, can be selected in the right-hand side menu, see Figure 6. Currently selected inputs and outputs are listed here, and additional options can be selected by ticking the corresponding boxes. The chosen input in the model for which screenshots were taken is total expenditure (ExpTotal), and outputs are total revenue (RevTotal) and the customer satisfaction scores (MysteryShopYTD).

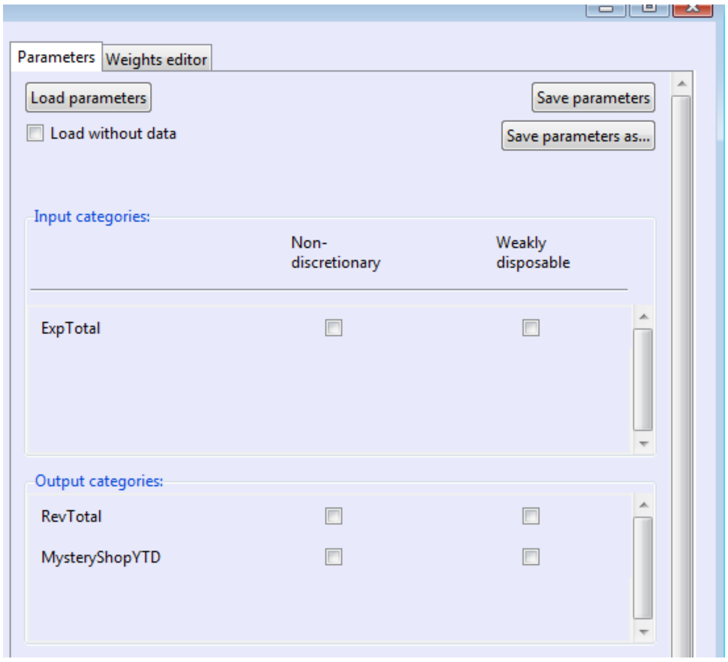


Figure 6: pyDEA screenshot showing Parameters

Finally, different DEA set-up options are available such as CRS, VRS, input or output orientation, envelopment or multiplier form of the models, and special DEA modes such as the so-called Two Phase approach to minimise slacks as described in Section 4 (Cooper, et al., 2007), super efficiency (Lovell & Rouse, 2003), or the so-called "Peel the onion" technique (Barr, et al., 2000), see Figure 7. For categorical models, the column in the data set which identifies the categories can also be selected here. The main window also has a "Weights editor" tab where weight restrictions, as described earlier, can be entered, see Figure 6. Finally, clicking the "Run" button (see Figure 7) solves the Linear Programmes to determine each DMU's efficiency. Linear programmes are formulated and solved using the Python PuLP package[h].

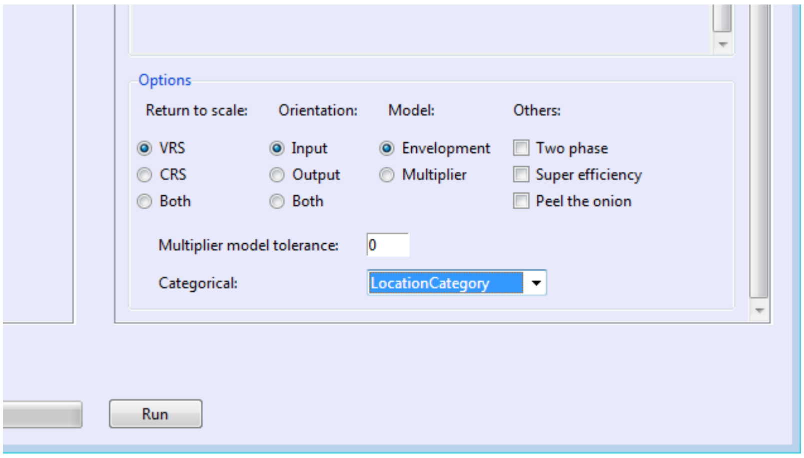


Figure 7: pyDEA screenshot showing Options and Run button.

[h] Optimization with PuLP https://pythonhosted.org/PuLP/

Results are displayed in the main window's "Solution" tab. Each solution can be saved in Excel and csv format, or copied and pasted to another document. The efficiency scores tab is shown in Figure 8, and other types of output such as peers, peer counts, weights and targets are also available. These will be explained in the context of the case study in Section 7. The parameters tab contains the parameters under which the model was run. All tabs are saved as worksheets when saving in Excel format. As the screenshot is for a model with categorical variable, the solution tab displays both efficiency scores and the values of the categorical variable from the input data (this makes it easier to interpret results without having to refer back to the original data). For a discussion of models with categorical variables we refer to Section 7, Model 2 with categorical variables.

Detailed instructions on running pyDEA, with more screenshots, are given in the Appendix (Section 9) for the case study presented in Section 7.

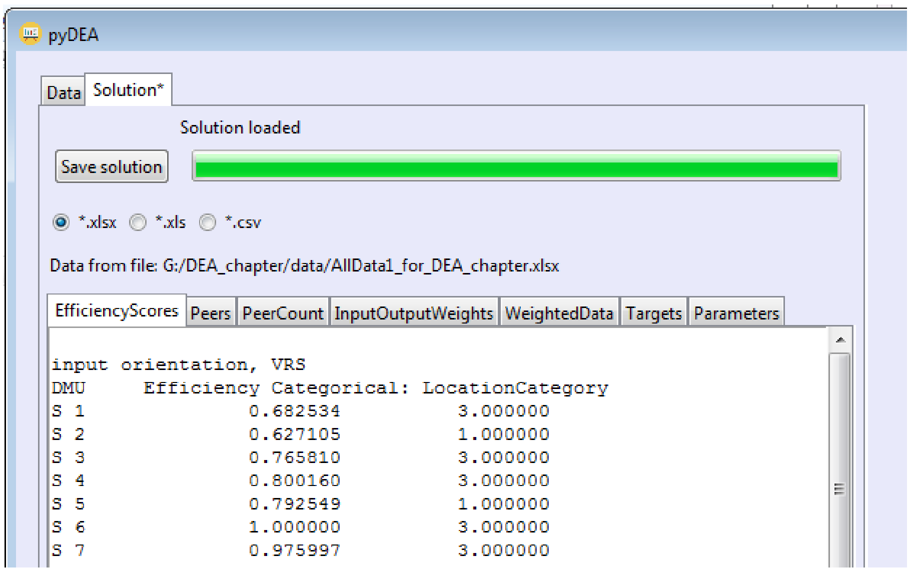


Figure 8: pyDEA Solution display.

6 METHOD AND MODEL BUILDING

In the following we introduce three different DEA models for the case study, each with increasing complexity. While there are many descriptors of store performance, as discussed earlier for this case study, the main aspects are summarised in Figure 9. Some factors are omitted as they do not vary much between stores. An example is the average number of sales lines per customer, which is 1.43 with a standard deviation 0.06. We note that floor area and rent are dropped from the analysis to simplify the example, and it could be argued that they are out of the immediate control of store management. We also omit staff FTE, as personnel expenditure captures staffing cost. For location we consider only the three categories rural, satellite urban (outer suburb of urban centre), and urban (large urban centre) and assume the categories define an increasingly favourable operating environment. Finally, we distinguish stores with full banking functions (a total of 48), and those with only postal functions, but without major banking functions (a total of 95; this group includes four stores with limited banking functions).

All DEA models in the following are *input-oriented VRS models*. We consider three different DEA models listed in Table 2, discuss their high-level results, introduce a few DEA concepts such as categorical models and weight restrictions, and conduct a more detailed analysis of Model 3.

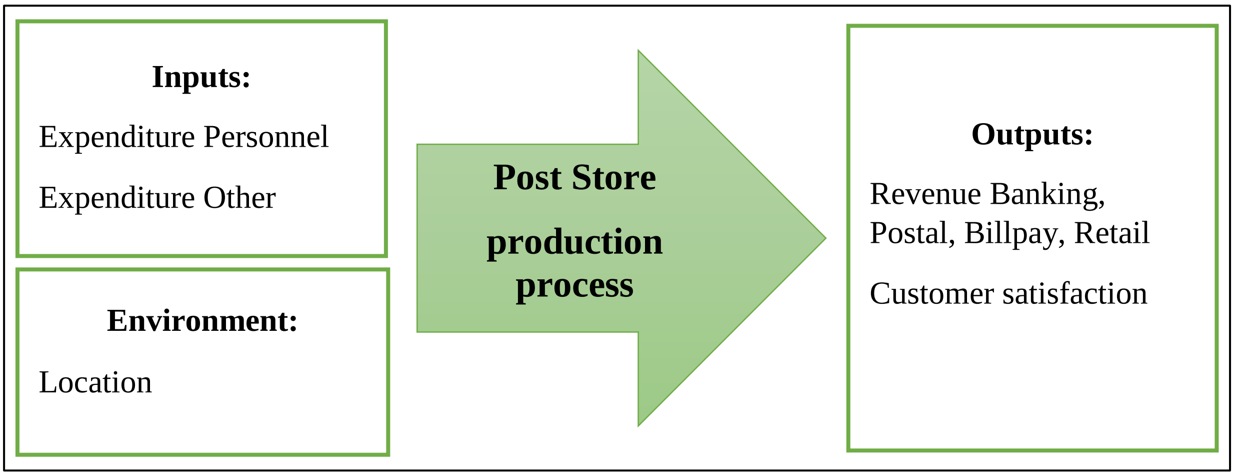


Figure 9: DEA Model for Post Stores

Table 2: DEA models applied to Post Store case study

	Inputs	Outputs	Stores (DMUs) considered	Orientation	Returns to scale
Model 1	Total Expenditure	Total Revenue Customer Satisfaction	All	Input	VRS
Model 2	Expenditure Personnel Expenditure Other	Revenue Bank (Banking) Revenue Post Total (Postal, Billpay, and Retail) Customer Satisfaction	All	Input	VRS
Model 3	Expenditure Personnel Expenditure Other	Revenue Post Total (Postal, Billpay, and Retail) Customer Satisfaction	Post only stores	Input	VRS

7 RESULTS

In this section we present the results of the case study for the different DEA models listed in Table 2. Throughout this section we show results for a subset of post stores which is arbitrarily chosen, as showing full results would require too much space. Detailed instructions on running `pyDEA` to produce the results outlined below, with screenshots, are given in the Appendix (Section 9) although, again, the full results cannot be shown due to space limitations.

Model 1 Results

In DEA Model 1 (without categorical variables), nine of the 143 post stores are found to be efficient. Five of these stores are located in an urban location, one in a satellite urban location, and three in a rural location. Table 3 lists the efficiency scores of Stores 10 to 20. In Table 3, Store 13 and 15 are efficient and other stores have lower efficiency scores. Overall, efficiency scores range from 0.43 to 1.00 with an average of 0.75.

Table 3: DEA Model 1 results: efficiency scores and weights of stores 10-20

store	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**
efficiency	0.75	0.73	0.73	**1.00**	0.58	**1.00**	0.73	0.79	0.53	0.70	0.75

Due to the single input and only two outputs of Model 1, we can visualise Model 1 (in two dimensional projections) in Figures 10 and 11. Figure 10 plots the input of Model 1 (Total Expenditure) against the first output (Total Revenue), Figure 11 again plots the input against the second output (Customer Satisfaction Score). The efficient post stores are labelled in the two plots: Main urban efficient stores are 6, 74, 75, 106 and 132, the satellite urban efficient store is 92, and the rural efficient stores are 13, 15, and 138. Since Model 1 has two outputs, some stores may appear efficient in only one of the two figures. An example is Store 138, which is dominated by other stores with lower expenditure and higher revenue in Figure 10. However, Store 138 is not dominated in Figure 11 where no other store has both lower expenditure and higher customer satisfaction score than Store 138. The figures also show that there appears to be a correlation between Total Expenditure and Total Revenue, but no clear relationship between Total Expenditure and the Customer Satisfaction Score. The three efficient rural stores stand out mainly due to their high customer satisfaction score. The efficient satellite urban store is efficient as it uses the least input (Total Expenditure) of all stores. This illustrates that in a DEA VRS analysis, a DMU that achieves at least one minimum input or maximum output will be considered efficient.

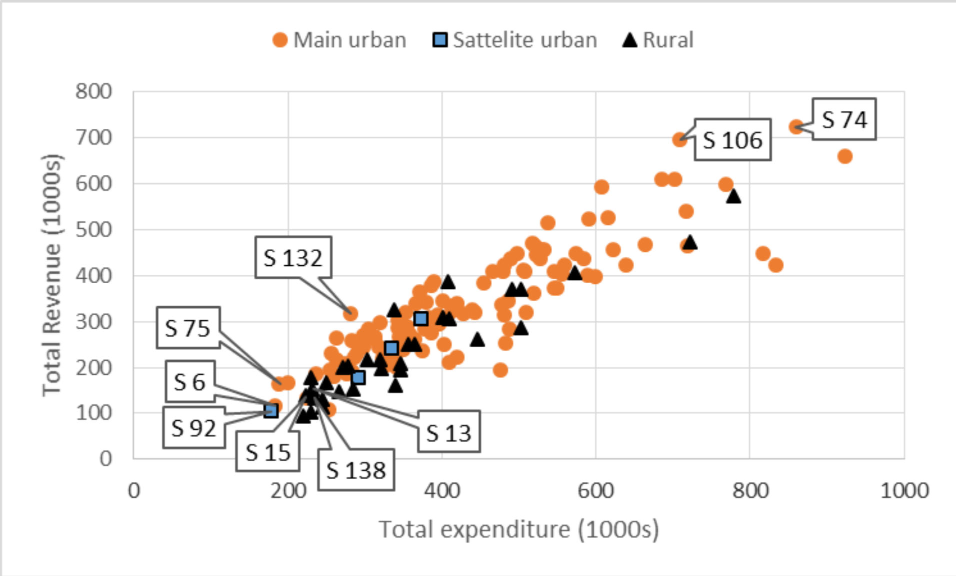


Figure 10: Total Expenditure and Total Revenue

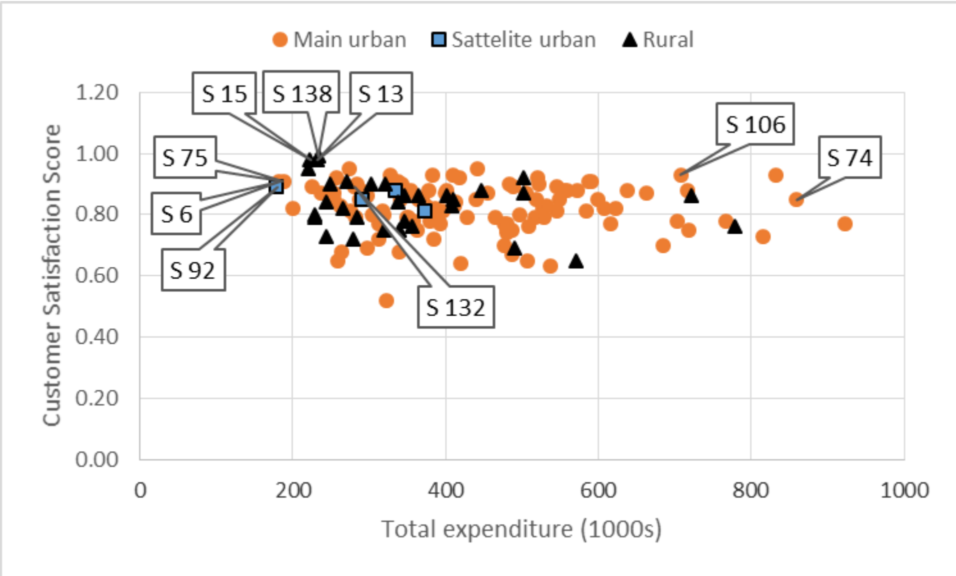


Figure 11: Total Expenditure and Customer Satisfaction Score

Model 2 Results

In DEA Model 2 (without categorical variables), 21 of the 143 post stores are found to be efficient. 16 of them are urban, one is satellite urban, and four are rural. The average efficiency score is now 0.82 with a minimum of 0.48 and a maximum of 1. Efficiency scores of Stores 10 to 20 are listed in Table 4. Comparing Tables 3 and 4, we observe that Stores 13 and 15 remain efficient, and the other efficiency scores shown increase, or remain unchanged.

Weight interpretation

When solving DEA VRS model (5), an efficiency score for each DMU is obtained. Other model outputs are the virtual input weights $\mu_r x_{j,r}$ (or weighted data values) for each input r and DMU j in model (5), as shown in Table 4. The weights indicate the importance placed by a store on each of the inputs. A low weight indicates that the corresponding input has little influence on the store's efficiency score meaning the store does not compare favourably with regards to this input, hence a low weight is placed on the input. Both efficient Stores 13 and 15 place a lower weight on "Expenditure Other" than on "Expenditure Personnel", and the same can be said for most of the stores shown in Table 4. Store 17 places a weight of 0.00 on "Expenditure Other" meaning this input is not considered in the efficiency computation at all. Only Stores 14 and 18 place a higher weight on "Expenditure Other" than on "Expenditure Personnel".

Table 4: DEA Model 2 results: efficiency scores and virtual weights of stores 10-20

Store / weight	10	11	12	13	14	15	16	17	18	19	20
efficiency	0.81	0.78	0.76	**1.00**	0.72	**1.00**	0.74	0.98	0.72	0.76	0.75
Virt. weight Exp Other	0.32	0.04	0.32	0.10	0.78	0.24	0.23	0.00	1.00	0.38	0.22
Virt. weight Exp Personnel	0.68	0.96	0.68	0.90	0.22	0.76	0.77	1.00	0.00	0.62	0.78

Comparing efficiency scores in Models 1 and 2

Figure 12 shows a histogram of the efficiency scores in Model 1 and Model 2. The figure shows that efficiency scores in Model 2 are higher. This is due to the higher dimensionality of the Model 2 dataset, i.e., considering two different aspects of expenditure as inputs, and two aspects of revenue as outputs. In a DEA model this leads to more freedom of choice of input and output weights for each DMU, and therefore the ability to choose more favourable aspects. Store 17, for instance, is now considered almost efficient with a score of 0.98, whereas it only had an efficiency score of 0.70 in Table 3. While the input (total expenditure) in Model 1 was fully considered for Store 17, the more favourable expenditure aspect of Store 17 (Expenditure Personnel) only is considered in Model 2, and this input now has a weight of 1.00. As Store 17 is compared to other stores under this more favourable weighting in Model 2, it is now operating almost efficiently, when assessed under DEA. In Model 2 all efficiency scores are higher than in Model 1, where efficiency scores in Model 2 are 8.7% higher than those in Model 1 on average. In a DEA analysis it is important to carefully choose the inputs and outputs to be considered to avoid overinflating efficiency scores. If too many are included, especially when there are only few DMUs under consideration, efficiency scores may increase because there is more opportunity for each DMU to appear efficient by considering only few favourable inputs and outputs (with others having their weight set to zero).

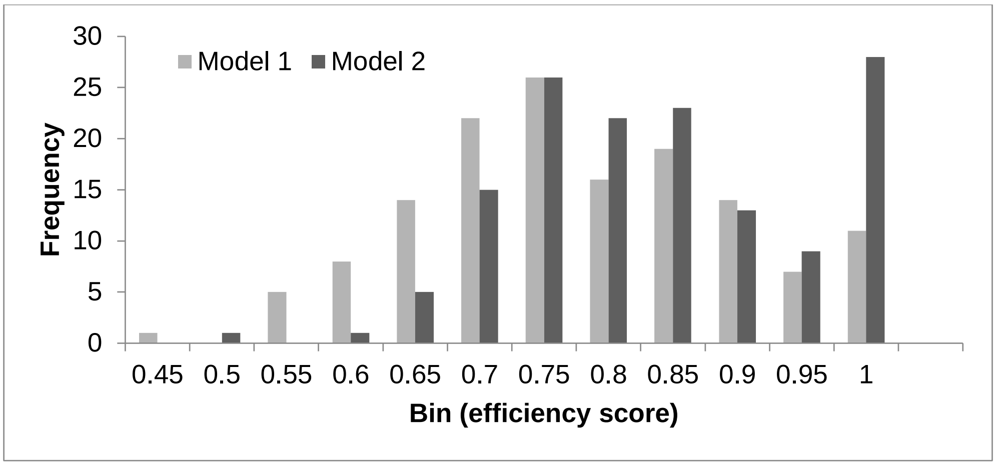


Figure 12: Histogram of efficiency scores.

Categorical variables

We now include categorical variables in the analysis where it is assumed the most favourable category is that of urban stores with a large market of potential customers, and the least favourable category consists of rural stores with fewer potential customers. Figure 10 confirms this as rural stores tend to have lower total expenditure, but also lower total revenue. With these categorical variables, rural stores are considered as the first group, then satellite urban stores are included, and finally all stores are considered when efficiency of urban stores is computed. Table 5 shows average efficiency scores of the different types of stores for Model 2 with and without categorical variables. The average efficiency of urban stores is not affected by the categorical model, as expected. The average efficiency scores of satellite urban and rural stores increase, where rural stores improve the most. For instance, the efficiency score of Store 11 was 0.78 (Table 4), and it is 0.90 in the model with categorical variables. This illustrates how categorical variables can be used to adjust efficiency scores when DMUs operate in different environments. Despite this Model 3 will not consider categorical variables as we wish to illustrate the power of virtual weights and peers as benchmarks in our discussion.

Table 5: DEA Model 2 average efficiency scores by category without and with categorical variables

	Number of stores	Model 2	Model 2 with categorical variables
Rural	32	0.78	0.96
Sattelite urban	4	0.80	0.92
Urban	107	0.83	0.83
overall	143	0.82	0.86

Model 3

Finally, DEA Model 3 is only run for the subset of stores without special banking facilities. Of the 95 stores, 16 are efficient. 11 efficient stores have an urban location, one a satellite urban location, and four are in a rural location. In Model 3 the average efficiency score is 0.81 with a minimum of 0.47 and a maximum of 1.00. Considering the first few stores in Table 6, we again observe different (virtual) weighting placed on each of the inputs and outputs. Virtual weights (or weighted data values) are again $\mu_r x_{j,r}$ for each input r and DMU j and $\nu_s y_{j,s}$ for each output s. The customer satisfaction measure is interesting to consider more closely, as it often receives a weight of 0.00 indicating that it is not favourable for the corresponding store to include in its efficiency computation. The VRS model weight ν_0, see also linear programming model (5), can give an indication whether each store operates under constant $\nu_0 = 0$, increasing $\nu_0 < 0$ or decreasing $\nu_0 > 0$ returns to scale. We note that parameter ν_0 is not unique due to the potential existence of alternate optima, and hence may not correctly identify returns to scale. One would need to examine returns to scale as discussed in (Seiford & Zhu, 1999). Store 6 operates under constant returns to scale with $\nu_0 = 0$, and would hence be efficient also in CRS models (2) or (3).

Table 6: DEA Model 3 results for the first ten post stores: efficiency scores and virtual weights

Store / weight	**3**	**4**	**5**	**6**	**8**	**9**	**11**	**12**	**13**	**14**
efficiency	0.72	0.81	0.79	**1.00**	0.78	0.74	0.72	0.75	**1.00**	0.78
Virt. weight Exp Other	0.31	0.21	1.00	0.15	0.33	1.00	0.19	0.31	0.10	1.00
Virt. weight Exp Personnel	0.69	0.79	0.00	0.85	0.67	0.00	0.81	0.69	0.90	0.00
Virt. weight Rev Post Total	0.38	0.48	0.85	0.00	0.41	0.87	0.44	0.32	0.36	0.82
Virt. weight Cust. Satisfaction	0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	2.52	0.00
Weight VRS ν_0	0.34	0.33	-0.06	0.00	0.37	-0.13	0.28	0.43	-1.88	-0.04

Peers as Benchmark DMUs

The envelopment form (4) of the DEA model allows further interpretation of the results. The envelopment model identifies peer weights λ for each DMU other than the DMU i that is being evaluated. The DMU under evaluation is mapped to a composite DMU $(X\lambda, Y\lambda)$, which consists of a convex combination of all other DMUs. Each DMU j with $\lambda_j > 0$ helps define this composite DMU. DMU j with $\lambda_j > 0$ is called a *peer* of DMU i, which is under evaluation. For each efficient DMU we can analyse how often it acts as a peer for other DMUs. In DEA Model 3, Store 6 acts as peer for 52 other DMUs. It is thus a valuable reference, or benchmark, store for other stores. All stores that are efficient, are benchmark stores. Store 13 is also efficient, but only acts as a peer DMU five times, whereas Store 68 only acts as a peer twice (once for itself). From this we can conclude that the mix of inputs and outputs of Stores such as 13 and 68 is somewhat unique and not comparable to that of many other DMUs. Stores 68 and 13 could be considered efficient mainly because they are different to other stores, not because they necessarily perform extremely well. Of the efficient stores, most act as a peer (or benchmark) for only a few stores. However, some of the efficient stores act as a peer many times, such as Stores 6, 75, 132, and 143. Stores 6 and 132 stand out as peers of 52 and 68 stores, respectively.

For an inefficient store, an analysis of the benchmark stores, and their contribution to the benchmark can provide store management with a reference set of stores they could aspire towards. Table 7 lists stores in the rows of column 1 and their respective peers in columns three to eight. For instance, benchmark references for Store 4 are Stores 129, 132 and 143 where Store 132 is the most important reference store with the highest weight. Reference stores for Store 12 are Stores 6, 132 and 143 where the latter carries the highest weight. Efficient Stores 6 and 13 are their own benchmark stores.

Table 7: DEA Model 3 results for the first 1o post stores: benchmark stores and peer weights λ

benchmark store / store	efficiency	6	13	129	132	135	143
3	0.72	0.25	-	-	0.62	-	0.12
4	0.81	-	-	0.21	0.51	-	0.28
5	0.79	0.36	-	-	0.64	-	-
6	**1.00**	1.00	-	-	-	-	-
8	0.78	0.14	-	-	0.60	-	0.26
9	0.74	-	-	-	0.91	0.09	-
11	0.72	-	-	0.05	0.78	-	0.17
12	0.75	0.35	-	-	0.16	-	0.48
13	**1.00**	-	1.00	-	-	-	-
14	0.78	0.06	-	-	0.94	-	-

Targets for inefficient DMUs

Considering inefficient stores, DEA also estimates targets for each input and output the stores should be able to achieve. An example of these targets for the first seven stores is shown in Table 8. DEA finds that none of the stores in Table 8 need to alter their postal revenue, however, most stores should be able to improve their customer satisfaction score. Store 3, for instance, should be able to improve its customer satisfaction score by about 4.29%, whereas Store 4 should aim for an improvement of 14.82%. These targets show the improvement necessary for a store to become efficient. Store management would need to identify actions that lead to the desired improvement. To improve its customer satisfaction score, Store 3 could visit peer stores to observe their customer processes, conduct staff training, and ask senior managers to assist in identifying areas of customer service shortfall. In Table 8 all in-efficient stores have a reduction target for both considered expenditure streams.

Table 8: DEA Model 3 results for the first 8 post stores: targets (in %)

store	3	4	5	6	8	9	11
efficiency	0.72	0.81	0.79	**1.00**	0.78	0.74	0.72
Exp Other	-27.99	-18.63	-20.90	0.00	-21.53	-26.35	-28.35
Exp Personnel	-27.99	-18.63	-34.02	0.00	-21.53	-33.01	-28.35
Rev Post Total	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Cust. Satisfaction	4.29	14.82	18.06	0.00	13.58	38.53	7.09

Weight restrictions: An example

When assessing store efficiency in the case of post stores, quite a few of the stores assign a weight of 0.00 to customer satisfaction, see Table 6. Management may not want to allow a DMU (post store) under assessment to place a weighting of 0.00 on this output as customer satisfaction is an important measure. Similarly, management also may not believe that a post store should be considered efficient when it places a weight that is too high on its customer satisfaction score, such as Stores 6 and 13 in Table 6. Weight restrictions can enforce more appropriate consideration of the customer satisfaction score in the DEA model. This can be achieved by placing a lower bound on the weight ν_s associated with customer satisfaction (output s) in the form of absolute or virtual weight restrictions: $\nu_s \geq \gamma$ or $\nu_s y_{j,s} \geq \bar{\gamma}$ for an appropriate choice of γ or $\bar{\bar{\gamma}}$. To demonstrate this we now require the virtual weight of the customer satisfaction score to be between 0.10 and 0.30.

Table 9 shows the resulting efficiency scores (with and without weight restrictions) and virtual weights for Models with weight restrictions. Efficiency scores either remain unchanged or decrease. The model with weight restrictions has only six efficient stores (four urban, and one each satellite urban and rural). Virtual weights of customer satisfaction are now always between 0.10 and 0.30, as required, and the other weights change as they adapt to the newly introduced weight restrictions.

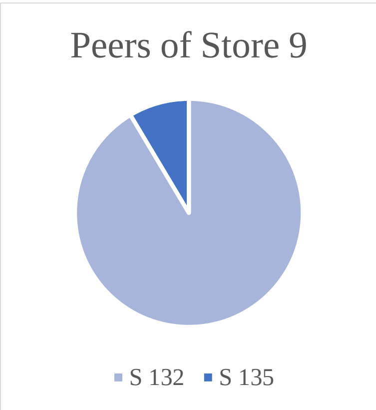
Store 3 had a virtual weight of 0.00 for Customer Satisfaction. The new lower bound of 0.10 ensures this virtual weight is now 0.10, and the virtual weight of Revenue Post Total drops slightly due to the newly enforced weight constraints. Overall, however, the efficiency score of Store 3 does not change. Store 5, on the other hand, also now has a virtual weight of 0.10 for Customer Satisfaction, which causes its efficiency to drop from 0.79 (Model 3) to 0.77 (Model 3 with weight restrictions). We also see that Stores 6 and 13, whose virtual weight for Customer Satisfaction was above 0.30, now both have a virtual weight of 0.30, the enforced upper bound. The efficiency score of Store 6 is not affected by this, whereas the efficiency of Store 13 drops significantly from originally 1.00 (Model 3) to 0.85 (Model 3 with weight restrictions). As also shown in Figure 11, Store 13 has a very high Customer Satisfaction score, but doesn't perform so well in other aspects, whereas Store 6 has very low expenditure, which explains its efficiency even when the contribution of the Customer Satisfaction score is limited.

Table 9: DEA Model 3 results with weight restrictions for the first ten post stores: efficiency scores and virtual weights

Store / weight	3	4	5	6	8	9	11	12	13	14
Efficiency Model 3	0.72	0.81	0.79	**1.00**	0.78	0.74	0.72	0.75	**1.00**	0.78
Efficiency with weight restrictions	0.72	0.80	0.77	**1.00**	0.77	0.70	0.71	0.74	0.85	0.76
Virt. weight Rev Post Total	0.37	0.49	0.85	0.00	0.40	0.93	0.45	0.32	0.31	0.83
Virt. weight Cust. Satisfaction	0.10	0.10	0.10	0.30	0.10	0.10	0.10	0.10	0.30	0.10

Summary: Benchmarking Model 3

In summary, DEA allows benchmarking of DMUs, or post stores in our case. For each post store an efficiency score indicates its level of performance when benchmarked against other stores. DEA allows inefficient stores to identify their "best practice" peer stores which operate similarly in terms of the considered inputs and outputs. These peers can act as comparable well-performing stores to guide management in improving performance. DEA also allows one to analyse the importance of inputs and outputs in a store's performance and thus to identify weaker aspects of performance. Finally, targets derived from DEA are based on observed performance and thus indicate achievable goals for each inefficient store to work towards. A summary of results for Stores 9 and 20, which have similar efficiency scores, is given in Figures 13 and 14, respectively. Store 9 has an efficiency score of 0.74. Its peers are stores 132 and 135, where Store 132 is the major peer with a peer weight of 0.91 (left chart in Figure 13). This would indicate that management of Store 9 should consult management of Store 132 to identify potential improvement strategies. The targets for Store 9 indicate that it should be able to decrease expenditure for personnel and other expenditure by 33% and 26.35%. As the target for customer satisfaction is to increase the current score by 38.5%, results also indicate that customer service at Store 9 needs to improve.



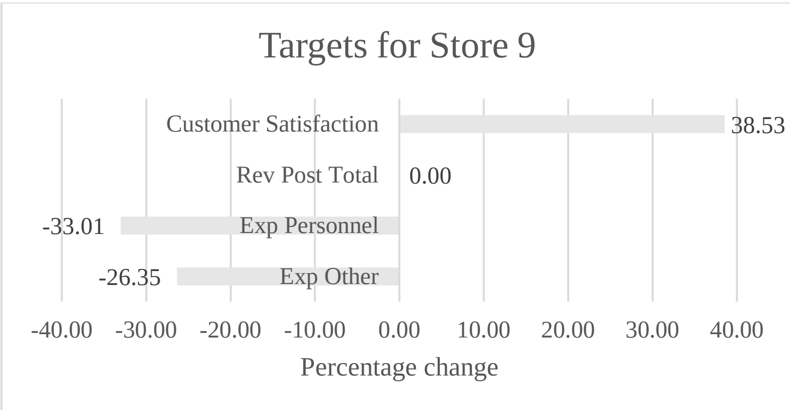


Figure 13: Results for in-efficient urban Post Store 9

Figure 14 shows results for rural post Store 20, which has an efficiency score of 0.75. Store 20 has three peers, Stores 6, 75 and 92, where Stores 6 and 75 carry the highest peer weights of 0.54 and

0.37, respectively. This indicates that they may both serve as benchmarks to identify management strategies to improve for Store 20. Targets for Store 20 indicate that there is potential to reduce both types of expenditure by 24.6% while increasing the customer satisfaction score by 24.4%.

Figure 14: Results for in-efficient urban Post Store 20

The presented DEA results could form the basis of a discussion around store performance and inform how to set performance targets for this store. The store's progress can be observed and assessed by applying DEA benchmarking annually.

8 Discussion and Conclusions

A brief discussion of advantages an disadvantages of DEA is presented before the chapter concludes.

Discussion of advantages and disadvantages of DEA

Advantages are of DEA are that no assumptions about distributional forms are made, i.e. DEA models are non-parametric. This means that DEA models do not require a single functional form that dictates how inputs produce outputs, but allow individual DMUs flexibility in their production configurations. Secondly, DEA has the units invariance property which means that inputs and outputs can be measured in different units.

One disadvantage of DEA is due to efficiency calculation being relative to the considered sample of DMUs, as well as inputs and outputs selected in the analysis. There can also be difficulties with dimensionality for large numbers of inputs and outputs and low numbers of DMUs, i.e. when many inputs and outputs are selected for a small set of DMUs most DMUs may appear efficient. DEA is also an extremal method, and there may be distortions due outliers in the form of very small or large values in inputs or outputs. On the other hand, these are identified easily in DEA which is an advantage.

Summary

This chapter introduced DEA as a tool for analysing and benchmarking performance of Decision Making Units. We discussed DEA in the context of a case study with the aim to benchmark the performance of post stores as introduced in Section 2. The underlying principles of a DEA analysis were introduced in Section 3, and their formulation as Linear Programming problems in Section 4. Section 5 outlined `pyDEA`, python-based open-source DEA software. In Section 6 we returned to the case study and considered three different DEA models with different sets of inputs and outputs. The results of the DEA analysis under the different models were presented in Section 7, together with discussions comparing different models and introducing DEA features such as categorical variables and weight restrictions.

Acknowledgements

The authors thank the Auckland Medical Research Foundation who partially supported the development of open-source software package `pyDEA` as part of project 1115021 *Knowledge-based radiotherapy treatment planning*.
The authors also thank NZ Post for letting us use their data for the presented case.

9 APPENDIX: MODELS FROM SECTION 7 IN PYDEA

In the following we provide instructions and screenshots of pyDEA settings to illustrate how the different DEA models for the case study are solved. The description contains most detail for Model 1, only differences are shown for the other models. All instructions are current at the time the chapter was written.

Installing and starting pyDEA

Windows operating system:

- Python version between 3.2 to 3.5 (at the time of writing) must be installed.
- Open a command window (cmd.exe).
- To find out which version of python is installed type in the command window:
  `py –-version`

  ```
  C:\Users\andrea>py –-version
  Python 3.5.4rc1
  ```

  In the example, the python version is 3.5

- To install pyDEA, type the following command, *where 3.x is replaced by your version of python* (see previous point).

```
py -3.x –m pip install pyDEA
```

- After successful installation, to run `pyDEA` type the following command in the command window:

```
py –m pyDEA.main_gui
```

Linux operating system:

- Python version between 3.2 to 3.5 (at the time of writing) must be installed.
- Open a terminal.
- In Linux, python 2.x (if installed) is usually available via command

```
python2
```

- whereas, python 3.x (if installed) is usually available via command

```
Python3
```

- The generic python command may point to either version:

```
python
```

- To find out which version of python is installed type in the terminal

```
python --version
```

For example:

```
andrea@computer:~$ python --version
Python 2.7.12
```

In the example, `python` maps to version 2.7, whereas `python3` maps to version 3.5:

```
andrea@computer:~$ python3 ––version
Python 3.5.2
```

- To install pyDEA, type the following command. If `python` maps to python 3.x use `pip` to install (or use the next option with `pip3`):

```
pip install pyDEA
```

- Otherwise (if python maps to python 2.x, and you have python3 installed, use pip3 to install):

```
pip3 install pyDEA
```

- After successful installation, to run it type the following command in the command window:

```
python3 –m pyDEA.main_gui
```

- Alternatively, simply type “pyDEA”, for example:

```
andrea@computer:~$ pyDEA
```

Using pyDEA

Starting `pyDEA` brings up the `pyDEA` main window, as shown in Figure 15. The main window has a Data section (left part of the window) and a parameter section (right part of the window).

Input data can be in csv, xls and xlsx format. The “load” button brings up a dialogue to browse to the location of the input file, and select it, as shown in Figure 16.

If the Excel file contains more than one worksheet a dialogue allows selection of the appropriate worksheet containing the data, see Figure 17.

Having loaded the data, it is displayed in the left data section of the screen, as shown in Figure 4.

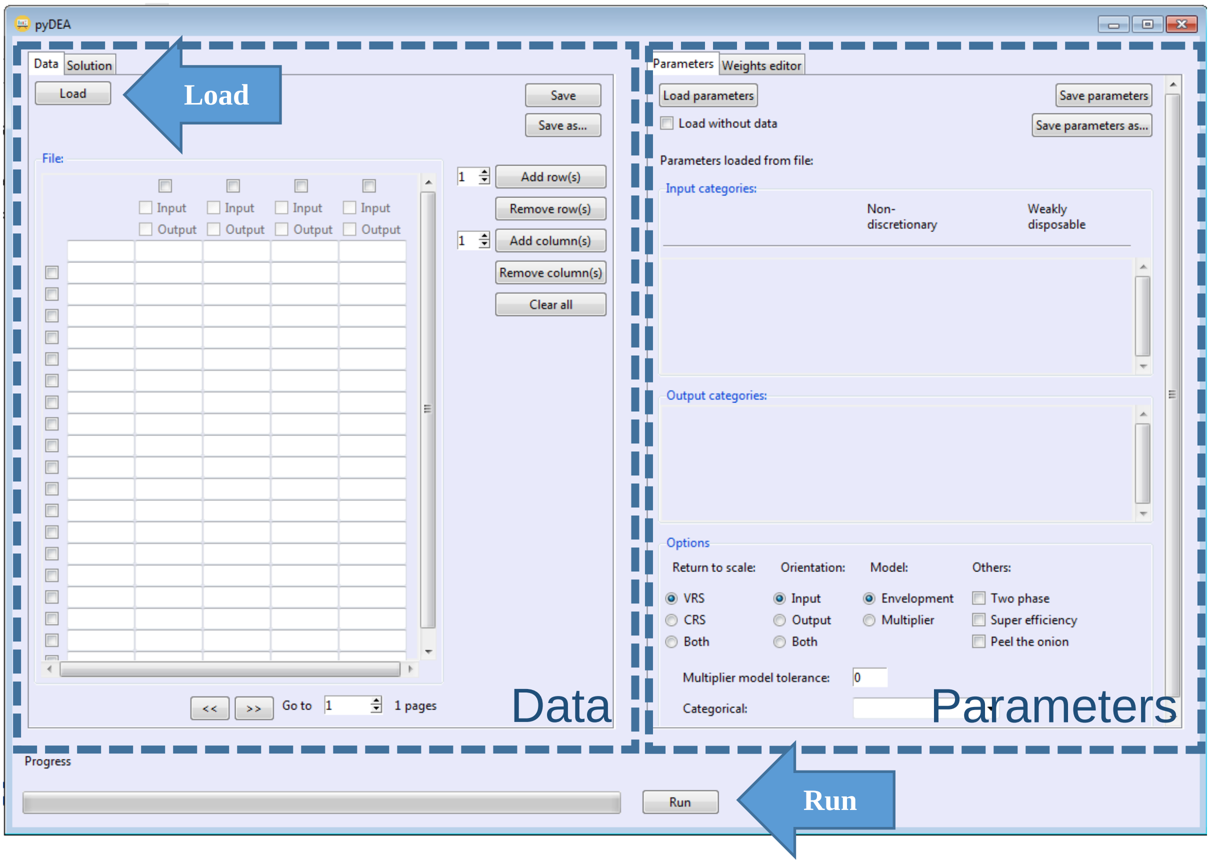


Figure 15: pyDEA main window.

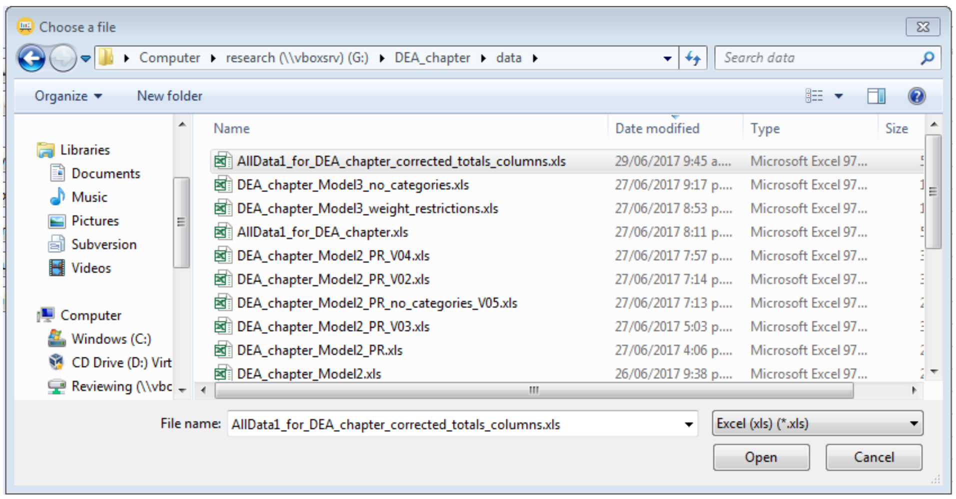


Figure 16: pyDEA input file selection window.

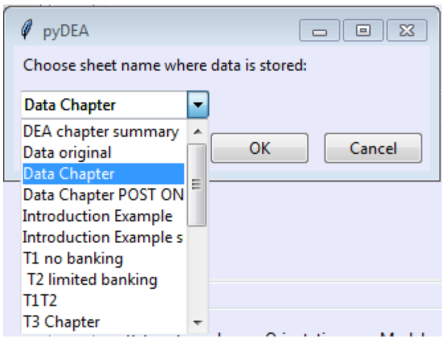


Figure 17: pyDEA input file worksheet selection.

Model 1

Start `pyDEA` and choose the input data set, as explained above. Choose the inputs and outputs for Model 1 by selecting "input" or "output" for the corresponding columns in the data, as shown in Figure 18. Note that the customer satisfaction score is stored as "MysteryShop" in the data set.

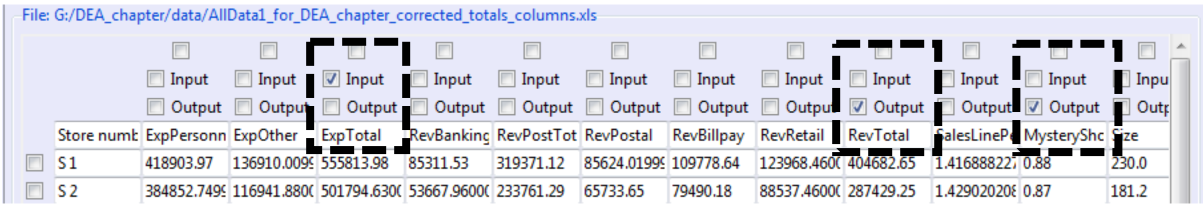


Figure 18: pyDEA input and output selection Model 1.

The parameters for a particular DEA model are chosen in the parameter section on the right side of the `pyDEA` window as highlighted in Figure 15. Since Model 1 is an input-oriented VRS model, these two options are selected in `pyDEA`. We could solve both the Envelopment or Multiplier form of DEA, but we keep the default Envelopment form here, see Figure 19.

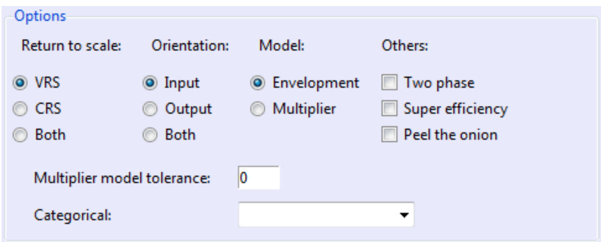


Figure 19: Parameters for Model 1 and 2.

Once all selections are made the "Run" button (Figure 15) computes DEA results, which are then displayed on tab "Solution" on the left side of the screen, see Figure 20. The asterisk next to "Solution" indicates that solutions have not yet been saved.

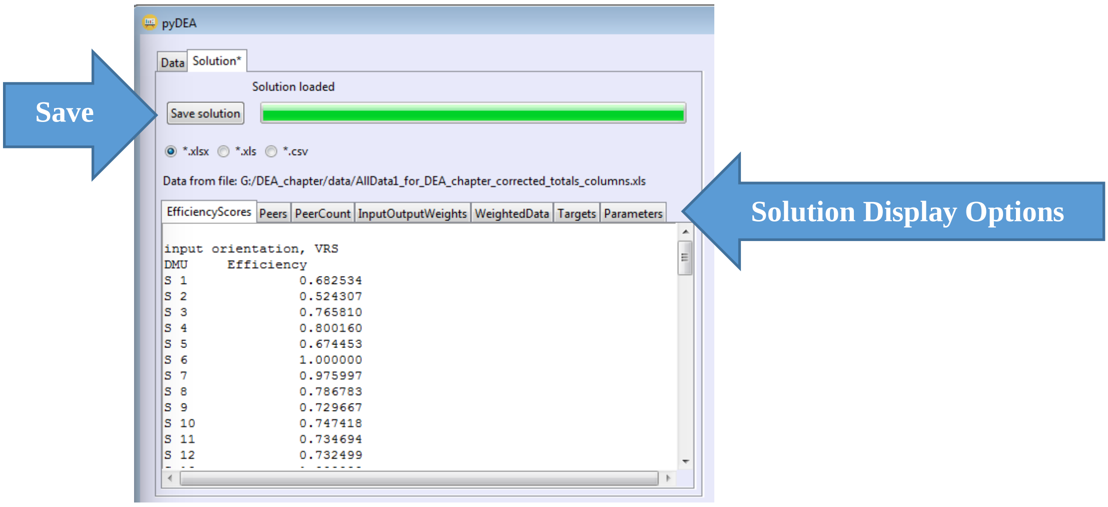


Figure 20: Solution tab for Model 1.

There are several solution displays available, such as “Peers”, “PeerCount”, “InputOutputWeights”, etc., as shown in Figure 20. They can be explored in `pyDEA`; they can be selected, copied and pasted; or they can be saved by clicking the “Save solution” button. A “Save As” dialogue opens allowing to browse to a destination folder and to type in the output file name, as shown in Figure 21. All solution display options will be stored as separate worksheets in an Excel file.

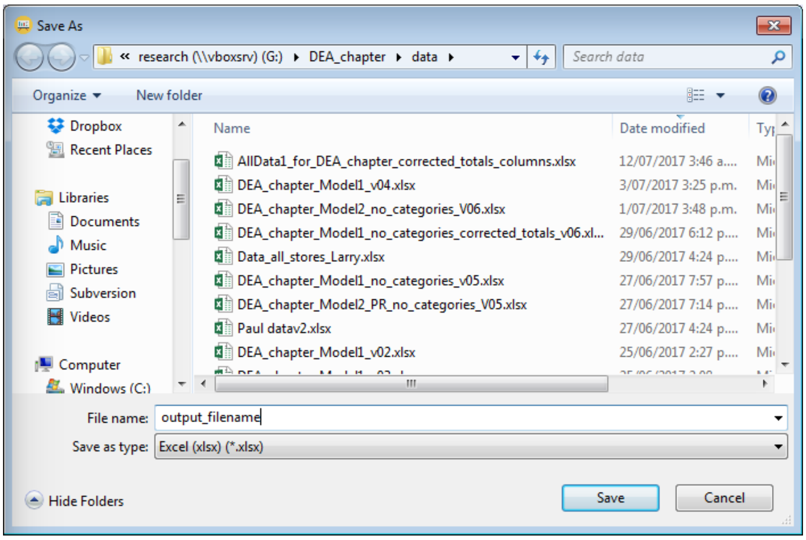


Figure 21: Save As dialogue.

Section 7 discusses the efficiency scores as shown under “EfficiencyScores” in Figure 20, some of which are included in Table 3.

Model 2

Input and output choices for Model 2 are shown in Figure 22.

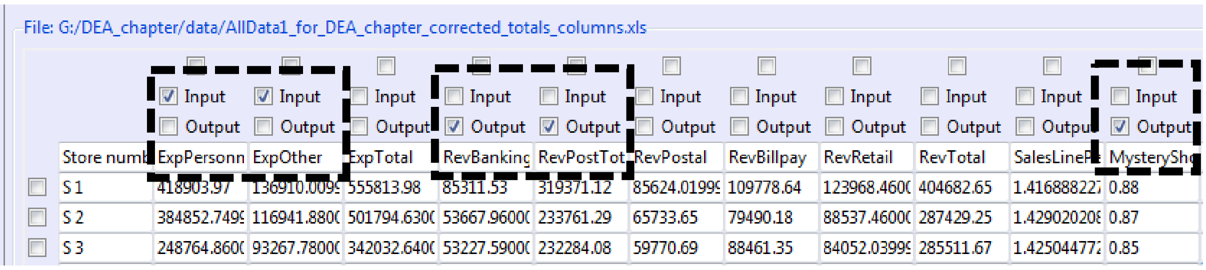


Figure 22: pyDEA input and output selection Model 2.

Model 2 is run with the same parameter choices as Model 1, see Figure 19. Efficiency scores and virtual weight data, as reported in Table 4, are displayed as part of the solution under "WeightedData", see Figure 23.

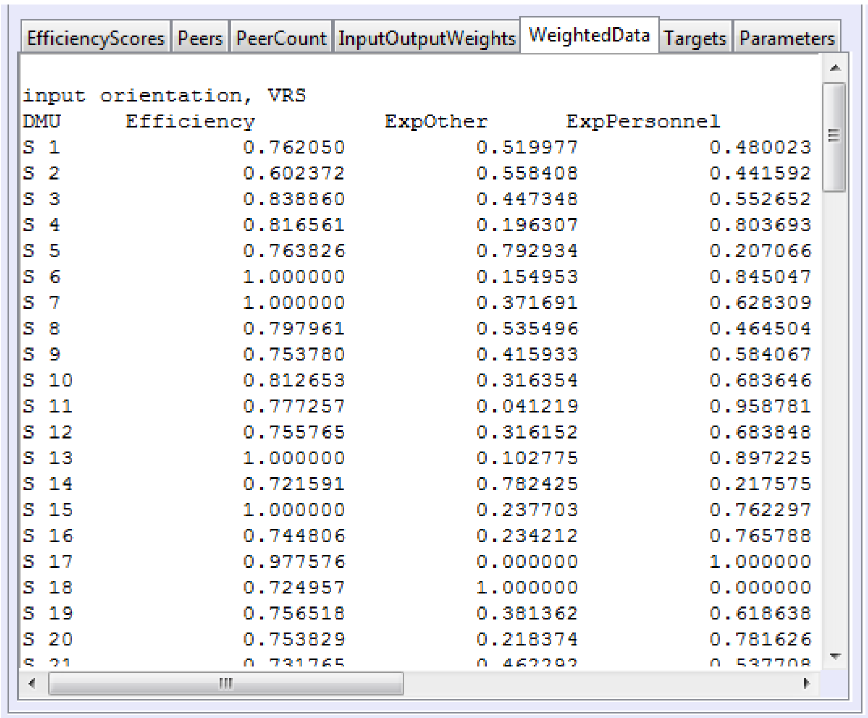


Figure 23: WeightedData tab for Model 2.

Model 2 is also run with categorical variables. To do this, the categories, which are originally rural, satellite urban and urban, need to be re-coded as pyDEA requires a numerical coding where category 1 is the least favourite category (rural stores in the case study), followed by category 2 (satellite urban) and category 3 (urban). Column "LocationCategory" contains this numerical coding. pyDEA will first consider only DMUs with location category 1, then consider categories 2 and 3 (but assess only efficiency scores of category 2 stores), and finally all categories (assessing efficiency of category 3 stores). Inputs and outputs are chosen as before (Figure 22), and the categorical variable is chosen under parameter "Options", as in Figure 24. Only column names that were not selected as input or output already, can be chosen here.

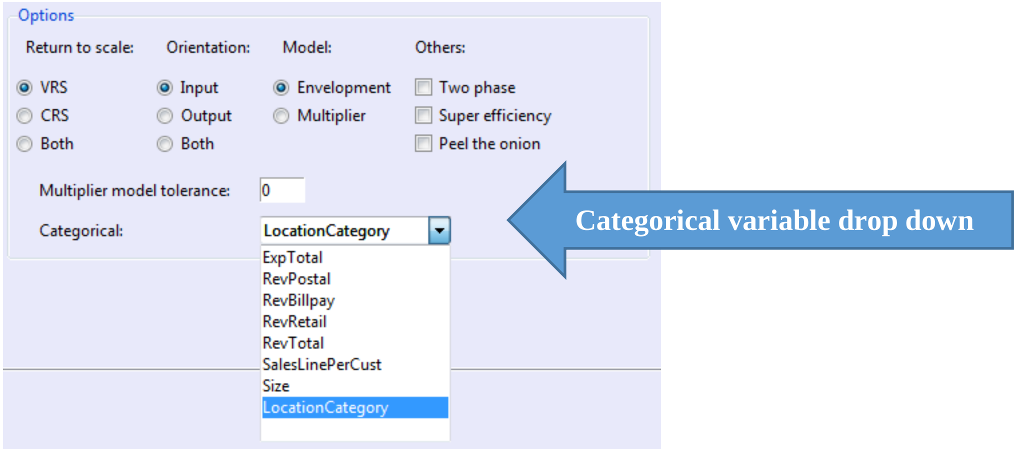


Figure 24: Categorical variable selection for Model 2.

The resulting efficiency scores are now for the categorical model, and categories are also displayed as part of the results, as shown in Figure 25.

```
EfficiencyScores | Peers | PeerCount | InputOutputWeights | WeightedData | Tarc

input orientation, VRS
DMU     Efficiency Categorical: LocationCategory
S 1              0.762050          3.000000
S 2              0.984815          1.000000
S 3              0.838860          3.000000
S 4              0.816561          3.000000
S 5              1.000000          1.000000
S 6              1.000000          3.000000
S 7              1.000000          3.000000
S 8              0.797961          3.000000
S 9              0.753780          3.000000
S 10             0.812653          3.000000
S 11             0.902499          2.000000
```

Figure 25: Model 2 "EfficiencyScores" with categorical variables.

Model 3

Model 3 considers the subset of stores without special banking facilities. The easiest way to select this subset is to prepare a separate Excel worksheet in the input data file which only contains the corresponding 95 stores. This is loaded as described above, and the correct worksheet is chosen (Figure 17). Inputs and outputs are selected as explained for Models 1 and 2, and parameter options are also as for Models 1 and 2 (Figure 19).

The virtual weights reported in Table 6 are from the "WeightedData" solution display, as shown in Figure 26. The VRS model weight v_0 is shown in the solution as column "VRS".

```
EfficiencyScores | Peers | PeerCount | InputOutputWeights | WeightedData | Targets | Parameters

input orientation, VRS
DMU     Efficiency      ExpPersonnel       ExpOther    MysteryShopYTD     RevPostTotal           VRS
S 3           0.720062        0.686571        0.313429        0.000000        0.375634        0.344428
S 4           0.813746        0.793105        0.206895        0.000000        0.481711        0.332035
S 5           0.790978        0.000000        1.000000        0.000000        0.847727       -0.056749
S 6           1.000000        0.849633        0.150367        1.000000        0.000000        0.000000
```

Figure 26: WeightedData tab for Model 3.

Solution display options “Peers” and “PeerCount” give rise to the discussion of peer stores presented for Model 3 in Section 7. As shown in Figure 27, the peers of each DMU are shown, with associated peer weights, under “Peers”.

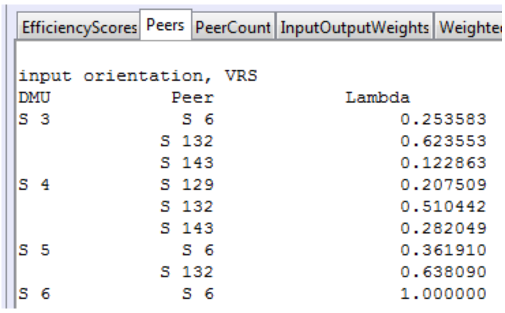


Figure 27: Peers tab for Model 3.

“PeerCount” is shown in Figure 28, and summarised in Table 7. For each DMU the benchmark stores are shown and their corresponding peer weight λ. In Figure 28 we mainly see the peer weight associated with Store 6 which acts as peer for many other stores. When a store is efficient, this store is its only peer with a peer weight of 1, examples are store 6, 13 and 15 in Figure 28. At the end of the PeerCount tab display, the total number of times a store acts as peer for other stores is listed as “Peer count”.

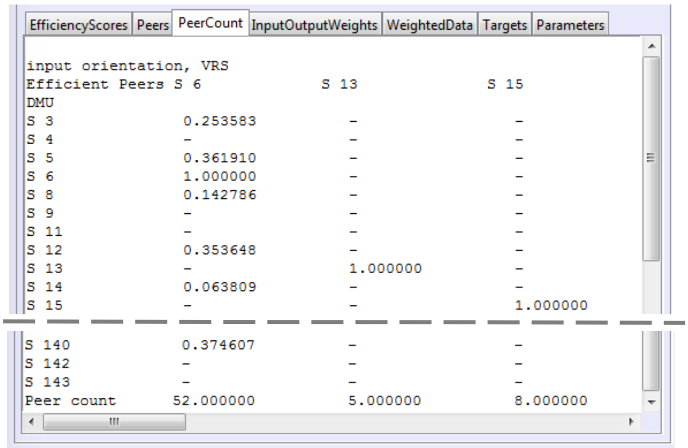
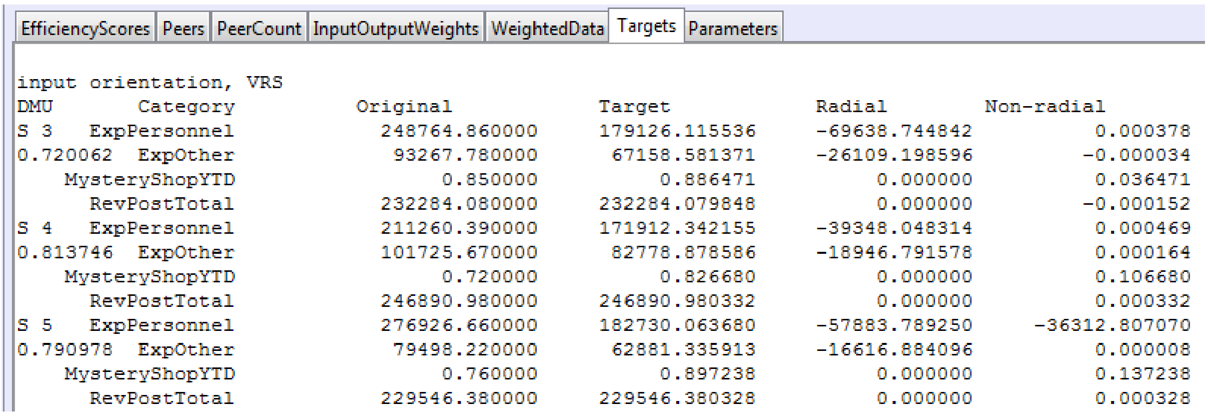


Figure 28: PeerCount tab for Model 3.

The targets listed in Table 8 can be found in `pyDEA` under “Targets”. Targets for stores 3, 4 and 5 are shown in Figure 29. The targets in Table 8 were shown as percentage changes, whereas targets in `pyDEA` are given in absolute terms.



EfficiencyScores | Peers | PeerCount | InputOutputWeights | WeightedData | Targets | Parameters

input orientation, VRS

DMU	Category	Original	Target	Radial	Non-radial
S 3	ExpPersonnel	248764.860000	179126.115536	-69638.744842	0.000378
0.720062	ExpOther	93267.780000	67158.581371	-26109.198596	-0.000034
	MysteryShopYTD	0.850000	0.886471	0.000000	0.036471
	RevPostTotal	232284.080000	232284.079848	0.000000	-0.000152
S 4	ExpPersonnel	211260.390000	171912.342155	-39348.048314	0.000469
0.813746	ExpOther	101725.670000	82778.878586	-18946.791578	0.000164
	MysteryShopYTD	0.720000	0.826680	0.000000	0.106680
	RevPostTotal	246890.980000	246890.980332	0.000000	0.000332
S 5	ExpPersonnel	276926.660000	182730.063680	-57883.789250	-36312.807070
0.790978	ExpOther	79498.220000	62881.335913	-16616.884096	0.000008
	MysteryShopYTD	0.760000	0.897238	0.000000	0.137238
	RevPostTotal	229546.380000	229546.380328	0.000000	0.000328

Figure 29: Targets tab for Model 3.

Finally, weight restrictions are added in pyDEA via the weights editor in the parameter section. A screenshot of the weights editor is shown in Figure 30. The weights editor has separate sections for absolute, virtual and ratio weights. The two virtual weights requiring the customer satisfaction score to be between 0.10 and 0.30 are show in Figure 30. Weight restrictions are entered as free text, based on the name of the input or output column and the restriction, such as "`>= 0.1`". The "Validate weight restrictions" button checks weight restrictions for typos. If there is an input error, the corresponding weight will be highlighted in red, see for instance Figure 31. Care needs to be taken with weight restrictions as they may render DEA models infeasible, which editor does not check.

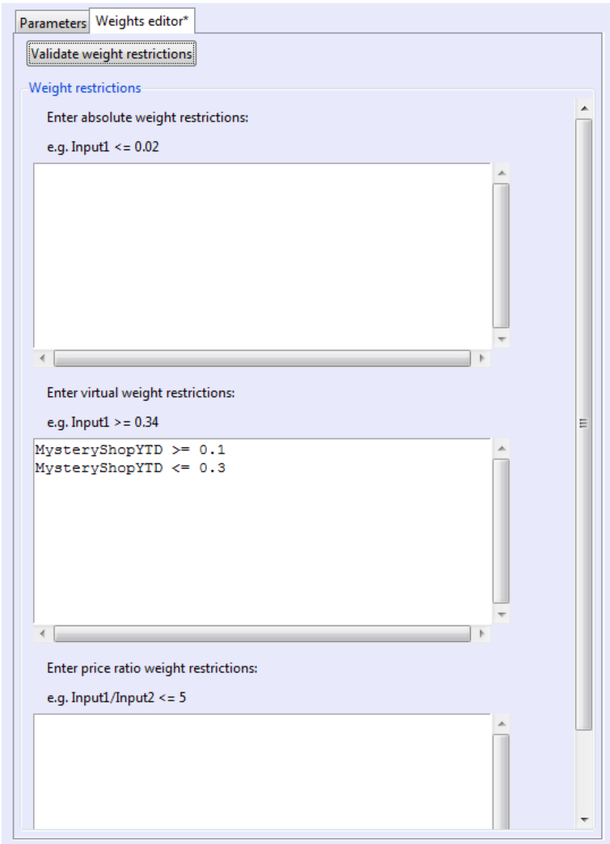


Figure 30: Weights Editor and weights for Model 3.

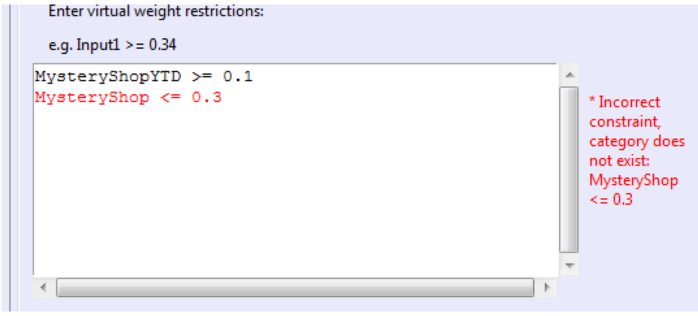
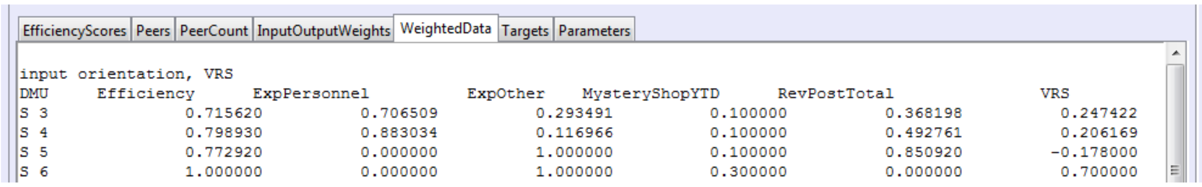


Figure 31: Example of incorrect weight and error message.

The resulting solution now respects the weight restriction. This is best seen on the WeightedData display, where the weights for MysteryShopYTD are now between 0.10 and 0.30 (Figure 32).



input orientation, VRS

DMU	Efficiency	ExpPersonnel	ExpOther	MysteryShopYTD	RevPostTotal	VRS
S 3	0.715620	0.706509	0.293491	0.100000	0.368198	0.247422
S 4	0.798930	0.883034	0.116966	0.100000	0.492761	0.206169
S 5	0.772920	0.000000	1.000000	0.100000	0.850920	-0.178000
S 6	1.000000	0.000000	1.000000	0.300000	0.000000	0.700000

Figure 32: WeightedData tab for Model 3.

BIBLIOGRAPHY

Ali, A. & Seiford, L., 1993. Computational accuracy and infinitesimals in Data Envelopment Analysis. *INFOR: Information Systems and Operational Research*, 31(4), pp. 290-297.

Allen, R., Athanassopoulos, R., Dyson, R. & Thanassoulis, E., 1997. Weight restrictions and value judgements in Data Envelopment Analysis: Evolution, development and future directions. *Annals of Operations Research*, 73(0), pp. 13-34.

Banker, R., Charnes, A. & Cooper, W., 1984. Some models for estimating technical and scale inefficiencies in Data Envelopment Analysis. *Management Science*, 30(9), pp. 1078-1092.

Banker, R. & Morey, R., 1986. Efficiency analysis for exogenously fixed inputs and outputs. *Operations Research*, 34(4), pp. 513-521.

Banker, R. & Morey, R., 1986. The Use of categorical variables in Data Envelopment Analysis. *Management Science*, 32(12), pp. 1613-1627.

Barr, R., Durchholz, M. & Seiford, L., 2000. *Peeling the DEA onion: layering and rank-ordering DMUs using tiered DEA*, Dallas, TX: Southern Methodist University.

Charnes, A. & Cooper, W., 1962. Programming with linear fractional functionals.. *Naval Research Logistics Quarterly*, 9(3-4), pp. 181-186.

Charnes, A., Cooper, W. & Rhodes, E., 1978. Measuring the efficiency of decision making units. *European Journal of Operational Research*, 2(6), pp. 429-444.

Coelli, T., Rao, D., O'Donnell, C. & Battes, G., 2005. Chapter 6. In: *An introduction to efficiency and productivity analysis*. s.l.:Springer Science & Business Media.

Cooper, W., Seiford, L. & Tone, K., 2007. Data Envelopment Analysis: A Comprehensive Text with Models, Applications, References and DEA-Solver Software. In: *Chapter 2.* New York: Springer.

Dyson, R. et al., 2001. Pitfalls and protocols in DEA. *European Journal of Operational Research,* 132(2), pp. 245-259.

Emrouznejad, A. & Yang, G.-I., 2017. A survey and analysis of the first 40 years of scholarly literature in DEA. *Socio-Economic Planning Sciences,* Volume 61, pp. 1978-2016.

Hosseinzadeh Lotfi, F. et al., 2008. An MOLP based procedure for finding efficient units in DEA models. *Central European Journal of Operations Research,* 17(1), pp. 1-11.

Leleu, H., 2006. A linear programming framework for free disposal hull technologies and cost functions: Primal and dual models. *European Journal of Operational Research,* 168(2), p. 340–344.

Lovell, C. & Rouse, A., 2003. Equivalent standard DEA models to provide super-efficiency scores. *Journal of the Operational Research Society,* 54(1), pp. 101-108.

Olesen, O. & Petersen, N., 2016. Stochastic data envelopment analysis—a review. *European Journal of Operational Research,* 251(1), pp. 2-21.

Priddey, H. & Harton, K., 2010. *Comparing the efficiency of stores at New Zealand post. Proceedings of the 45th Annual Conference of the ORSNZ.* Auckland, New Zealand, s.n.

Seiford, L., 1990. Models, extensions, and applications of data envelopment analysis. *Computers, Environment, and Urban Systems*, 14(2), pp. 171-175.

Seiford, L. & Zhu, J., 1999. An investigation of returns to scale in data envelopment analysis. *Omega*, 27(1), pp. 1-11.

Yougbaré, J. & Teghem, J., 2007. Relationships between Pareto optimality in multi-objective 0–1 linear programming and DEA efficiency. *European Journal of Operational Research,* 183(2), pp. 608-617.

Yu, G., Wei, Q., Brockett, P. & Zhou, L., 1996. Construction of all DEA efficient surfaces of the production possibility set under the generalized data envelopment analysis model. *European Journal of Operational Research,* 95(3), pp. 491-510.