Management Science and Information Systems Studies

Final Year Project Report

Performance Appraisal Scheduling for Deloitte

Ciarán Doyle

March 2015
DELOITTE
Performance Appraisal Scheduling

March 2015

Prepared by: Ciarán Doyle  Supervisor: Prof. Frank Bannister
DECLARATION

I declare that the work described in this dissertation has been carried out in full compliance with the ethical research requirements of the School of Computer Science and Statistics.

Signed: ___________________

Ciarán Doyle

March 2015
This report describes the development of a new performance appraisal scheduling process for Deloitte Touche Tohmatsu Limited, commonly referred to as Deliotte. The aim of this project was to manipulate the company’s existing scheduling process in order to provide a process that is faster, more efficient and scalable to meet Deloitte’s growing workforce. A combination of R, Microsoft Excel and Visual Basic was used to test potential algorithms, manipulate spreadsheets and create an application that is flexible and extendable for the company. This application fully meets the company’s requirements.
This project was initiated on behalf of Deloitte, an international professional services firm. Deloitte provides audit, consulting, financial advisory, risk management, and tax services to selected clients.

This project was specifically concerned with Deloitte's Analytics department in Dublin which, due to growth in recent years, has found its performance appraisal scheduling process increasingly difficult to manage. The purpose of the project was to modify the existing process to efficiently handle the increased number of employees in the Analytics department. A generic solution was required so it could be applied to all departments in Deloitte if successful. During the course of the project, several algorithms were created and tested using R statistical software package. One of these algorithms was then chosen to be implemented in a Microsoft Excel application that was created using Visual Basic.

The complexity of this problem far exceeded expectations and was a great test of my technical skills but also my creative competence and perseverance. These skills were paramount when developing countless algorithms that failed more often than they succeeded. This adversity only made finding a solution more rewarding.

I would like to express my sincere gratitude to my project supervisor, Professor Frank Bannister, without whom this project could not have been possible.

I must also express my appreciation and thanks to my client’s, David and Ciarán, who gave me as much time as I needed and guided me in the right direction.

A special thank you is extended to my colleague Tom, who made working the many long nights achievable.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>NO.</th>
<th>SECTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>INTRODUCTION AND SUMMARY</td>
<td>1</td>
</tr>
<tr>
<td>1.1</td>
<td>The Client Company</td>
<td>1</td>
</tr>
<tr>
<td>1.2</td>
<td>Project Background</td>
<td>1</td>
</tr>
<tr>
<td>1.3</td>
<td>Terms of Reference</td>
<td>2</td>
</tr>
<tr>
<td>1.4</td>
<td>Achievement of Objectives</td>
<td>2</td>
</tr>
<tr>
<td>1.5</td>
<td>Summary</td>
<td>2</td>
</tr>
<tr>
<td>2.</td>
<td>CONCLUSIONS AND RECOMMENDATIONS</td>
<td>3</td>
</tr>
<tr>
<td>2.1</td>
<td>Conclusions</td>
<td>3</td>
</tr>
<tr>
<td>2.2</td>
<td>Recommendations</td>
<td>4</td>
</tr>
<tr>
<td>3.</td>
<td>STRUCTURE AND SCOPE OF PROJECT</td>
<td>5</td>
</tr>
<tr>
<td>3.1</td>
<td>Terminology</td>
<td>5</td>
</tr>
<tr>
<td>3.2</td>
<td>Standards of Appraisal</td>
<td>6</td>
</tr>
<tr>
<td>3.3</td>
<td>The Current Process</td>
<td>6</td>
</tr>
<tr>
<td>3.4</td>
<td>Objectives</td>
<td>8</td>
</tr>
<tr>
<td>3.5</td>
<td>Sample Data</td>
<td>8</td>
</tr>
<tr>
<td>3.6</td>
<td>Methodology</td>
<td>9</td>
</tr>
<tr>
<td>3.7</td>
<td>Software Selection</td>
<td>10</td>
</tr>
<tr>
<td>3.8</td>
<td>Assumptions</td>
<td>11</td>
</tr>
<tr>
<td>3.9</td>
<td>Constraints</td>
<td>11</td>
</tr>
<tr>
<td>4.</td>
<td>DESCRIPTION OF WORK DONE</td>
<td>13</td>
</tr>
<tr>
<td>4.1</td>
<td>Requirements Phase</td>
<td>13</td>
</tr>
<tr>
<td>4.2</td>
<td>Algorithms Phase</td>
<td>13</td>
</tr>
<tr>
<td>4.3</td>
<td>Testing Phase</td>
<td>18</td>
</tr>
<tr>
<td>4.4</td>
<td>Scheduling Application Phase</td>
<td>18</td>
</tr>
</tbody>
</table>
5. RESULTS AND FUTURE WORK

5.1 Algorithm Selection 22
5.2 Process Evaluation 22
5.3 Further Work 23

APPENDICES

<table>
<thead>
<tr>
<th>NO.</th>
<th>CONTENT</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Original Project Report</td>
<td>A.1</td>
</tr>
<tr>
<td>B</td>
<td>Interim Report Term 1</td>
<td>B.1</td>
</tr>
<tr>
<td>C</td>
<td>Interim Report Term 2</td>
<td>C.1</td>
</tr>
<tr>
<td>D</td>
<td>Project Clarifications 1</td>
<td>D.1</td>
</tr>
<tr>
<td>E</td>
<td>Project Clarifications 2</td>
<td>E.1</td>
</tr>
<tr>
<td>F</td>
<td>Log Book</td>
<td>F.1</td>
</tr>
<tr>
<td>G</td>
<td>Results</td>
<td>G.1</td>
</tr>
<tr>
<td>H</td>
<td>Code</td>
<td>H.1</td>
</tr>
</tbody>
</table>

REFERENCES

USER MANUAL
1. INTRODUCTION AND SUMMARY

This chapter describes the client’s company, project background, terms of reference, achievement of objectives, and summary of the remaining chapters of the report.

1.1 The Client Company

Deloitte is the brand under which tens of thousands of dedicated professionals in independent firms throughout the world collaborate to provide professional services to selected clients. These services include audit, consulting, financial advisory, risk management, and tax. In Ireland, the Deloitte Analytics department provides solutions to a wide variety of data, technology and business challenges. In Dublin, there are over 250 employees in Deloitte’s Analytics department and this number is growing year on year as companies begin to rely more on data when it comes to making decisions. This project is concerned with providing the Analytics Department with an improved process for organising their employee’s performance appraisal meetings, which can then be used for the remaining departments in the company.

1.2 Project Background

Twice a year, Deloitte undertakes a performance appraisal process for all staff. This process is intended to both evaluate performance and give employees the opportunity to discuss their personal and career objectives with their managers, directors, and Deloitte partners. Currently, an employee meets with his/her assigned appraiser (generally their supervisor) and, after discussing the employee’s performance and objectives, they are given a provisional rating out of 5. In the appraisal meeting, which is comprised of all members of staff above the level of the employee, the appraiser represents the employee by presenting to the congregation the rating and reasons why they have given the employee that rating. Anyone in the room that has worked with the employee can then give input as to whether they agree or disagree with the rating until a final rating is agreed. This process is done one level at a time so when all employees at the lowest level have been appraised, the second lowest level will be asked to leave the room and they will be appraised by all levels above them. This process continues to the Partner level. Therefore the employee being appraised will never be present in the room when they are being discussed.

The size of Deloitte’s workforce in Dublin is growing and has reached a point where organising these appraisal meetings is becoming increasingly hard to manage both in terms of time and scalability. The aim of this project was to take the current process and improve it in these two aspects; scalability and time.
1.3 Terms of Reference

Deloitte plans to double in size in the coming years and needs to improve its current process for organising its performance appraisal meetings as the existing one is becoming impractical. Below are the terms of reference as detailed in the Interim Report Term 1 (Appendix B, p. B.1):

- Improve/alter the current process in order to handle the larger numbers more efficiently.
- Propose a generic solution that can be applied twice-yearly to all departments.
- Ensure that the solution is scalable.

1.4 Achievement of Objectives

All objectives set out in the terms of reference were achieved for the company. The process’ efficiency was improved based on the accuracy metric defined in Section 3.6, p. 9. The solution that is proposed is generic and can be applied to all departments. It can cater for a varying workforce size and is adaptable to the company’s needs. The solution proposed is also scalable: any department can be split into a user specified number of groups allowing the solution to cope with departments of any size. These functions extend beyond the client’s primary requirements.

1.5 Summary

- Chapter 2 details the conclusions and recommendations of the report.
- Chapter 3 defines the structure and scope of the project.
- Chapter 4 describes the work carried out throughout the projects duration.
- Chapter 5 sets out the main results and future work.
2. CONCLUSIONS AND RECOMMENDATIONS

This chapter sets out the conclusions and recommendations relating to the work carried out on this project.

2.1 Conclusions

Key Findings:

It is possible to significantly improve the efficiency of the way the performance appraisal meetings are organised, reducing both the numbers in each meeting and the overall time the process takes. Currently the groups are split by department but some departments have grown substantially in size and the process is taking longer and becoming harder to manage. A temporary solution that is currently being used is to split any large department in half so as two smaller, more manageable groups are formed. By splitting the groups several more times, more meetings can be held simultaneously thus shortening the time it takes to get through every individual employee that needs to be appraised. By splitting large groups into smaller ones, the accuracy in these groups is reduced. Accuracy is dependent on having managers that have worked with as many of the employees being appraised, in the same groups.

Several algorithms for doing this were examined. Of those examined, the ‘Maximum Proportions’ algorithm, detailed in Section 4.2, p. 13, was found to give the best results. Due to the large quantity of employees in Deloitte, a Microsoft Excel application was created using Visual Basic to allow the user to easily view the groups that have been created. This application applies the algorithm proposed to split larger groups into a specified number of smaller groups.

Out of all the algorithms that were tested, none could find an optimum solution as this is an NP-hard permutations problem. For this reason, coding by exception was necessary which gives the user discretion when the application’s output is conflicting with constraints (Section 3.9, p. 11). This allows for small changes to be made by the user which can improve on the recommendations made by the application.

While tests indicate a collective average improvement in accuracy of 72.8% (Section 5.2, p. 22) by using the new process, it is probable that further improvements may be possible using other algorithms. The use of Genetic Algorithms for scheduling problems is widely accepted as effective. There was insufficient time to generate and test algorithms such as this.

It was found that the existing process is capable of performing as required for the short term but it may be less accurate and more time consuming than implementing the new process.
2.2 **Recommendations**

The recommendations are as follows:

**Modifications to the Existing Process:**

The company should make the following modifications to the way it organises its appraisal meetings:

- Issue a company-wide survey to determine which employees have worked with whom. This gathered data should be tabularised in the same format as the sample data given for this project. Appraisers should be assigned to employees that are in the same department in advance of the appraisal meetings.

- Import the collected data into the application proposed in this report. The application can then determine any issues in the collected data and sort it into initial groups.

- Apply the built in sorting algorithm to each group’s level that is deemed too large by the user. Large groups can be split into a specified number of smaller groups. The aim of this is to have several meetings happening simultaneously thus getting through everyone that needs to be appraised faster whilst maintaining an acceptable level of accuracy in each meeting.

- When possible, all of the subgroups should hold their meeting simultaneously in the same room. This allows for quick swaps between groups when necessary which can substantially increase the accuracy of the groups. Swaps occur when a manager in one group moves to a different group to give appraisal on someone he/she has worked with. This can happen often depending on the level of accuracy desired by the company. If the company wishes to achieve 100% accuracy, it is highly improbable that this can be accomplished without swaps.

**Implementation:**

The company should implement the modified appraisal scheduling process at the next appraisal meeting in the Analytics Department in October 2015. This will allow the new process to be fully compared to the existing process. Any issues that may arise from this test can then be adjusted before the process is used in all other departments.
3. STRUCTURE AND SCOPE OF PROJECT

This chapter outlines the structure and scope of the project.

3.1 Terminology

This section explicitly defines a number of terms that are used throughout the report to categorise different groups. Colour codes and other terms used in the application are also defined.

- **“Junior”** categorises a staff member that is being appraised. The columns in the application output represent Juniors.

- **“Senior”** refers to a staff member that is eligible to give appraisal on a Junior. A staff member is eligible to give appraisal on employees below their staff level. Rows in the application output represent Seniors.

- **“Appraiser”** refers to the manager that is assigned to a Junior to represent them in the appraisal meeting. Every Junior is assigned exactly one Appraiser.

- **“Control”** describes a Senior that is present during a Junior's appraisal presentation but has not worked with that Junior before.

For the tabular output of the application:

- **Yellow Colour Coding** – Recommended groups to hold the appraisal meetings in are marked in yellow. Each of these groups represents a separate meeting.

- **Orange Colour Coding** – Within each group marked in yellow, it is possible to have sub groups. These subgroups are marked in orange. Sub groups are ideally in the same room as their parent group, to allow for quick swaps of seniors between the subgroups.

- **Red Colour Coding** – In the margins of the output, any cells marked in red represent conflicts. These conflicts can be with criteria from the problem or recommendations by the application.

- **Green Colour Coding** – Appraisers are marked in green. All green marks should be in a group.

- **Black Colour Coding** - This represents a work relationship between a Senior and a Junior. Any instance that the senior has worked with the Junior is classed as a work relationship.
3.2 Standards of Appraisal

Performance appraisal meetings are intended to both evaluate an employee’s performance and give an employee the means to voice their personal and career objectives to their managers. Employees can be appraised by any staff member that is of a higher employment level (a ‘Senior’). The levels of employment in Deloitte in order of seniority are: Analyst, Consultant, Senior Consultant, Manager, Senior Manager, Director and Partner. Therefore, if a Manager is being appraised (the ‘Junior’) they can be appraised by Senior Managers, Directors and Partners (the ‘Seniors’ of this Junior). If an Analyst is being appraised, they can be appraised by anyone that is more senior than an Analyst. The ideal appraisal meeting would have every Senior that has worked with the Junior present in the meeting. Having 70% present is acceptable and swaps (Section 3.6, p. 9) can be introduced to increase this if necessary. It is also important to have a number of Controls in the meeting. These are Seniors that have not worked with the Junior. This ensures that there is an equal standard of appraisal for all members of staff.

3.3 The Current Process

The current process is used on each department separately. It attempts to meet the standards of appraisal discussed in Section 3.2, p. 6, through the following steps:

- The Junior is assigned an Appraiser who has worked with the Junior before and is generally two levels above the Junior.

- The Junior meets with assigned Appraiser prior to the appraisal meeting. This meeting is the Junior’s opportunity to openly discuss the work they have done since the last appraisal meeting, and their personal and career objectives.

- The Junior is given a performance rating by the Appraiser out of 5, with 1 being a very poor performance and 5 being an excellent performance.

- A single slide is created with the Junior’s name, rating and overview of what was discussed between the Junior and the Appraiser.

- Slides are collected and merged into one PowerPoint presentation that is put in increasing order of employment level.

- All staff members in the department, except the lowest level (Analyst), assemble in one room. The PowerPoint is started and each Appraiser of the Junior on each slide stands and gives a short two minutes presentation justifying the Junior’s rating.

- Three minutes is then dedicated to open discussion of who agrees or disagrees with the rating and reasons why.
• The rating is either adjusted or finalised.

• After the slides for each Analyst are finished, the Consultants are asked to leave the room and their slides will be run through in the same fashion.

• This process continues to Partner level

Once the rating for every Junior is finalised, decisions can be made as to who can be promoted or given more responsibility. This is done manually as there are constraints that must be met in order to be promoted; such as experience.

Evaluation:

The accuracy metric used to evaluate the process is based on three variables: overall average percentage, minimum percentage and number of swaps. These variables are explained in Section 3.6, p. 9. The existing process works well and allows for a high accuracy. This high accuracy is attributable to the large size of the groups. If everyone in the department is present at one meeting, then the majority of the employees that can offer appraisal will be present. The accuracy for the existing process with the given data was 41% overall average with a 0% minimum. This does not consider swaps for now. These accuracy figures may be unrealistically low because of shortcomings in the sample data. The sample data is discussed in detail in Section 3.5, p. 8.

• Firstly, it does not consider the assumption made that all Appraisers are assigned to Juniors in the same department (Section 3.8, p. 11).
• Secondly, the likelihood of two employees to work in the same department is assumed to be greater than the sample data considers.

For these reasons, it is highly improbable for there to be a case where there is a 0% minimum. This would mean that a Junior has not worked with any other staff member from the same department. For the same reasons, the 41% average is also assumed to be lower than expected for real data. The data which these figures are calculated from is the same data that is used in testing the new process and therefore the new process will have equally worse results than with real data. A comparison between the two processes is discussed in Section 5.2, p. 22.

Limitations:

The limitations associated with the existing process are the main motivation for a new process to be implemented.

The primary limitation of the current process is practicality. With an increasing number of employees in Deloitte, fitting the staff from an entire department into one room is becoming harder to manage. It is also taking longer to get through all staff members that need to be
appraised. The company is currently handling this problem by splitting any large departments in half. No process is in place for doing this effectively. This significantly reduces accuracy and forces the groups to swap members constantly in order to achieve a level of accuracy that is acceptable by the standards of appraisal discussed in Section 3.2, p.6.

The current process omits the input of employees that have worked with an individual but are from a different department. It is not likely that this will have great effects on accuracy but some employees may have worked on large projects that involved people from various departments.

The larger groups are resulting in the process taking longer. More senior levels are particularly affected by this as they must be present for the most number of meetings. Splitting large groups in half is a temporary workaround to this. Ideally they would need to be split several times further.

3.4 Objectives

Primary:

The primary objectives of this project are to:

- improve the current process in order to handle the larger numbers more efficiently.
- propose a generic solution that can be applied twice-yearly to all departments.
- ensure that the solution is scalable.

Secondary:

The secondary objectives are to:

- propose a solution that is adaptable and extendable.
- investigate existing algorithms with the potential for finding an absolute solution to the problem.
- create an application that is easy to use and displays the data in an intuitive format.

3.5 Sample Data

Sample Link Matrix.csv

A mock data set was given in the form of a binary matrix where the digit ‘1’ represented a work relationship between a Senior and a Junior and the digit ‘0’ filled the remaining cells in the matrix. This data was called “Sample Link Matrix” and was presented in a .csv format.
Sample Staff List.csv

A mock list of Juniors and their corresponding Appraisers was given. This was called “Sample Staff List” and was also presented in .csv format.

Limitations

A number of limitations/faults with the sample data were noticed:

- A list of partners was not available and therefore was excluded from both files. The only consequence of this is that Directors (the next level down from Partner) had no one to contribute to their appraisal.

- There is a weighting applied to data so employees from the same department are more likely to work with each other. However, after analysing the data it was clear that this weighting was not heavy enough.

- Other than the weight applied to departments, the data was randomly generated meaning any natural group structures that could exist from employees working on the same projects do not exist in the sample data.

- The “Sample Staff List” had been generated randomly and thus the majority of the Appraisers assigned to Juniors were from different departments. This had to be edited to reflect more realistic data.

- As a result of the list of Appraisers being separate to the matrix data, code had to be written to insert a ‘9’ in the place of an appraiser in the matrix data. ‘9’ was chosen so it could act as a weight to give appraisers priority over normal work relationships.

3.6 Methodology

The majority of the analysis for this project was carried out manually and by using the statistical software package, R. The final solution is implemented in Excel using Visual Basic.

Approaching the problem:

After all of the client’s requirements were gathered, it was decided that the best approach for meeting the terms of reference was to sort the data matrix into its natural groups, the departments. Then take for sub-groups within the departments so several meetings could be held simultaneously. To do this, an algorithm needed to be created that could take a large group and split it into a specified number of smaller groups whilst maintaining a level of accuracy that meets the standards of appraisal discussed in Section 3.2, p.6. Several
algorithms were created and tested in R. An algorithm was then picked and implemented in a Microsoft Excel application.

**Accuracy:**

Accuracy is measured off three conditions:

- **Average Percentage** of seniors that are present in the junior’s appraisal meeting out of the total managers they have worked with.

- **Minimum Percentage** is the junior with the minimum percentage of seniors present.

- **Number of Swaps** – This is the number of times a senior has to change group in order for the process to reach an acceptable average percentage that complies with the standards of appraisal discussed in Section 3.2, p. 6.

### 3.7 Software Selection

A combination of R statistical software, Microsoft Excel and Visual Basic was used throughout the duration of this project.

**R:**

R is a programming language and software environment for computational statistics and graphics. The libraries in R contain a wide range of data manipulation, statistical modelling and graphical plots that made it an obvious tool to use when presented with the original sample data. The sample data was presented as a binary matrix in a .csv format. This made it easy to read into R and once the correct libraries for manipulating matrices were identified, the algorithms which had been created could be tested. The built in statistical models could also be investigated to see if there were any existing algorithms for similar problems.

**Microsoft Excel:**

Microsoft Excel had many benefits to this project. It was chosen for the main implementation of the projects solution because of its ease in manipulating spreadsheets. The results can be visually represented well in an easily understood format. Microsoft Excel is universally used in personal and working lives. This means that most people have some knowledge of how to use it and does not require someone with prior training to understand the solutions output. It therefore can be used by the HR department to perform analytical operations for creating the groups required for good appraisal meetings.
Visual Basic:

Visual Basic is a programming language that can be used to manipulate spreadsheets in Microsoft Excel. It was used in this project to create form controls, manipulate the data and execute the algorithm used to sort the data into groups.

3.8 Assumptions

Some assumptions were made when creating algorithms for this problem. They are detailed below:

Natural Groups:

It is assumed that the likelihood of a Junior working with senior from the same department is far greater than the sample data has allowed for. This allows for well-defined natural grouping by department and a reduction in the noisiness of the sample data. It is also assumed that there is an increased likelihood of a Junior working with someone closer to his/her employment level. This is not reflected in the sample data.

Appraisers:

It is assumed that all Appraisers are assigned to Juniors in the same department. The sample data does not reflect this.

3.9 Constraints

There are a number of constraints associated with a solution to the problem. This section sets out these constraints in detail.

Appraisers:

A Junior’s assigned Appraiser must be present in the meeting that is discussing the Junior. The Appraiser is the person that is to represent the junior at his/her appraisal meeting and therefore the meeting cannot take place without the Appraiser.

Group Size:

One of the current impracticality issues with the existing process is the size of the groups. The largest department contains 88 staff members. For all the algorithms that ordered the rows and columns first and then determined the groups, a default maximum group size was set to 40 people. However, multiple groups can be in the same room if desired. The purpose of this is to allow for quick and easy swaps between groups if they are necessary. A default maximum room capacity was set to 100. This means that two groups of 40 can be holding
the appraisal meeting in the same room and there will still be room for a group of 20. The algorithm that was implemented in the final solution allows the user to determine the number and size of the groups he/she creates, thus a constraint for this was not necessary.

Accuracy:

There is a desired level of accuracy based on the accuracy metric discussed in Section 3.6, p. 9. This desired level is set to a minimum of 70%. This means swaps will have to be applied to the group that has any Juniors with less than 70% of the Seniors he/she has worked present. There is no constraint associated with the number of swap or minimum percentage. This is because the minimum percentage will be increased by increasing the number of swaps. The aim is to have as few swaps as possible.
4. DESCRIPTION OF WORK DONE

4.1 Requirements Phase

Several meetings with the client were held at the beginning of this project. These were an integral part in determining the client’s requirements and the scope of the project. Clarification documents were exchanged which were of great use in determining exactly what the limitations of the current process are. (Appendix D, p. D.1 and Appendix E, p. E.1) A log book was kept during the requirements phase to ensure that nothing was overlooked when determining exact terms of reference. (Appendix F, p. F.1)

4.2 Algorithms Phase

There were several algorithms that were created and tested throughout the duration of this project. This section details the algorithms that made it to the testing phase, which is described in Section 4.3, p 18. Many other algorithms were created and tested by hand on smaller matrices but were deemed inefficient before making it to the testing phase. Each algorithm discussed in this section concentrates on finding a permutation of rows and columns of the sample data that creates a matrix with a near block diagonal structure. An example of this can be seen in Figure 4.2.1. All of these algorithms were original ideas or were original ideas that were later discovered in the literature.

Some of the algorithms in this section first classify the groups and then reorder the rows and columns to fill these predetermined groups. For these algorithms, groups are determined by dividing the rows and columns by the number of groups desired. For example, if 3 groups were needed from a 12 by 9 matrix, the resultant groups would be three 4 by 3 matrices. This can be seen in Figure 4.2.1.

Other algorithms order the rows and columns first and then determining where the groups (or blocks) are after. For these algorithms a separate grouping algorithm must be applied after the initial ordering algorithm. This grouping algorithm uses two constraints set by the user. The first constraint is the number of groups and the second constraint is how much variance there can be in the group sizes. The algorithm then looks at all possible groupings within these two constraints and decides which is best based on the highest percentage of ‘1’s captured by the grouping. This grouping algorithm is referred to by several of the algorithms discussed in this section.

FIGURE 4.2.1
Barycentric Algorithm (Average position of ‘1’s)

The Barycentric algorithm attempts to order the matrix and then applies the grouping algorithm that was discussed in the introduction of this Section 4.2, p.13. The Barycentric algorithm operates on the rows of the matrix only meaning the rows will change places but the columns will remain in a fixed permutation. Randomising the permutation of the columns is applied before applying the algorithm.

In each row, the average column position of the ‘1’s is known as that row’s Barycentre. For example, to calculate the barycentre for row ‘M1’ in Figure 4.2.1 the positions of each ‘1’ in the row (1 and 2) are added and then divided by the total number of ‘1’s in the row, \((1+2) \div 2 = 1.5\). For ‘M2’ the calculation is \((3 + 5 + 6) \div 3 = 4.6\).

The Barycentric algorithm follows the following steps:

- Calculate Barycentre for all rows.
- Reorder rows based on increasing Barycentre. Figure 4.2.3
- Transpose matrix.
- Repeat all previous steps till convergence.
- Apply grouping algorithm discussed in the introduction of this Section 4.2, p. 13.

Using an input of 3 for the number of groups and 1 for the variance in group size, the final output of the Barycentric algorithm for the example given can be seen in Figure 4.2.4.
Diagonal Dominance (Grouping Along the Diagonal)

The Diagonal Dominance algorithm attempts to order the matrix first and then applies the grouping algorithm that was discussed in the introduction of this Section 4.2, p. 13. A diagonally dominant matrix is a square matrix where the majority of its data lies on the main diagonal (Top left to bottom right). The sample data is not in the form of a square matrix but the concept of achieving diagonal dominance could still be applied. By treating the data like and x, y plane with the base of the matrix being the x-axis and the left side being the y-axis, it was possible to use rudimentary geometry to find the equation of the line that represented the main diagonal as seen in orange in Figure 4.2.5. Once the equation of this line was known it was possible to calculate the perpendicular distance from each data entry in the matrix to the line, as seen in red in Figure 4.2.5.

The best solution is considered the one with the shortest overall distance to the main diagonal. The steps of the Diagonal Dominance algorithm were:

- Randomise row and column permutations.
- Calculate perpendicular distance from each data entry to the main diagonal.
- Sum these calculations.
- If sum of distances is less than previous distance recorded then store row and column permutations and the distance recorded.
- Repeat all steps until acceptable best distance is found.
- Apply grouping algorithm.

Maximum Proportions Algorithm

The proportions algorithm splits the matrix into a specified number of groups. For each row, it calculates the percentage of ‘1’s that are in each group. Rows are then reassigned to different groups based on the group with the highest percentage. If percentages are the same for two groups, the smallest group is selected. An example of this process is detailed below using figures:

Step 1: For each row, the percentage in each group is calculated. In row ‘M1’ there are five ‘1’s. One is in group 1, two are in group 2 and two are in group 3. Therefore 20% is in group 1, 40% in group 2 and 40% in group 3. These calculations are stored in the table to the right of the matrix under the heading “Groups”. Similar calculations are made for all rows. For each row, the group with the highest percentage is highlighted in red unless that is the group
the row is already in. Yellow represents the groups. Based on the group with the highest percentage, a recommendation is made as to what group the given row should be in. These recommendations are under the heading “Recommend”. If it is recommended for a row to change groups, the recommendation is highlighted in red.

**FIGURE 4.2.6**

Step 2: The rows are reordered using the recommended swaps. If there are conflicting recommendations then preference is given to the smallest group.

**FIGURE 4.2.7**

Step 3: The matrix is then transposed and the algorithm is applied again. These steps are repeated until no more changes occur or the user is happy with the

**FIGURE 4.2.8**
assigned groups. In the above example in figure, there is a row that can be reordered to give a better result. This would have been the user’s decision not to reorder so as to keep the groups at similar sizes.

Seriation

Seriation reorders only the rows of the matrix to create dense groups of ‘1’s in each column thus minimising the ‘0’s between them. An example of this can be seen in Figure 4.4.1. By reordering the columns to form what is known as a “battleship curve”, the matrix displays a stepped structure mainly centred on the diagonal as seen in Figure 4.2.10. Seriation is mainly used in archaeology in order to put excavation sites in chronological order. For example, when excavating graves, items such as pots have varying styles depending on the time. If the rows in figure 1 and 2 represent pots of varying styles and the columns represent graves that the pots were found in, then preforming seriation and reordering the columns would list the graves in a chronological order (forward or backward) where the pots would represent time. For this project, it is the structure of the matrix and its ability to fit the most amount of manages that have worked with an employee together that makes seriation work well. The statistical software package R allows the generation of the seriation algorithm.
4.3 Testing Phase

Each algorithm discussed in Section 4.2, p. 13 was written in R to be tested. Tests were conducted using the sample data provided. The accuracy metric discussed in Section 3.6, p. 9, determined how good an algorithm was. Different constraints were used such as different number of groups and different variances in group size. Once the accuracy was recorded for different variables, a good picture as to where the algorithm performs best and worst could be seen.

4.4 Scheduling Application Phase

An application in Microsoft Excel was created using Visual Basic to implement the chosen algorithm. The user manual for this application is attached to this report. The aim of the application is to give the user as much discretion as possible when organising the appraisal meetings. Human opinion is a variable that will always exist but cannot be programmed into any algorithm. It is important to be able to recommend a solution that can be altered to a user’s preferences rather than force one solution that has to be used. The Microsoft Excel application offers this to the user.

Functions:

- **Visual Aid** – Once the data has been imported from the user specified files, it is checked to ensure it is in the correct format and then sorted by departments and then
by levels. Figure 4.5.2 illustrates what the data looks like when all departments and levels are in view. The departments are highlighted in yellow, employees of similar level are highlighted in orange and the appraisers are highlighted in green. Any department and/or level can be selected using the dropdown lists on the home screen, seen in Figure 4.5.1. This will then display only the selected department and

![Home Screen](image)

**FIGURE 4.4.2**
level so the user can see clearly which groups every person is in.

- **Application of Algorithm** – By selecting any level in a department, the user can choose to edit the existing groups. By editing, the user is prompted to enter the desired number of groups and the algorithm is implemented to create the recommended row and column permutations that will result in the number of groups being formed.

- **User Flexibility** – If the user thinks that the recommended groups need small edits, it is possible to perform single swaps between groups. This is easily done by simply changing the number of the group an individual is in to the desired group number. A guide is in the margins of the matrix which the user can use. This guide displays information that can be used to see where the second best group to put a person is, or third best, etc.

![Edit Group](image)

**FIGURE 4.4.3**
The algorithm will have already put each person in the best possible group within the problems constraints.

- **Highlight Swaps** - If there is a junior that has less than 70% of the people he/she has worked with in the group that they are to be discussed in, then that column is marked in red in the margins. This highlights the seniors that may have to swap groups in order to have a high accuracy rate for all juniors. Swapping between groups is to be expected and helps improve the overall accuracy of the process.

- **Export Groups** - When the user is happy with the groups that have been created for a particular level in a department, they can then export the matrix to a separate worksheet so it remains unchanged when making edits to other groups.

- **Recommended Appraisers** - When files are imported after opening the programme, the appraisers list is checked to ensure that each Junior is assigned exactly one appraiser but it also check if the assigned appraiser is in the same department. If there are any that are not the user is alerted and they can update. It is not necessary to have the appraiser in the same department as the junior but it makes for better results as they are more likely to have worked with more of the same people.
Handling Constraints:

All of the algorithms that were considered were primarily focused on creating dense groups that have high accuracy. This does not take into consideration the constraints of the problem (Section 3.9, p. 11). For this reason, the Microsoft Excel application handles the constraints at each step of the implemented algorithm. When row or column permutations change, the application checks to ensure none of the primary constraints have been broken. If they have the application reorders the rows and columns to rectify the conflicts before the algorithm continues. The most important constraint is that the appraiser of each junior must be in the same group as the junior. Many appraisers have multiple juniors to appraise. Therefore it makes logical sense for all of these juniors to be side by side in the columns. This is the first step the application takes. After the appraisers have been groups as seen in Figure, they are kept in these groupings for the rest of the algorithm. The first four columns are grouped in Figure 4.4.7, therefore if the algorithm attempts to move column 1 it must move column 2, 3 and 4 with it. This also increases the efficiency of the algorithm because these groups of columns can be thought of as one collective column. In Figure 4.4.6 there are 22 separate columns to consider, but after they have been grouped in Figure 4.4.7, there are now only 11 columns.

Once the matrix is split into groups the application also works on the rows to ensure that appraisers are in the right groups. It does this by using the fact that the appraisers are reprogrammed as ‘9’s when the program first imports the data. A higher priority is given to the appraisers using the ‘9’s as a weighting when ordering the rows.
5. RESULTS AND FUTURE WORK

5.1 Algorithm Selection

The algorithm selected was the ‘Maximum Proportions’ algorithm for the following reasons:

The Maximum Proportions algorithm gave promising results in the testing phase. All algorithms were tested using R. The results varied depending on the group size variations and number of groups required. This led to some algorithms performing equally as well as, and sometimes better than, the Maximum Proportions algorithm. This was the case for the Seriation and the Barycentric algorithms in particular. However, even though the Maximum Proportions algorithm was out performed in many tests, it remained the most consistent throughout.

The algorithm also benefits from its ability to handle dynamic group sizes. The group sizes vary at every step of the algorithm which allows for a greater accuracy. The Maximum Proportions algorithm is the only algorithm which simultaneously determines the groups and attempts to find a permutation to fill each group. All other algorithms attempt to find a permutation of rows and columns followed by determining the groups. These algorithms use the same permutation of rows and columns, no matter how many groups are required. For this reason, the Maximum Proportions algorithm is able to handle more groups better than the other algorithms.

The Maximum proportions algorithm does not rely on a successful grouping algorithm to be applied. This is the case for the other algorithms. The grouping algorithm that is used works well when only splitting into a small number of groups that do not vary in size by more than 10 rows/columns. Any more than this and the algorithm takes an impractical length of time to run.

5.2 Process Evaluation

When the Maximum Proportions algorithm was finalised as the best choice for this project, it was implemented in the Microsoft Excel application and tested further. It was found that the accuracy of the new process improved by 55.8% in the number of swaps needed to reach a 70% average (Section 3.6, p. 9), 111.1% improvement in the overall average percentage in each group and 51.5% improvement in the minimum percentage in each group.

This evaluation used the analysts from three separate departments with a varying number of groups required. Analysts were used as they are the largest groups in each department and therefore are in greater need of splitting into smaller groups. The departments TI, FS and FT were used because they have the largest number of analysts. Each test applied the old process followed by the new process and calculated the accuracy of each. The output from these tests can be seen in Appendix G, p. G.1. The results can be seen in Figure 5.2.1.
When the application attempted to split the Analysts in the FT department into two groups, the result was one very large group and a second small group. It is at the user’s discretion as to what constitutes an impractical sized group. For the purpose of this evaluation it was decided that the algorithm was closer to one large group than it being split into two smaller groups. This suggests that the algorithm could not find acceptable groupings for the specified number of group. A different group size is therefore recommended.

The new process shows a combined average improvement of 72.8% compared with the old process. It allows more flexibility for the company in organising its appraisal meetings and can significantly improve on the efficiency of splitting large groups into smaller ones which both reduces the overall time the meetings take and the impractical number of people in certain groups.

By finding an algorithm that can efficiently split large groups into several smaller ones, the terms of reference were met. This both reduces the time it takes to perform the performance appraisal process and handles its scalability issues by having smaller groups. Most of the secondary objectives were also met by creating the Microsoft Excel application which is both flexible and extendable.

5.3 Further Work

The algorithm that is proposed in this report is not an absolute solution to the grouping problem. There have been good results to problems similar to this by using genetic algorithms however more research into implementing them for this problem is needed. If better solutions can be found using alternate algorithms, they can be easily incorporated into the Visual Basic used to manipulate the final Microsoft Excel solution. By doing this, the
Microsoft Excel application can benefit from better accuracy in the grouping of staff but it will also be able to offer its other functions to meet the various constraints that are required for a solution to the problem.

In the proposed solution, it is explained how to use Excel's Solver function if the results of splitting large groups into smaller groups do not meet adequate standards of accuracy. This is only effective if the groups are small enough but it could be incorporated into the final application in the future using Visual Basic.

The process proposed requires a survey to be circulated through the company in order to work out who has worked with whom. The process of surveying everyone in the company may be made simpler using Deloitte's timesheet tracker. The timesheet tracker is used to record all employees' work with clients. The analytics team has already used this data in the past to create a social network diagram using Gephi. By using this social network it would be possible to see who has worked on the same projects. Using this social network to classify who has worked with whom explicitly will not work because some projects can have in excess of 50 people working on them. Therefore not everyone will have come in direct contact. However, it does classify a large portion of the people that a person has not worked with. By surveying each person to just include the people that the Gephi social network thinks a person has worked with will reduce the size of each survey substantially. Future work into whether this is a possible concept is needed.

Controls (Section 3.1, p. 5) exist in the overall appraisal meetings to ensure a consistent standard for all staff but not in the initial one-on-one meeting with the appraiser. Introducing controls for this initial one-on-one meeting could help improve the equality of all appraisal meetings by countering any biased opinions the appraiser may have, good or bad, about the employee being appraised.

It is possible to reduce the length of time spent discussing each employee at the appraisal meeting. This can be done effectively by implementing a voting system. Currently 2 minutes is dedicated to the presentation by the appraiser and 3 minutes to discussion on the rating given. By allowing a vote by all staff members that have worked with the employee instead of open discussion, this three minute window could be reduced to less than one minute. If an employee has been given a provisional rating of 3, the staff in the group that has worked with the individual could simply vote what rating they would have given the employee. The average of the votes is then the final rating. This not only could reduce the planned 3 minutes but could also avoid disputes between managers as to what rating a person should get.

Due to insufficient time, adjustments need to be made to the Microsoft Excel application. These adjustments are primarily concerned with the constraints on group sizes. In some cases, the program will over sort the data and groups will be made larger than the user may want.
A. ORIGINAL PROJECT REPORT

Client: Deloitte
Project: Performance Appraisal Scheduling
Location: Hatch Street Upper, Dublin 2
Client Contact: Richard Southern, 01 417 3831, risouthern@deloitte.ie
Ciarán Tobin, 01 417 2408, ctobin@deloitte.ie
David Horn, dahorn@deloitte.ie

Client Background

Deloitte is the brand under which tens of thousands of dedicated professionals in independent firms throughout the world collaborate to provide audit, consulting, financial advisory, risk management, and tax services to selected clients. Deloitte Analytics in Ireland deliver solutions to data, technology, and business challenges. Our comprehensive approach to analytics is fuelled by our deep industry knowledge, broad functional experience, and mastery of technology.

Client Requirement

Twice a year, Deloitte undertakes a performance appraisal process for all staff. This process is intended to both evaluate performance and give employees the opportunity to discuss their personal and career objectives with their managers, directors, and Deloitte partners. It is important that where possible all senior staff members who have worked with a particular individual are present at that individual's appraisal meeting to fairly evaluate their performance and discuss their personal objectives. Over the course of a year, each individual staff member is likely to have worked on multiple projects with a large number of other people. In previous years it was possible to simply split the senior staff group in half based on department (Technology, Strategy, Financial etc) and review their respective junior staff members. However due to the continuous large growth of the consulting department, the entire process is becoming increasingly difficult to manage in terms of time. It would be more efficient if the groups could be split several smaller groups whilst still allowing for the correct people in be in attendance at each employees review. This has the potential to save a large amount of time bearing in mind there are approximately 250 employees in Deloitte consulting.

In practice, this is a scheduling problem similar to exam or class timetabling. There are a number of entities (staff members) that must be present at a number of locations (appraisal meetings) within a certain time constraint. However there are other constraints such as attempting to minimise the number of times staff members have to switch groups. In effect we don't want everyone switching tables every 5 minutes. It may also be possible to approach this as a metaheuristic problem where we run an unsupervised clustering algorithm in order to create these smaller groups.
What is involved for the student?

- Research possible techniques and solutions to similar problems.
- Recommend or design a solution. This may involve the suggestion to acquire some existing software or require the student to build a simple prototype/solution (depending on the complexity of the solution required).
- The student will be provided with sample data that will be required to solve the problem.
B. INTERIM REPORT TERM 1

Project: Performance Appraisal Scheduling
Client: Deloitte
Student: Ciaran Doyle (11440278)
Supervisor: Frank Bannister
Date: 3-12-2014

Review of Background and Work to Date

Twice a year, Deloitte undertakes a performance appraisal process for all staff. This process is intended to both evaluate performance and give employees the opportunity to discuss their personal and career objectives with their managers, directors, and Deloitte partners. Currently, an employee meets with his/her assigned appraiser (generally their supervisor) and after discussing the employee’s performance and objectives, they are given a provisional rating out of 5. In the appraisal meeting, which holds all members of staff above the level of the employee, the appraiser represents the employee by presenting to the congregation the rating and reasons why they have given the employee this rating. Anyone in the room that has worked with the employee will give input as to whether they agree or disagree with the rating until it is finalised. This process is done one level at a time so when all the analysts have been appraised, the consultants will be asked to leave the room and they will be appraised by all levels above them. This process continues to partner level. Therefore the employee being appraised will never be present in the room when they are being discussed.

So far, two meetings have been held with the clients and three with the supervisor of this project in order to establish clear client requirements and discuss possible solutions to these requirements.

Terms of Reference

The problem with the current process is scalability. Now that Deloitte has grown and plans to double in size in the coming years, it is becoming increasingly hard to manage the appraisal meetings. The aim of this project is to:

- Improve/alter the current process in order to handle the larger numbers more efficiently.
- Propose a generic solution that can be applied twice-yearly to all departments.
- Ensure that the solution is scalable.

Further Work

- Identify any techniques and existing software packages that can handle the current problem and access how well they can handle it.
- Establish a working fix to the problem.
- Investigate the possibility of solving the problem by approaching it as purely a scheduling problem.

Conclusions

Deloitte wish to ensure the appraisal of their employees is fair by having everyone present that can offer input into an employee’s performance as well as having a certain number of people not affiliated with the employee present to ensure consistency. They are open to a completely new process, however, would prefer if there was a way of simply improving the current one. This project aims to deliver a working solution that is as efficient as possible.
C. INTERMIM REPORT TERM 2

Project: Performance Appraisal Scheduling
Client: Deloitte
Student: Ciaran Doyle (11440278)
Supervisor: Frank Bannister
Date: 3-2-2015

0. Background

The purpose of this report is to summarise; work to date, how this work compares to the work laid out in the interim report, future work in order to comply with the terms of reference in the interim report and possible additional work. As per the interim report, this project aims to handle the scalability problem of Deloitte’s current appraisal process. The current process is not scalable for two fundamental reasons:

1. Time Scale – The increasing number of staff members is leading to the appraisal process spanning 2 to 3 days. With the current process, this time scale is only going to increase further.
2. Meeting Size – With more staff members, more people are eligible to give feedback in appraisal meetings and therefore the meetings are becoming increasingly harder to manage.

Both of these problems can be addressed with one solution; split the managers\(^1\) into several groups with each assigned a group of employees\(^2\) to appraise.

1. Work to Date

1.1 Overview

A mock data set has been given in the form of a binary matrix where the digit ‘1’ represents a work relationship between a manager and an employee and the digit ‘0’ fills the remaining cells in the matrix. Each employee is assigned a manager that represents them in the appraisal meeting as the person being appraised is never present in a meeting they are being discussed in. These managers must be present for their corresponding employee’s appraisal meeting. A mock list of employees and corresponding managers was also given.

1.2 Algorithms

There are two primary algorithms needed to complete this project:

1. Grouping
2. Ordering

\(^1\) “Managers” in this report refers to all staff members above the group being appraised.
\(^2\) “Employees” in this report refers to the staff members that are being appraised.
Grouping is an algorithm that, given an input number of groups, will establish the best coordinates on the matrix to split so as to maximise the number in these groups. ‘Ordering’ is an algorithm that will reorder the rows and columns of the matrix so as to optimise the ‘grouping’ algorithm. Algorithms to implement ‘grouping’ and ‘ordering’ together have been investigated but have not materialised, often due the number of permutations needed exceeding computational power. Ordering algorithms will be discussed first for reasons that will be explained when discussing Grouping Algorithms.

1.2.1 Ordering Algorithms

In an attempt to stay brief, the names of algorithms that have been tried are given. Full explanations and issues associated with them will be given in the final report as it is necessary to explain what may seem like ‘obvious’ solutions to this project that did not work. All algorithms are original ideas or were original ideas that were later discovered in the literature:

1. Diagonal Dominance (Grouping Along the Diagonal)
2. Barycentric Algorithm (Average position of ‘1’s)
   1. Grouping Rows All Permutations
   2. Grouping Rows Best Overlap
   3. Best Overlap with merit/loss to form block diagonal matrix
   4. Best Overlap with merit/loss incorporating both Ordering and Grouping algorithms
   5. Seriation

These algorithms were the most significant of many that were considered. Algorithm 7, Seriation, is the most promising of all algorithms tested. Seriation stemmed from the previous four algorithms (3 to 6) in that it reorders only the rows of the matrix to create dense groups of ‘1’s in each column thus minimising the ‘0’s between them. An example of this can be seen in Figure 1. By then reordering the columns to form what is known as a “battleship curve”, the matrix displays a stepped structure mainly centred on the diagonal as seen in Figure 2. Seriation is mainly used in archaeology in order to put excavation sites in chronological order. For example, when excavating graves, items such as pots have varying styles depending on the time. If the rows in figure 1 and 2 represent pots of varying styles and the columns represent graves that the pots were found in, then preforming seriation and reordering the columns...
would give the graves in a chronological order (forward or backward) where the pots would represent time. For this project, it is the structure of the matrix and its ability to fit the most amount of managers that have worked with an employee together that makes seriation work well. R supplies a package called seriation that will be investigated further to see if it can perform what is required for this project.

1.2.2 Grouping Algorithms

A grouping algorithm that permutes all possible splits within constraints (groups need to be relatively the same size but slight variation is allowed) should work well with the seriation algorithm. Due to seriation not clustering rows and columns simultaneously to form groups, grouping must take place after row and column ordering and for this reason it made more sense to discuss ordering prior to grouping. The final program should allow for inputs such as number of groups and group size variation. These inputs will hopefully be available to make from a user friendly interface using R Shiny. This will aid Deloitte’s HR department when using the program and not have to search through unfamiliar R code. However, further investigation into R Shiny is needed.

2. Handling Conditions

2.1 Appraisers

As explained in section 1.1, managers that are assigned to employees must be present in that employee’s appraisal meeting. To handle this, a macro in excel was written to recode these appraisers as a higher number than 1. By doing this, seriation places more priority on the cell with the higher value and thus the ‘1’s will be grouped around the higher value cells. The exact value of this cell is to be investigated further as too low a number will not ensure all appraisers are present in the meeting but too high and the seriation algorithm may find the ‘1’s in the matrix too insignificant and thus not result in good grouping. Similar code to the macro needs to be written in R so as to keep all code to the one platform.

2.2 Accuracy

Accuracy is measured off two conditions; 1) the average percentage of managers that are present in the employees’ appraisal meetings out of the total managers they have worked with and 2) the employee with the minimum percentage of managers present. Another input will be minimum requirements for these two conditions. How to handle columns that do not fulfil these requirements needs further investigation.

1. Further Work and Conclusions

In conclusion, further work is needed for the following; the seriation package in R to improve accuracy, R shiny for use of user-friendly inputs, potential inputs to include, minimum conditions and handling minimum conditions, grouping algorithms, what value to give appraisers that works best with the seriation algorithm, and error handling. How to organise the process itself also needs work. This will include how the day is run, what groups will be
separated and how many different group changes will be made throughout the day (possibly one for each tier of employment) and estimating time frames for each meeting and the appraisal process as a whole.

The algorithms that have been discussed address all terms of reference contained in the first interim report which were; to improve/alter the current process in order to handle the larger numbers more efficiently, propose a generic solution that can be applied twice-yearly to all departments and to ensure that the solution is scalable.
D. PROJECT CLARIFICATIONS 1

Project: Performance Appraisal Scheduling
Client: Deloitte
Student: Ciaran Doyle (11440278)
Supervisor: Frank Bannister

Date: 13-11-2014

Clarifications Required:

- Clarification of what systems are in place that retains complementary employee information to be used for the project. i.e.
  - Payroll
  - HR system - Does this retain employee appraisal info
  - Project Management - Does this retain list of employees that worked on each project.

And are these of concern to the project or will the data be simply given.

Answer: I would have to clarify exactly how the end results are recorded but as far as I am aware it is a manual/Excel process. Although currently that will most likely be outside of the scope of the project. Before the consistency meeting each employee sends in a document relating to their yearly goals/performance and a small summary slide. This is collected by HR and this forms the list of employees to be reviewed. So we will have approx. 50 slides for just the analysts.

- Require an understanding of how your current method of scheduling works by dividing the senior staff group in half by department.

Answer: At the moment there is no real process in place of dividing the team. What happens usually is that the appraisers (managers etc.) are split in some sort of general sense. For example we might put all of the appraisers that work in AIB, Citi, and Kerry Group in one group and all the appraisers that work together on the Department of Social Protection, BOI, and some other client on the other. That way you end up with managers that can talk about people that they work with. Going forward what I think we could do is send out a survey beforehand that everyone could fill out simply stating their name and who they would like to talk about during the process. Then you would end up with a matrix similar to below.

<table>
<thead>
<tr>
<th></th>
<th>Mr A1</th>
<th>Miss A2</th>
<th>Mr A3</th>
<th>Miss A4</th>
<th>Mr A5</th>
<th>Miss A6</th>
</tr>
</thead>
<tbody>
<tr>
<td>David Horn</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Ciaran Tobin</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Neil Brett</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Aideen Keaney</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
So in this Ciaran and Aideen want to talk about Analyst 1 and Neil and I want to give input on Analyst 2. This would then form the basis of our problem to solve. In this very basic case, you would probably end up splitting me and Neil into one group and Ciaran and Aideen into another. Although it’s not a perfect solution it is probably the best one. You can see we might have to get Aideen to come over to our group when we review Analyst 6.

- Require an understanding of the limitations of your current process.

**Answer:** The main issue at the moment is that there isn’t really a process in place. We want to put some sort of formal process in place. It is simply done on a “what seems like a good way” to split the groups.

- A clear definition of what information will be retained in the proposed system.

**Unanswered**

- Review any possible interfaces or data feeds that may be of benefit to or from existing systems to avoid duplication and enhance functionality.

**Answer:** There is a timesheet tracker that is used to record all employees work with clients. The analytics team have actually used the data in the past to create a social network diagram using Gephi in the past (just as a proof of concept). The only thing about using this data is that although you may have worked in the same client you may not want/be able to give feedback on a person. For example I work in the Revenue Commissioners but so do about 40 other Deloitte people on different teams (Dev, Test, Live Support, BI). So I haven’t worked with everyone so can’t give feedback on them.

- In addition to key functionality requirements in the proposed new system, I would also like to get an understanding of any aspirational features you would like to see in the new system.

**Answer:** Limited movement of people during the meetings would be one thing. If we were being really ambitious it would be great to have the system administrating over the whole thing on the day as well. As in you would upload all the consistency slides and it would play the whole thing – show a timer on the screen and automatically flick to the next person (with a break for moving around). But we may have to see what we can encompass in the scope. At the heart of it, we want a system to schedule the meetings/organise the groups. I could see the whole thing being hard to run even with a perfect plan worked out. People are inevitably going to run over time.
E.  PROJECT CLARIFICATIONS 2

Project: Performance Appraisal Scheduling
Client: Deloitte
Student: Ciaran Doyle (11440278)
Supervisor: Frank Bannister

Date: 24-11-2014
Clarifications Required:

- Require an understanding of how appraisers are picked? Is it always that managers are assigned to analysts/consultants, senior managers are assigned to senior consultants, directors are assigned to managers, etc.?

  Answer: Usually an appraiser is recommended to be at least two levels above their appraisee, and should be at least a manager. So the grades are in order: Analyst 1, Analyst 2, Consultant, Senior Consultant, Manager, Senior Manager, Director, and Partner. Analysts/Consultants will usually have managers. Senior Consultants probably have Senior Managers. Managers might have directors etc. Don’t think it’s too strict though.

- What is the minimum number of people required to be in the room to appraise an employee? (ie. can I consider the possibility of one on one) And of this number what is the proportion or minimum number of people directly working with the employee needed?

  Answer: Well at the moment all of a department is in the same room but divided in two. So for Technology there are about 70 in a room at one time. Remember the employee is not in the room at this time, so it can’t be a one on one. This is a meeting where all management decide on the rating of a person. You as an employee meet with your appraiser beforehand to discuss your yearly work and are given a provisional/suggested rating but they are the person that mainly represents you in this meeting. The aim of the meeting is to ensure consistency across all the different projects.

- What exactly is the time frame I have to work with? I understand at present it is 1 to 2 days of appraisals. Can this be stretched? (For example a person is possibly assigned one or two hours for Monday and Wednesday of a week to take part in an appraisal related commitment)

  Answer: This should be an input into your algorithm. You might not know up front what the time constraint is, and you should be able to account for whatever we want the time limit to be.
• Are there any specific guidelines set by Deloitte when it comes to performance appraisals? (For example I assume all Deloitte firms are required to do the appraisals twice a year) Is every Deloitte expected to do their appraisals the same way, if not do they all do it the same way?

**Answer:** Not sure. I'm fairly sure the process is fairly standard but can ask HR. We do meetings twice a year but the specifics around them aren’t set in stone.

• Are you aware of any regulatory bodies or employee rights that the current system is adhering to?

**Answer:** Not a clue! This probably goes beyond the scope of your project, so you shouldn't worry about it.

• Originally, what were the objectives of this project or was it just intended to improve on what is already there by essentially solving a mechanical problem? Or has the idea of coming up with an entirely new concept been in the pipeline from the start?

**Answer:** The objectives are really just to improve what's already going on. I’d imagine the process will be exactly the same in the end, just the time organisation will be much better.
F. LOG BOOK

22 Sep 14 – Project given. Client stated that they ideally want a generic solution that could be used every year. Suggested OptaPlanner, custom R solution or Excel. Also do research on the literature out there on similar projects

28 Sep 14 – Accepted project. Emailed on clarifications document. (Appendix D, p. 1)

30 Sep 14 – Reply from David with answers to clarifications document. Suggested time to meet. (Appendix D, p. 1)

1 Oct 14 – Agreed to meet at 2:30 on the 2 Oct.

2 Oct 14 – Meeting 1 - Met Clients to discuss project. Notes taken by me after meeting:

“Everybody from the department except the analysts meet in a room. Each analyst submits a slide about what they have done and their goals. An appraiser (who must be present for his/her analyst review) gets up and talks about his assigned analyst. Everyone else expresses their opinions and so on. Then all the consultants leave the room and the higher tiers do the same for the consultants.

Now they might split into two groups so as they can do it quicker. People might move over to the other group during the day so as they can sit in on someone they wish to review.”

7 Oct 14 – Sample data given. Email explained the data. There were two files attached. File 1 was a large matrix of everyone working in Deloitte and had randomly assigned the matches by giving everyone between 4 to 9 people above their grade as well as slightly weighting this so that people in the same department were more likely to work with each other. No partners included as he did not have access to a list of them. In the end we may limit this to just TI. File 2 contained the list of who is appraising who.

13 Oct 14- Supervisor Meeting 1- Briefed Supervisor on the project. Agreed to get exact specifications as to what I can and cannot do. How appraisers are picked? Always one tier above? Minimum number of people needed to appraise a person? Of the people, what minimum number that are affiliated with the individual have to be there? Can it be one on one? What is the timeframe needed? What are the objectives of this project or is it literally a mechanical problem? Overall, is there any particular guidelines you have to follow or regulations within Deloitte that are required to fulfil aside from holding an appraisals twice a year? Is there any regulatory body or employee rights to consider when it comes to employee appraisals? The idea is that we come up with some sort of system that gives every person being appraised the exact same level of appraisal, so we don't end up having one person being appraised with 8/9 of his supervisors present and another person only having 1/9 of his supervisors present. Are there perhaps any criteria we can come up with to judge the performance of people, so being a bit more precise about how they are doing like rating career ambition out of 10, consistency out of 10, work ethic out of 10. Approaching it as if they have interviewed for a promotion and picking the top 10 percent of people to be promoted. At the end people will then have a rank (only for the use of whoever it is that is
deciding on promotions) Using Bayesian inference perhaps then we can use each appraisals data each year when considering the rank of an employee. That way we can also take into consideration how long the employee has been in the company by the number of times he has been appraised. So the more times you have been appraised the better the rating you should have.

Focus group with analysts from the company to see how they feel it should be done.?

15 Nov 14 – I sent on a new clarifications sheet to Clients. Expressed enthusiasm about coming up with innovative new ideas of how to go about this project. I suggested holding a focus group for the new ideas.

18 Nov 14 – Got a response from Client to the new clarifications. They suggested that the scope of the project is sufficient and therefore doing something new is unnecessary. Therefore a focus group would be unnecessary and infeasible.

21 Nov 14 – Supervisor Meeting 2 – Talked to Supervisor about the clarifications that I had sent on to the client and their response, how they felt there was enough in the original project without broadening the scope. Supervisor still expressed concerns about the scope and wanted to meet with the clients.

23 Nov 14 – Organised meeting with the clients for the next day (24 Nov).

24 Nov 14 – Had meeting with supervisor and client. It was decided that the problem has sufficient work in it without having to come up with new ways of doing the process.

Main concern was that the project would be too easy and be done before Christmas but not so complicated that I would not be able to reach a staging on it. The problem itself is hard enough without getting too side tracked according to the client. They are looking for a black box approach in which we have a number of inputs and spits out a schedule. The problem: Each person meets with their assigned appraiser and they give you a rating out of 5. The appraiser then puts this rating to the rest of the senior staff and then anyone who has worked with that person can give input as to whether they agree with the rating or

Looking to double the size of the company. 100 people in TI which is the biggest department. They would like this problem to solve just for the TI department but ideally if they could use it for all of consulting that would be the main objective. This is because someone in TI could have worked with Strategy but the people in strategy would not get to give input into your appraisal.

Clustering solution or a scheduling solution such as scheduling exams. The problem with the latter is that you do not want people jumping from one spot to another. As well as having a timing issue. It would have to be strict timing. I was thinking that if every group has a button they click when they are finished but they do not move until every group has clicked the button.

For optimising we would need a gini index.
Clusters along the diagonal of the matrix.


2 Dec 14 – Investigated the Diagonal matrix solution with a small matrix (3*10). This looked promising however would leave a large misclassification rate.

3 Dec 14 – Interim Presentation - Presentation with supervisor. It was suggested to look into Haslem appraisal process/360° Feed back

18 Dec 14 – Further investigation into diagonal matrix, or “block matrix”. Need one 0 (person not affiliated with the employee) and one 2 (appraiser). A supervisor may have more than just one employee to appraise and therefore more than just one meeting they are committed to attend. These minimum requirements must be met first in the solution and then add in the rest.

To Do:
- Program in appraisers into matrix (with original data)
- Create new matrix of considerable smaller size
- Establish an algorithm that will rearrange rows and columns to best fit the diagonal of the matrix for each set of appraisees. (on smaller matrix)
- Program a function to split the matrix into a specified number of groups (with equal numbers in each group) (on smaller matrix)
- Program a function to split the matrix into a specified number of groups (with differing numbers in each group if it improves the solution) (on smaller matrix)
- Test solution’s scalability on larger matrix.
- Test solution on original matrix.

Questions:
- How are the appraisers picked? Is that something I need to assign or will the appraiser just have to fill out who their appraiser is at the time the survey is handed out.
- How many Employees can on manager appraise.
- In the data given, quite a lot of the appraisers are not in the same department as the employee they are appraising. Is this deliberate as the appraiser is supposed to be someone that is not to have worked with the individual before and thus only knows them from regular meeting or is this a mistake and should the appraiser generally be the employee’s supervisor and therefore in the same department.

What was done: Analysing the original data further I created a table that relays the number of people in each department as we as the number of employeeed in each employee rank. This made me realise that there would be more than on appraisee per appraiser potentially. My small matrix was decided on; The TI department with just Analysts and their appraisers (managers), a 39 by 17 matrix.
Thoughts: Just because a person works in a particular department does not necessarily mean that most of their colleagues are in that department. I.e. they could work for analytics but on a project with people from other departments than analytics. Therefore that person should be grouped with the department he/she has done the most work for. This can be quantified by adding the total number of people that have worked with a particular person.

Block diagonal matrix. This only works with a square matrix but by adding empty rows evenly across the matrix to make it square, fitting a block diagonal matrix with a specified number of blocks would work.
G. RESULTS

G.1 Splitting Analysts in TI department into 2 Groups

Old Process
### Table: New Process

<table>
<thead>
<tr>
<th>Position</th>
<th>Consultant</th>
<th>Manager</th>
<th>Senior Manager</th>
<th>Senior Consultant</th>
<th>Consultant</th>
<th>Manager</th>
<th>Senior Manager</th>
<th>Senior Consultant</th>
<th>Consultant</th>
<th>Manager</th>
<th>Senior Manager</th>
<th>Senior Consultant</th>
</tr>
</thead>
<tbody>
<tr>
<td>AVERAGE: 0.83</td>
<td>MINIMUM: 0.4</td>
<td>SWAPS: 6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exit Edit</td>
<td>Sort</td>
<td>Export</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
G.2 Splitting Analysts in TI department into 3 Groups

Old Process
G.3  Splitting Analysts in TI department into 4 Groups

Old Process
G.4  Splitting Analysts in TI department into 5 Groups

Old Process
New Process

Exit Edit  Sort  Export

AVERAGE: 0.523

MINIMUM: 0.167

SWAPS: 30
G.5 Splitting Analysts in FS department into 2 Groups

Old Process

New Process
G.6 Splitting Analysts in FS department into 3 Groups

**Old Process**

<table>
<thead>
<tr>
<th>Group</th>
<th>Department</th>
<th>Position</th>
<th>Analyst/Analyst/Analyst/Analyst/Analyst/Analyst/Analyst/Analyst/Analyst/Analyst/Analyst</th>
<th>Analyst/Analyst/Analyst/Analyst/Analyst/Analyst/Analyst/Analyst/Analyst/Analyst/Analyst</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 FS</td>
<td>Consultant</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1.2 FS</td>
<td>Consultant</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1.3 FS</td>
<td>Analyst/Analyst</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1.4 FS</td>
<td>Analyst/Analyst</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1.5 FS</td>
<td>Analyst/Analyst</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1.6 FS</td>
<td>Analyst/Analyst</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1.7 FS</td>
<td>Analyst/Analyst</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1.8 FS</td>
<td>Analyst/Analyst</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1.9 FS</td>
<td>Analyst/Analyst</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1.10 FS</td>
<td>Analyst/Analyst</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1.11 FS</td>
<td>Analyst/Analyst</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1.12 FS</td>
<td>Analyst/Analyst</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1.13 FS</td>
<td>Analyst/Analyst</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

**New Process**

<table>
<thead>
<tr>
<th>Group</th>
<th>Department</th>
<th>Position</th>
<th>Analyst/Analyst/Analyst/Analyst/Analyst/Analyst/Analyst/Analyst/Analyst/Analyst/Analyst</th>
<th>Analyst/Analyst/Analyst/Analyst/Analyst/Analyst/Analyst/Analyst/Analyst/Analyst/Analyst</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1 FS</td>
<td>Consultant</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1.2 FS</td>
<td>Consultant</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1.3 FS</td>
<td>Analyst/Analyst</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1.4 FS</td>
<td>Analyst/Analyst</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1.5 FS</td>
<td>Analyst/Analyst</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1.6 FS</td>
<td>Analyst/Analyst</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1.7 FS</td>
<td>Analyst/Analyst</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1.8 FS</td>
<td>Analyst/Analyst</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1.9 FS</td>
<td>Analyst/Analyst</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1.10 FS</td>
<td>Analyst/Analyst</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1.11 FS</td>
<td>Analyst/Analyst</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1.12 FS</td>
<td>Analyst/Analyst</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1.13 FS</td>
<td>Analyst/Analyst</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

AVERAGE: 0.742

MINIMUM: 0.5

SWAPS: 5

**G.6.5 Exit Edit**

**G.6.6 Sort**

**G.6.7 Export**

---

**Page G.10**
G.7 Splitting Analysts in FT department into 2 Groups

### Old Process

#### 2 Groups

<table>
<thead>
<tr>
<th>Group</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Position

|----------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|

#### AVERAGE: 0.435

#### MINIMUM: 0

#### SWAPS: 13

### New Process

#### 2 Groups

<table>
<thead>
<tr>
<th>Group</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
<th>FT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Position

|----------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|

#### AVERAGE: 0.837

#### MINIMUM: 0.2

#### SWAPS: 2

### Exit Edit | Sort | Export
G.8  Splitting Analysts in FT department into 3 Groups

Old Process

New Process
Sub Reset()
On Error GoTo errHandler
Sheets("Information").Activate
Application.ScreenUpdating = False

Dim ws As Worksheet
Application.DisplayAlerts = False
For Each ws In Worksheets
If ws.Name <> "Information" Then ws.Delete
Next
Application.DisplayAlerts = True
ActiveSheet.Shapes("departmentDropDown").ControlFormat.Value = 1
With ActiveSheet.Shapes("departmentDropDown").ControlFormat
    .Range("viewDepartment").Value = .List(.Value)
End With
ActiveSheet.Shapes("levelDropDown").ControlFormat.Value = 1
With ActiveSheet.Shapes("levelDropDown").ControlFormat
    .Range("viewLevel").Value = .List(.Value)
End With

With Sheets("Information")
    Dim nshapes As Integer
    For nshapes = 1 To .Shapes.count
        .Shapes(nshapes).Visible = False
    Next nshapes
    .Shapes("opnFiles").Visible = True
    .Shapes("Picture 1").Visible = True
    ' Application.Goto Reference:="Table1"
    ' If Selection.ListObject.ListColumns.count > 2 Then
        ' Dim i As Integer
        ' For i = 3 To Selection.ListObject.ListColumns.count
        ' Selection.ListObject.ListColumns(i).Delete
        ' Next i
    ' End If
    ' Application.Goto Reference:="Table2"
    ' If Selection.ListObject.ListColumns.count > 2 Then
        ' For i = 3 To Selection.ListObject.ListColumns.count
        ' Selection.ListObject.ListColumns(i).Delete
        ' Next i
    ' End If
    ' Columns("B:F").EntireColumn.AutoFit
End With

Range("Table1[Count]").Select
    Selection.ClearContents
Range("Table2[Count]").Select
    Selection.ClearContents

Cells(1, 1).Select
Application.ScreenUpdating = True
Exit Sub
errHandler:
Application.ScreenUpdating = True
MsgBox ("A fatal error has occurred. Please restart the program. If problem persists contact supervisor")
Exit Sub
End Sub

Sub OpenFiles()
'
' OpenFiles Macro
'
',

Dim ws As Worksheet
Dim folder As FileDialog
Dim file As String
ThisWorkbook.Worksheets("Information").Activate
Dim workbookName As String
workbookName = ActiveWorkbook.Name
workbookName = Mid(workbookName, 1, InStr(workbookName, ".") - 1)

Call Reset
Application.ScreenUpdating = False

'Use file browser to select folder hat files are in
Set folder = Application.FileDialog(msoFileDialogFolderPicker)
With folder
 .Title = "Select a Folder"
 .AllowMultiSelect = False
'If folder selection is cancelled then exit sub
If .Show <> -1 Then Exit Sub
file = .SelectedItems(1)
End With
'Directory is changed to selected file location
ChDir file

'Delete all other worksheets except "Information" worksheet
Application.DisplayAlerts = False
For Each ws In Worksheets
 If ws.Name <> "Information" Then ws.Delete
 Next
Application.DisplayAlerts = True

'open matrix and add to current workbook
On Error GoTo errHandler1
Workbooks.Open Filename:="Sample Link Matrix.csv"
Windows("Sample Link Matrix.csv").Activate
Sheets("Sample Link Matrix").Move After:=Workbooks(workbookName).Sheets(1)
ActiveWindow.Zoom = 10

'open staff list and add to current workbook
On Error GoTo errHandler2
Workbooks.Open Filename:="Sample Staff List.csv"
Windows("Sample Staff List.csv").Activate
Sheets("Sample Staff List").Move After:=Workbooks(workbookName).Sheets(2)
'Save new workbook as...
'   ActiveWorkbook.SaveAs Filename:="C:\Users\Pauline\Desktop\FYPSolution.xlsm" _
   , FileFormat:=xlOpenXMLWorkbookMacroEnabled, CreateBackup:=False

On Error GoTo errHandler3
Dim datas As Worksheet, staffws As Worksheet
Dim datastartLength As Integer, datastartWidth As Integer, dataLength As Integer
Dim dataWidth As Integer, i As Integer, staffLength As Integer
Dim department As String
Dim Row As String, Col As String

Set datas = Sheets("Sample Link Matrix")
Set staffws = Sheets("Sample Staff List")

'record the width and length of the matrix
With datas
  dataLength = .Cells(.Rows.count, "A").End(xlUp).Row - 1
  dataWidth = .Cells(1, Columns.count).End(xlToLeft).Column - 1
End With

'insert the width and length of the matrix into information sheet
With Sheets("Information")
  .Range("dataLength").Value = dataLength
  .Range("dataWidth").Value = dataWidth
End With

Call boundries
  With Sheets("Information")
    datastartLength = .Range("datastartLength").Value
    datastartWidth = .Range("datastartWidth").Value
    datafinishLength = .Range("datafinishLength").Value
    datafinishWidth = .Range("datafinishWidth").Value
  End With

'Check if matrix is in correct format
datas.Activate
Dim cl As Range
For Each cl In Range(Cells(2, 2), Cells(dataLength + 1, dataWidth + 1))
  If cl.Value <> 1 And cl.Value <> 0 And cl.Value <> 9 Then GoTo errHandler3
  If cl.Value = 0 Then cl.Value = "-
Next cl
Call updateAppraisers

dataws.Activate
'Insert 3 new columns
With Columns("B:B")
  .Insert Shift:=xlToRight
  .Insert Shift:=xlToRight
  .Insert Shift:=xlToRight
End With

'Insert 3 new rows
With Rows("2:2")
  .Insert Shift:=xlDown
  .Insert Shift:=xlDown
  .Insert Shift:=xlDown
End With

'In the third new column put employee ID
With Range("D5")
  .Formula = "=SUBSTITUTE(MID(A5,FIND("\\",A5)-3,3),"\\ ","")"
  .AutoFill Destination:=Range(Range("D5"), .Offset(dataLength - 1, 0)), Type:=xlFillDefault
End With

'In the second new column put position
With Range("C5")
  .Formula = "=LEFT(A5,FIND(D5,A5)-2)"
  .AutoFill Destination:=Range(Range("C5"), .Offset(dataLength - 1, 0)), Type:=xlFillDefault
End With

'In the first new column put department
With Range("B5")
  .Formula = "=MID(A5,FIND("\\",A5)+1,LEN(A5)-FIND("\\ ",A5)-2)"
  .AutoFill Destination:=Range(Range("B5"), .Offset(dataLength - 1, 0)), Type:=xlFillDefault
End With

'In third new row put employee ID
With Range("E4")
  .Formula = "=SUBSTITUTE(MID(E1,FIND("\\",E1)-3,3),"\\ ","")"
  .AutoFill Destination:=Range(Range("E4"), .Offset(0, dataWidth - 1)), Type:=xlFillDefault
End With

'In second new row put position
With Range("E3")
  .Formula = "=LEFT(E1,FIND(E4,E1)-2)"
  .AutoFill Destination:=Range(Range("E3"), .Offset(0, dataWidth - 1)), Type:=xlFillDefault
End With

'In first new row put department
With Range("E2")
  .Formula = "=MID(E1,FIND("\\",E1)+1,LEN(E1)-FIND("\\ ",E1)-2)"
  .AutoFill Destination:=Range(Range("E2"), .Offset(0, dataWidth - 1)), Type:=xlFillDefault
End With

'Turn formulas into values
With Columns("A:D")
  .Copy
  .PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
    :=False, Transpose:=False
End With
With Rows("1:4")
  .Copy
End With

'aesthetics
Columns("B:B").ColumnWidth = 11.71
Columns("C:C").ColumnWidth = 17.43
Columns("D:D").ColumnWidth = 4
Range("B2").FormulaR1C1 = "Department"
Range("C3").FormulaR1C1 = "Position"
Range("D4").FormulaR1C1 = "ID"

'Formatting: Highlight all cells greater that 0 in black and set column width
With Range(Cells(5, 5), Cells(dataLength + 4, dataWidth + 4))
  .FormatConditions.Add Type:=xlCellValue, Operator:=xlEqual, _
    Formula1:="=1"
  .FormatConditions(.FormatConditions.count).SetFirstPriority
  With .FormatConditions(1)
    .Font.ThemeColor = xlThemeColorLight1
    .Font.TintAndShade = 0
    With .Interior
      .PatternColorIndex = xlAutomatic
      .ThemeColor = xlThemeColorLight1
      .TintAndShade = 0
    End With
  End With
End With
End With
End With

With Range(Cells(5, 5), Cells(dataLength + 4, dataWidth + 4))
  .FormatConditions.Add Type:=xlCellValue, Operator:=xlEqual, _
    Formula1:="=9"
  .FormatConditions(.FormatConditions.count).SetFirstPriority
  With .FormatConditions(1)
    .Font.ThemeColor = xlThemeColorLight1
    .Font.TintAndShade = 0
    .Font.Color = 5287936
    With .Interior
      .PatternColorIndex = xlAutomatic
      .ThemeColor = xlThemeColorLight1
      .TintAndShade = 0
      .Color = 5287936
    End With
  End With
End With
End With
End With

Range(Cells(5, 5), Cells(5, dataWidth + 4)).ColumnWidth = 5
'clear column A and row 1 to be used for groups
Columns("A:A").Delete
Rows("1:1").Delete
Columns("A:A").Insert Shift:=xlToRight
Rows("1:1").Insert Shift:=xlDown
Cells(1, "A").FormulaR1C1 = "Group"
'Add Borders
Range("E1").Select
With Range( Cells(1, "E"), Cells(1, "E").Offset(dataLength + 3, 0)).Borders(xlEdgeLeft)
  .LineStyle = xlContinuous
  .ColorIndex = 0
  .TintAndShade = 0
  .Weight = xlThin
End With

With Range( Cells(4, "A"), Cells(4, "A").Offset(0, dataWidth + 3)).Borders(xlEdgeBottom)
  .LineStyle = xlContinuous
  .ColorIndex = 0
  .TintAndShade = 0
  .Weight = xlThin
End With

Dim top As Integer, left As Integer, gap As Integer
gap = Cells(1, 5).Left
With datasws
  top = .Cells(datafinishLength + 9, datastartWidth).Top
  left = .Cells(datafinishLength + 9, datastartWidth).Left
End With
Dim zz As Integer
If datasws.Shapes.Count > 0 Then
datasws.Shapes.SelectAll
Selection.Delete
End If

Dim shp As Shape
datasws.Activate
zz = ActiveWindow.Zoom
With Sheets("Information")
  .Activate
  Set shp = ActiveSheet.Shapes("Reset")
End With

With staffws
  .Activate
  Selection.Name = "Home"
  Selection.Characters.Text = "Home"
End With
With dataws
  .Activate
  .Buttons.Add(left, top, shp.Width, shp.Height).Select
  Selection.Name = "Home"
  Selection.Characters.Text = "Home"
  With Selection
    .Font.Name = "Calibri"
    .Font.FontStyle = "Regular"
    .Font.Size = 11 * (100 / zz)
    .OnAction = "goHome"
  End With
  With dataws.Shapes("Home")
    .ScaleWidth 100 / zz, msoFalse, _
    msoScaleFromTopLeft
    .ScaleHeight 100 / zz, msoFalse, _
    msoScaleFromTopLeft
  End With

.Buttons.Add((left * 8), top, shp.Width, shp.Height).Select
  Selection.Name = "Edit"
  Selection.Characters.Text = "Edit"
  With Selection
    .Font.Name = "Calibri"
    .Font.FontStyle = "Regular"
    .Font.Size = 11 * (100 / zz)
    .OnAction = "editing"
  End With
  With dataws.Shapes("Edit")
    .ScaleWidth 100 / zz, msoFalse, _
    msoScaleFromTopLeft
    .ScaleHeight 100 / zz, msoFalse, _
    msoScaleFromTopLeft
  End With
.Buttons.Add((left * 8), top, shp.Width, shp.Height).Select
    Selection.Name = "Order"
    Selection.Characters.Text = "Order"
With Selection
    .Font.Name = "Calibri"
    .Font.FontStyle = "Regular"
    .Font.Size = 11 * (100 / zz)
    .OnAction = "Ordering"
End With
With dataws.Shapes("Order")
    .ScaleWidth 100 / zz, msoFalse, msoScaleFromTopLeft
    .ScaleHeight 100 / zz, msoFalse, msoScaleFromTopLeft
End With
End With
ActiveSheet.Shapes.Range(Array("Edit", "Home", "Order")).Select
    Selection.ShapeRange.group.Name = "dataButtons"
With ActiveSheet.Shapes("dataButtons")
    .Height = 300
    .Width = 1800
End With

Range(Cells(1, 1), Cells(dataLength + 80, dataWidth + 19)).Select
ActiveWindow.Zoom = True
zz = ActiveWindow.Zoom
With dataws.
    .Shapes.Range(Array("dataButtons"))
    .ScaleWidth 10 / zz, msoFalse, msoScaleFromTopLeft
    .ScaleHeight 10 / zz, msoFalse, msoScaleFromTopLeft
End With
With .Buttons("Home")
End With
With .Buttons("Edit")
    .Visible = False
End With
With .Buttons("Order")
End With

End With
With Sheets("Information")
    .Shapes("Reset").Visible = True
    .Shapes("viewData").Visible = True
End With
Cells(1, 1).Select
With staffws
staffLength = .Cells(.Rows.count, "A").End(xlUp).Row
End With

Sheets("Information").Activate
' Range("Table1[[#All],[Levels]]").Select
' Selection.ListObject.ListColumns.Add
' Range("Table1[[#Headers],[Column1]]").Select
' ActiveCell.FormulaR1C1 = "Count"
Sheets("Information").Activate

Range("Table1[Count]").Select
ActiveCell.FormulaR1C1 = "=COUNTIF('Sample Link Matrix'!R5C3:R & dataWidth + 4 & "C3,[@Levels])"
' Range("Table2[[#All],[Departments]]").Select
' Selection.ListObject.ListColumns.Add
' Range("Table2[[#Headers],[Column1]]").Select
' ActiveCell.FormulaR1C1 = "Count"
Range("Table2[Count]").Select
ActiveCell.FormulaR1C1 = "=COUNTIF('Sample Link Matrix'!R5C2:R & dataWidth + 4 & "C2,[@Departments])"
'Columns("B:H").EntireColumn.AutoFit

'if count is greater than 0 then display message box warning
If checkAppraisers(staffLength, dataWidth) > 0 Then
    staffws.Cells(1, 1).Select
Else
    Sheets("Information").Select
End If

ActiveWindow.WindowState = xlMaximized
Application.ScreenUpdating = True
Exit Sub
errHandler1:
MsgBox ("No File named 'Sample Link Matrix.csv' found.")
Call Reset
Exit Sub
errHandler2:
MsgBox ("No File named 'Sample Staff List.csv' found.")
Call Reset
Exit Sub
errHandler3:
MsgBox ("An error has occurred. Please ensure that the files selected are in the correct format.")
Call Reset
Exit Sub
End Sub
Sub updateAppraisers()
Dim dataws As Worksheet, staffws As Worksheet
Dim dataLength As Integer, dataWidth As Integer, staffLength As Integer
Set dataws = Sheets("Sample Link Matrix")
Set staffws = Sheets("Sample Staff List")

'record the width and length of the matrix
With dataws
    dataLength = .Cells(.Rows.count, "A").End(xlUp).Row
    dataWidth = .Cells(1, Columns.count).End(xlToLeft).Column
    .Range(Cells(2, 2), Cells(dataLength, dataWidth)).Select
End With

'replace appraisers relationships with 9 instead of 1
With staffws
    staffLength = .Cells(.Rows.count, "A").End(xlUp).Row
End With
For i = 2 To staffLength
    With staffws
        Col = .Cells(i, 1).Value
        Row = .Cells(i, 1).Offset(0, 3).Value
    End With
    For j = 2 To dataLength
        With dataws
            If .Cells(j, 1) = Row Then
                For k = 2 To dataWidth
                    If .Cells(1, k) = Col Then .Cells(j, k).Value = 9
                End If
            End If
        End With
    Next j
Next i

If checkAppraisers(staffLength, dataWidth) > 0 Then
    MsgBox ("There are " & checkAppraisers(staffLength, dataWidth) & " out of " & staffLength & " appraisers that are assigned to employees not in their department." & ", it is recommended to assign an appraiser to an employee that is in the same department."
    & " These conflicts will be highlighted in red in the ""Sample Staff List"" worksheet.")
    staffws.Cells(1, 1).Select
Else
    Sheets("Information").Select
End If
End Sub

Function checkAppraisers(staffLength As Integer, dataWidth As Integer) As Integer
Dim staffws As Worksheet
Set staffws = Sheets("Sample Staff List")
Check if Appraiser is in same department as appraisee

Dim count, start, finish As Integer
Dim n As Range
Dim s As String
Dim rng1 As Range
Set rng1 = Range("C2:C" & staffLength)
count = 0
With Range("A1:F" & staffLength).Interior
    .Pattern = xlNone
    .TintAndShade = 0
    .PatternTintAndShade = 0
End With
For Each n In rng1
    With n
    If .Offset(1).Value <> "NA" Then
        start = InStr(.Offset(1).Value, ")")
        finish = InStr(.Offset(1).Value, ")")
        s = Mid(.Offset(1).Value, start + 1, finish - start - 1)
        If .Value <> s Then
            count = count + 1
            With Range(Cells(n.Row, 1), Cells(n.Row, 5)).Interior
                .Pattern = xlSolid
                .PatternColorIndex = xlAutomatic
                .Color = 255
                .TintAndShade = 0
                .PatternTintAndShade = 0
            End With
        End If
    End If
    End With
Next n

With Range(Cells(2, 1), Cells(dataWidth + 1, 5))
    .FormatConditions.Add Type:=xlExpression, Formula1:="=$C2 <> MID($D2,FIND("\"\"",$D2)+1,FIND("\"\"",$D2)-FIND("\"\"",$D2) -1)"
    .FormatConditions(1).SetFirstPriority
    With .FormatConditions(1)
        .Font.ThemeColor = xlThemeColorLight1
        .Font.TintAndShade = 0
    End With
    With .Interior
        .PatternColorIndex = xlAutomatic
        .ThemeColor = xlThemeColorLight1
        .TintAndShade = 0
        .Color = 255
    End With
End With
End If
End If
End With
Next n
checkAppraisers = count
End Function
Sub boundries()
Dim info As Worksheet, matrix As Worksheet
Dim rowStart As Integer, colStart As Integer
Dim rowEnd As Integer, colEnd As Integer
Dim dataLength As Integer, dataWidth As Integer
Dim i As Integer, rng As Range
Dim level As String, department As String

Set info = Sheets("Information")
Set matrix = Sheets("Sample Link Matrix")

With info
    level = .Range("viewLevel").Value
    department = .Range("viewDepartment").Value
    dataLength = .Range("dataLength").Value
    dataWidth = .Range("dataWidth").Value
End With

With matrix
    If level = "All Levels" And department = "All Departments" Then
        rowStart = 5
        colStart = 5
        rowEnd = dataLength + 4
        colEnd = dataWidth + 4
    ElseIf department = "All Departments" Then
        rowStart = 5
        rowEnd = dataLength + 4
        For i = 5 To dataWidth + 4
            If .Cells(3, i).Value = level Then Exit For
        Next
        colStart = i
        For i = dataWidth + 4 To 5 Step -1
            If .Cells(3, i).Value = level Then Exit For
        Next
        colEnd = i
    ElseIf level = "All Levels" Then
        For i = 5 To dataLength + 4
            If .Cells(i, 2).Value = department Then Exit For
        Next
        rowStart = i
        For i = dataLength + 4 To 5 Step -1
            If .Cells(i, 2).Value = department Then Exit For
        Next
        rowEnd = i
        For i = 5 To dataWidth + 4
            If .Cells(2, i).Value = department Then Exit For
        Next
        colStart = i
        For i = dataWidth + 4 To 5 Step -1
            If .Cells(2, i).Value = department Then Exit For
        Next
    End If
End With
Next  
colEnd = i  
Else  
For i = 5 To dataLength + 4  
  If .Cells(i, 2).Value = department Then Exit For  
Next  
rowStart = i  
For i = dataLength + 4 To dataLength + 4 Step -1  
  If .Cells(i, 2).Value = department Then Exit For  
Next  
rowEnd = i  
For i = 5 To dataWidth + 4  
  If .Cells(2, i).Value = department And .Cells(3, i).Value = level Then Exit For  
Next  
colStart = i  
For i = dataWidth + 4 To dataWidth + 4 Step -1  
  If .Cells(2, i).Value = department And .Cells(3, i).Value = level Then Exit For  
Next  
colEnd = i  
End If  
End With  
With info  
.Range("datastartLength").Value = rowStart  
.Range("datastartWidth").Value = colStart  
.Range("datafinishLength").Value = rowEnd  
.Range("datafinishWidth").Value = colEnd  
End With  
End Sub  
Sub ccc()  
MsgBox Range("Table2[Departments]").Address()  
End Sub  
Sub Ordering()  
If ActiveWorkbook.Worksheets.count = 1 Then GoTo errHandler1  
Dim datas As Worksheet  
Dim dataLength As Integer, dataWidth As Integer, minAverage As Integer  
Set datas = Sheets("Sample Link Matrix")  
Application.ScreenUpdating = False  
Sheets("Information").Activate  
ActiveSheet.Shapes("departmentDropDown").ControlFormat.Value = 1  
With ActiveSheet.Shapes("departmentDropDown").ControlFormat  
  Range("viewDepartment").Value = .List(.Value)  
End With  
ActiveSheet.Shapes("levelDropDown").ControlFormat.Value = 1  
With ActiveSheet.Shapes("levelDropDown").ControlFormat  
  Range("viewLevel").Value = .List(.Value)  
End With
Call listBoxChng
    Application.ScreenUpdating = False

With Sheets("Information")
    Dim nshapes As Integer
    For nshapes = 1 To .Shapes.count
        .Shapes(nshapes).Visible = True
    Next nshapes
End With
On Error GoTo errHandler3

With Sheets("Information")
    dataLength = .Range("dataLength").Value
    dataWidth = .Range("dataWidth").Value
    minAverage = .Range("minAverage").Value
End With
Call boundries
With Sheets("Information")
    datastartLength = .Range("datastartLength").Value
    datastartWidth = .Range("datastartWidth").Value
    datafinishLength = .Range("datafinishLength").Value
    datafinishWidth = .Range("datafinishWidth").Value
End With
    On Error GoTo errHandler2
dataws.Activate
    Cells(1, 1).Select
    ActiveWindow.Zoom = 10

"Assigning Groups"""""""""

'assign groups
With Cells(5, "A")
    .Formula = "=LOOKUP($B5,'Information'!" & Range("Table2[Departments]").Address() & ",'Information'!" & Range("Table2[No.]").Address() & ")&""""
    .AutoFill Destination:=Range(Cells(5, "A"), .Offset(dataLength - 1, 0)), Type:=xlFillDefault
End With
With Cells(1, "E")
    .Formula = "=LOOKUP(E$2,'Information'!" & Range("Table2[Departments]").Address() & ",'Information'!" & Range("Table2[No.]").Address() & ")&""""
    .AutoFill Destination:=Range(Cells(1, "E"), .Offset(0, dataWidth - 1)), Type:=xlFillDefault
End With
'Formulas to values
With Columns("A")
    .Copy
    .PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
:=False, Transpose:=False
End With
With Rows(1)
  .Copy
  .PasteSpecial Paste:=xlPasteValues, Operation:=xlNone, SkipBlanks _
  :=False, Transpose:=False
End With
Application.CutCopyMode = False
Columns("A:A").ColumnWidth = 8

Call sortandGroup

Range(Cells(1, 1), Cells(dataLength + 80, dataWidth + 19)).Select
ActiveWindow.Zoom = True

Dim zz As Integer
Dim shp As Shape
dataws.Activate
zz = ActiveWindow.Zoom
With Sheets("Information")
Set shp = .Shapes("Reset")
End With
With dataws
  .Activate
  .Shapes("dataButtons").top = .Cells(dataLength + 9, datastartWidth).top
  .Shapes("dataButtons").left = .Cells(dataLength + 9, datastartWidth).left
  .Shapes("dataButtons").Height = 300
  .Shapes("dataButtons").Width = 1800
  With .Shapes.Range(Array("dataButtons"))
    .ScaleWidth 10 / zz, msoFalse, msoScaleFromTopLeft
    .ScaleHeight 10 / zz, msoFalse, msoScaleFromTopLeft
  End With
  With .Buttons("Home")
  End With
  With .Buttons("Edit")
  End With
End With

dataws.Shapes("Order").Visible = False
'dataws.Shapes("Edit").Visible = True

Cells(1, 1).Select
ActiveWindow.Zoom = 10

Application.ScreenUpdating = True
Exit Sub
errHandler1:
    MsgBox ("Please open files first.")
Call Reset
Exit Sub
errHandler2:
    MsgBox ("An error has occurred. Please ensure files are formatted correctly")
Exit Sub
errHandler3:
    MsgBox ("Select Department First")
Exit Sub
End Sub
Sub Calculate()
    On Error GoTo errHandler
    Dim dataws As Worksheet
    Dim minAverage As Double
    Dim numGroups As Integer
    Dim dataLength As Integer, dataWidth As Integer, datastartLength As Integer
    Dim datastartWidth As Integer, datafinishLength As Integer, datafinishWidth As Integer
    Set dataws = ActiveSheet

    With Sheets("Information")
        minAverage = .Range("minAverage").Value
        datastartLength = .Range("datastartLength").Value
        datastartWidth = .Range("datastartWidth").Value
        datafinishLength = .Range("datafinishLength").Value
        datafinishWidth = .Range("datafinishWidth").Value
        dataLength = datafinishLength - (datastartLength - 1)
        dataWidth = datafinishWidth - (datastartWidth - 1)
        datastartLength = 5
        datastartWidth = 5
        datafinishLength = dataLength + 4
        datafinishWidth = dataWidth + 4

        If .Range("viewDepartment").Value = "All Departments" And .Range("viewLevel").Value <> "All Levels" Then
            With dataws
                dataWidth = .Cells(3, Columns.count).End(xlToLeft).Column - 4
            End With
        End If

    'Count number of groups
    Cells(2, 1).FormulaArray = "='SUM(IF(FREQUENCY(VALUE(R" & datastartLength & "C1:R" & datafinishLength & "C1),VALUE(R" & datastartLength & "R" & datafinishLength & "C1:R" & datafinishLength & "C1))>0,1))"
    numGroups = Cells(2, 1).Value

    dataws.Activate
    End Sub
'Calculate sum in groups, sum in column and average
With Cells(dataLength + 6, "E")
  .Formula = "=COUNTIFS(E$5:E$" & dataLength + 4 & ",","" & 0,$A$5:$A$" & dataLength + 4 & ",E$1"
  .AutoFill Destination:=Range(Cells(dataLength + 6, "E"), .Offset(0, dataWidth - 1)), Type:=xlFillDefault
End With

With Cells(dataLength + 7, "E")
  .Formula = "=COUNTIF(E$5:E$" & dataLength + 4 & ",","" & 0)
  .AutoFill Destination:=Range(Cells(dataLength + 7, "E"), .Offset(0, dataWidth - 1)), Type:=xlFillDefault
End With

With Cells(dataLength + 8, "E")
  .Formula = "=E" & dataLength + 6 & "/ E" & dataLength + 7
  .AutoFill Destination:=Range(Cells(dataLength + 8, "E"), .Offset(0, dataWidth - 1)), Type:=xlFillDefault
End With

'Calculate sum in groups, sum in Rows and average
With Cells(5, dataWidth + 6)
  .FormulaR1C1 = "=COUNTIFS(RC5:RC" & dataWidth + 4 & ",","" & 0,R1C5:R1C" & dataWidth + 4 & ",,RC1"
  .AutoFill Destination:=Range(Cells(5, dataWidth + 6), .Offset(dataLength - 1, 0)), Type:=xlFillDefault
End With

With Cells(5, dataWidth + 7)
  .FormulaR1C1 = "=COUNTIF(RC5:RC" & dataWidth + 4 & ",","" & 0)
  .AutoFill Destination:=Range(Cells(5, dataWidth + 7), .Offset(dataLength - 1, 0)), Type:=xlFillDefault
End With

With Cells(5, dataWidth + 8)
  .FormulaR1C1 = "=If(RC[-1] = 0, 1,RC[-2]/RC[-1])"
  .AutoFill Destination:=Range(Cells(5, dataWidth + 8), .Offset(dataLength - 1, 0)), Type:=xlFillDefault
End With

'Copy department names and paste
Range(Cells(dataStartLength, 1), Cells(dataFinishLength, 2)).AdvancedFilter
Action:=xlFilterCopy, CopyToRange:=Cells(dataFinishLength + 7, 2), Unique:=True
Range(Cells(dataFinishLength + 7, 2), Cells(dataFinishLength + 7, 3)).Delete
Shift:=xlShiftUp

'Add number of Employees in each group
With Cells(dataLength + 11, "D")
  .FormulaR1C1 = "=COUNTIF(R1C5:R1C" & dataWidth + 4 & ",,RC2"
  .AutoFill Destination:=Range("D" & dataLength + 11 & ":D" & dataLength + 11 + numGroups - 1), Type:=xlFillDefault
End With
'sum for each group
With Cells(dataLength + 11, "E")
    .FormulaR1C1 = "=If( R" & dataLength + 7 & "C = 0 , 0,(COUNTIFS(R5C:R" & dataLength
+ 4 & ","&"0,R5C1:R" & dataLength + 4 & ","C1,RC2)/R" & dataLength + 7 & "C))")
"/(SUM(R" & dataLength + 11 & ","C4:R" & dataLength + 17 & ","C4)-RC4)/100"
    .AutoFill Destination:=Range(Cells(dataLength + 11, "E"), .Offset(0, dataWidth - 1)),
Type:=xlFillDefault
End With
With Range(Cells(dataLength + 11, "E"), Cells(dataLength + 11, "E").Offset(0, dataWidth - 1))
    .AutoFill Destination:=Range(Cells(dataLength + 11, "E"), Cells(dataLength + 11 +
numGroups - 1, "E").Offset(0, dataWidth - 1)), Type:=xlFillDefault
End With

'Copy department names and paste
Range(Cells(datafinishLength + 7, 2), Cells(datafinishLength + 7 + numGroups - 1, 3)).Copy
Cells(2, datafinishWidth + 7).PasteSpecial Transpose:=True

'Add number of managers in each group
With Cells(4, dataWidth + 11)
    .FormulaR1C1 = "="&COUNTIF(R5C1:R" & dataLength + 4 & ","C1,RC2)"
    .AutoFill Destination:=Range(Cells(4, dataWidth + 11), Cells(4, dataWidth + 11 +
numGroups - 1)), Type:=xlFillDefault
End With
'sum for each group
With Cells(5, dataWidth + 11)
    .FormulaR1C1 = "="&If(R5C1:R" & dataWidth + 7 & "," = 0,(SUMIF(R1C5:R1C" & dataWidth + 4 & ","R2C,RC5:RC" & dataWidth + 4 & ",")/RC" & dataWidth + 7 & ",")" "/(SUM(R4C" &
dataWidth + 11 & ","&"R4C" & dataWidth + 17 & ",")-R4C)/100"
    .AutoFill Destination:=Range(Cells(5, dataWidth + 11), .Offset(dataLength - 1, 0)),
Type:=xlFillDefault
End With
With Range(Cells(5, dataWidth + 11), Cells(5, dataWidth + 11).Offset(dataLength - 1, 0))
    .AutoFill Destination:=Range(Cells(5, dataWidth + 11), Cells(5, dataWidth + 11 +
numGroups - 1).Offset(dataLength - 1, 0)), Type:=xlFillDefault
End With

'Insert group cumulative count
Dim groupCount As Range
Dim gc As Integer
Dim i As Range
Dim c As Integer
C = 1
Set groupCount = Cells(4, dataWidth + 11)
For Each i In Range(Cells(5, dataWidth + 10), Cells(dataLength + 4, dataWidth + 10))
    i.Value = c
    If (c = CInt(groupCount.Value)) Then
        c = 0
        Set groupCount = groupCount.Offset(1)
End If
   c = c + 1
Next i

"RECOMMENDED GROUP
'For columns
With Cells(dataLength + 11 + numGroups + 1, 5)
   .Formula = "=INDEX(R[" & numGroups + 1 & "]C2:R[-2]C2,MATCH(Max(R[" & numGroups + 1 & "]C:R[-2]C),R[" & numGroups + 1 & "]C:R[-2]C,0),1)"
   .AutoFill Destination:=Range(Cells(dataLength + 11 + numGroups + 1, 5),
   Cells(dataLength + 11 + numGroups + 1, dataWidth + 4)), Type:=xlFillDefault
End With

'For Rows
With Cells(5, dataWidth + 11 + numGroups + 1)
   .FormulaR1C1 = ",IF(COUNTIF(RC[" & numGroups + 1 & "]:RC[-2],MAX(RC[" & numGroups + 1 & "]:RC[-2]))>1,-1,INDEX(R2C[" & numGroups + 1 & "]:R2C[-2],1,MATCH(Max(RC[" & numGroups + 1 & "]:RC[-2]),RC[" & numGroups + 1 & "]:RC[-2],0))))"
   .AutoFill Destination:=Range(Cells(5, dataWidth + 11 + numGroups + 1),
   Cells(dataLength + 4, dataWidth + 11 + numGroups + 1)), Type:=xlFillDefault
End With

"""FORMATTING""
'Highlight cells below the set average
With Range(Cells(dataLength + 8, 5), Cells(dataLength + 8, 5).Offset(0, dataWidth - 1))
   .FormatConditions.Delete
   .FormatConditions.Add Type:=xlCellValue, Operator:=xlLess, _
   Formula1:="=Information!$L$7"
   .FormatConditions(.FormatConditions.count).SetFirstPriority
   With .FormatConditions(1).Interior
   .PatternColorIndex = xlAutomatic
   .Color = 255
   .TintAndShade = 0
   End With
   .FormatConditions(1).StopIfTrue = False
End With

'Format on calculations on columns
With Range(Cells(dataLength + 11, 5), Cells(dataLength + 11, 5).Offset(numGroups - 1,
dataWidth - 1))
   .FormatConditions.Delete
   .FormatConditions.Add Type:=xlExpression, _
   Formula1:="=AND(RC2<>R1C,RC = MAX(R" & dataLength + 11 & ")C:R" & dataLength + 11 + numGroups - 1 & ")""
   .FormatConditions(1).SetFirstPriority
   With .FormatConditions(1)
   .Font.ThemeColor = xlThemeColorLight1
   .Font.TintAndShade = 0
   End With
.Interior.PatternColorIndex = xlAutomatic
.Interior.Color = 255
.Interior.TintAndShade = 0
.StopIfTrue = False
End With

.FormatConditions.Add Type:=xlExpression, _
With .FormatConditions(2)
.Font.ThemeColor = xlThemeColorLight1
.Font.TintAndShade = 0
.Interior.PatternColorIndex = xlAutomatic
.Interior.Color = 65535
.Interior.TintAndShade = 0
.StopIfTrue = False
End With
End With

'Format on calculations on rows
With Range(Cells(5, dataWidth + 11), Cells(5, dataWidth + 11).Offset(dataLength - 1, numGroups - 1))
.FormatConditions.Delete
.FormatConditions.Add Type:=xlExpression, _
Formula1:="=AND(R2C<>RC1,RC = MAX(RC & dataWidth + 11 & ":RC" & dataWidth + 11 + numGroups - 1 & ")")"
.FormatConditions(1).SetFirstPriority
With .FormatConditions(1)
.Font.ThemeColor = xlThemeColorLight1
.Font.TintAndShade = 0
.Interior.PatternColorIndex = xlAutomatic
.Interior.Color = 255
.Interior.TintAndShade = 0
.StopIfTrue = False
End With
End With

.FormatConditions.Add Type:=xlExpression, _
Formula1:="=OR(R2C=RC1,RC>LOOKUP(RC1,R2C & dataWidth + 11 & ":R2C" & dataWidth + 11 + numGroups - 1 & ",RC" & dataWidth + 11 & ":RC" & dataWidth + 11 + numGroups - 1 & ")")"
With .FormatConditions(2)
.Font.ThemeColor = xlThemeColorLight1
.Font.TintAndShade = 0
.Interior.PatternColorIndex = xlAutomatic
.Interior.Color = 65535
.Interior.TintAndShade = 0
.StopIfTrue = False
End With
End With

'Format Recomendations
With Range(Cells(dataLength + 11 + numGroups + 1, 5), Cells(dataLength + 11 + numGroups + 1, dataWidth + 4))
  .FormatConditions.Delete
  .FormatConditions.Add Type:=xlExpression, 
    Formula1:="=R" & dataLength + 11 + numGroups + 1 & "C <> R1C"
  .FormatConditions(1).SetFirstPriority
With .FormatConditions(1)
  .Font.ThemeColor = xlThemeColorLight1
  .Font.TintAndShade = 0
  .Interior.PatternColorIndex = xlAutomatic
  .Interior.Color = 255
  .Interior.TintAndShade = 0
  .StopIfTrue = False
End With
End With

With Range(Cells(5, dataWidth + 11 + numGroups + 1), Cells(dataLength + 4, dataWidth + 11 + numGroups + 1))
  .FormatConditions.Delete
  .FormatConditions.Add Type:=xlExpression, 
    Formula1:="=RC <> RC1"
  .FormatConditions(1).SetFirstPriority
With .FormatConditions(1)
  .Font.ThemeColor = xlThemeColorLight1
  .Font.TintAndShade = 0
  .Interior.PatternColorIndex = xlAutomatic
  .Interior.Color = 255
  .Interior.TintAndShade = 0
  .StopIfTrue = False
End With
End With

Dim top As Integer, left As Integer
With datasws
  top = .Cells(dataLength + 14 + numGroups, 5).top
  left = .Cells(dataLength + 14 + numGroups, 5).left
End With
Dim zz As Integer

Dim shp As Shape
datasws.Activate
zz = ActiveWindow.Zoom
With Sheets("Information")
  .Activate
  Set shp = ActiveSheet.Shapes("Reset")
End With
With Sheets("Editing")
  .Activate
  If .Shapes.count > 0 Then
    .Shapes.SelectAll
    Selection.Delete
  End If
End If
With datasw.
  .Activate
  .Buttons.Add(left, top, shp.Width, shp.Height).Select
  Selection.Name = "exitEditing"
  Selection.Characters.Text = "Exit Edit"
  With Selection
    .Font.Name = "Calibri"
    .Font.FontStyle = "Regular"
    .Font.Size = 11
    .OnAction = "exitEditing"
  End With

  .Buttons.Add(left * 2.5, top, shp.Width, shp.Height).Select
  Selection.Name = "Sort"
  Selection.Characters.Text = "Sort"
  With Selection
    .Font.Name = "Calibri"
    .Font.FontStyle = "Regular"
    .Font.Size = 11
    .OnAction = "SortingonRows"
  End With

  .Buttons.Add(left * 4, top, shp.Width, shp.Height).Select
  Selection.Name = "export"
  Selection.Characters.Text = "Export"
  With Selection
    .Font.Name = "Calibri"
    .Font.FontStyle = "Regular"
    .Font.Size = 11
    .OnAction = "export"
  End With

End With

ActiveSheet.Shapes.Range(Array("exitEditing", "Sort", "Export")).Select
Selection.ShapeRange.group.Name = "editingsheetButtons"
With ActiveSheet.Shapes("editingsheetButtons")
  .Height = 40
  .Width = 300
End With

Range(Cells(1, 1), Cells(dataLength + 30, dataWidth + 10)).Select
ActiveWindow.Zoom = True
zz = ActiveWindow.Zoom
With datasw.
  With .Shapes.Range(Array("editingsheetButtons"))
    .ScaleWidth 100 / zz, msoFalse, msoScaleFromTopLeft
    .ScaleHeight 100 / zz, msoFalse, msoScaleFromTopLeft
  End With
End With
With .Buttons("exitEditing")
End With
With .Buttons("Sort")
End With
With .Buttons("Export")
End With
End With
' Exit Sub

'Calculate average:

With Range(Cells(dataLength + 5, dataWidth + 4 + numGroups), Cells(dataLength + 7, dataWidth + 10 + numGroups))
  .HorizontalAlignment = xlLeft
  .VerticalAlignment = xlCenter
  .MergeCells = True
  .FormulaR1C1 = "=""AVERAGE: "&ROUND(AVERAGE(R" & dataLength + 8 & "C5:R" & dataLength + 8 & "C" & dataWidth + 4 & "),3)"
  .Font.Size = 18
  .Font.Bold = True
End With

With Range(Cells(dataLength + 9, dataWidth + 4 + numGroups), Cells(dataLength + 11, dataWidth + 10 + numGroups))
  .HorizontalAlignment = xlLeft
  .VerticalAlignment = xlCenter
  .MergeCells = True
  .FormulaR1C1 = "=""MINIMUM: "&ROUND(MIN(R" & dataLength + 8 & "C5:R" & dataLength + 8 & "C" & dataWidth + 4 & "),3)"
  .Font.Size = 18
  .Font.Bold = True
End With

With Range(Cells(dataLength + 13, dataWidth + 4 + numGroups), Cells(dataLength + 15, dataWidth + 10 + numGroups))
  .HorizontalAlignment = xlLeft
  .VerticalAlignment = xlCenter
  .MergeCells = True
  .FormulaR1C1 = "=""SWAPS: "&ROUND(COUNTIF(R" & dataLength + 8 & "C5:R" & dataLength + 8 & "C" & dataWidth + 4 & ",",<"&.7),3)"
  .Font.Size = 18
  .Font.Bold = True
End With

Exit Sub
With Range(Cells(dataLength + 63 + numGroups, 25), Cells(dataLength + 73 + numGroups, 50))
 .HorizontalAlignment = xlLeft
 .VerticalAlignment = xlCenter
 .MergeCells = True
 .FormulaR1C1 = "=""AVERAGE: 0.4118"
 .Font.Size = 72
 .Font.Bold = True
End With

With Range(Cells(dataLength + 63 + numGroups, 55), Cells(dataLength + 73 + numGroups, 80))
 .HorizontalAlignment = xlLeft
 .VerticalAlignment = xlCenter
 .MergeCells = True
 .FormulaR1C1 = "=""MINIMUM: 0"
 .Font.Size = 72
 .Font.Bold = True
End With

Exit Sub
errHandler:
 MsgBox ("An error has occured. Please ensure files are formatted correctly")
Call Reset
Exit Sub
End Sub

Sub exitEditing()
Dim ws As Worksheet
Application.DisplayAlerts = False
For Each ws In Worksheets
If ws.Name = "Editing" Then ws.Delete
Next
Application.DisplayAlerts = True
Sheets("Sample Link Matrix").Activate
Cells(1, 1).Select
End Sub

Sub sortandGroup()
Dim dataLength As Integer, dataWidth As Integer, minAverage As Integer

With Sheets("Information")
dataLength = .Range("dataLength").Value
dataWidth = .Range("dataWidth").Value
minAverage = .Range("minAverage").Value
End With

'Format the groups
With Range(Cells(5, 5), Cells(dataLength + 4, dataWidth + 4))
 .FormatConditions.Add Type:=xlExpression,
 Formula1:="=MID($A5,1,FIND(""."",$A5))=MID(E$1,1,FIND(""."",E$1))"
End With
With .FormatConditions(.FormatConditions.count)
.Font.ThemeColor = xlThemeColorLight1
.Font.TintAndShade = 0
.Interior.PatternColorIndex = xlAutomatic
.Interior.Color = 65535
.Interior.TintAndShade = 0
.StopIfTrue = False
End With
End With

'Sort the matrix''''''''
With ActiveWorkbook.Worksheets("Sample Link Matrix").Sort
 .SortFields.Clear
 'sort groups
 'by rows
 .SortFields.Add Key:= _
   Range(Cells(5, "A"), Cells(5, "A").Offset(dataLength - 1, 0)),
SortOn:=xlSortOnValues, Order:=xlAscending, DataOption _
   :=xlSortNormal
 .SetRange Range(Cells(5, "A"), Cells(5, "A").Offset(dataLength - 1, dataWidth + 3))
 .Header = xlGuess
 .MatchCase = False
 .Orientation = xlTopToBottom
 .SortMethod = xlPinYin
 .Apply

 'by columns
 .SortFields.Clear
 .SortFields.Add Key:= _
   Range(Cells(1, "E"), Cells(1, "E").Offset(0, dataWidth - 1)), SortOn:=xlSortOnValues,
Order:=xlAscending, DataOption:= _
   xlSortNormal
 .SetRange Range(Cells(1, "E"), Cells(1, "E").Offset(dataLength + 3, dataWidth - 1))
 .Header = xlGuess
 .MatchCase = False
 .Orientation = xlLeftToRight
 .SortMethod = xlPinYin
 .Apply
End With

'Format the employee positions
With Range(Cells(5, "E"), Cells(5, "E").Offset(dataLength - 1, dataWidth - 1))
 .FormatConditions.Add Type:=xlExpression, Formula1:= _
   "=AND(INDEX(Information!$E$4:$E$10,MATCH($C5,Information!$E$4:$E$10,0)-1)=$E$3,MID($A5,1,FIND("."$,A5))=MID(E$1,1,FIND("."$,E$1)))"
 With .FormatConditions(.FormatConditions.count - 1)
 .Font.ThemeColor = xlThemeColorLight1
 .Font.TintAndShade = 0
 .Interior.PatternColorIndex = xlAutomatic
 .Interior.Color = 49407
With Range(Cells(5, "E"), Cells(5, "E").Offset(dataLength - 1, dataWidth - 1))
    .FormatConditions.Add Type:=xlExpression, Formula1:=
        "=IF(LEN($A5)-FIND(""","",$A5) >0, MID($A5,FIND(""","",$A5)+1,LEN($A5)-
        FIND(""","",$A5)))=IF(LEN(E$1)-FIND(""","",E$1) >0, MID(E$1,FIND(""","",E$1)+1,LEN(E$1)-
        FIND(""","",E$1)),1)"
    With .FormatConditions(.FormatConditions.count - 1)
        .Font.ThemeColor = xlThemeColorLight1
        .Font.TintAndShade = 0
        .Interior.PatternColorIndex = xlAutomatic
        .Interior.Color = 49407
        .Interior.TintAndShade = 0
        .StopIfTrue = False
    End With
End With
' sort position and ID
With ActiveWorkbook.Worksheets("Sample Link Matrix").Sort
    ' byrows
    .SortFields.Clear
    .SortFields.Add Key:=
        Range(Cells(5, "A"), Cells(5, "A").Offset(dataLength - 1, 0)),
    SortOn:=xlSortOnValues, Order:=xlAscending, DataOption _
    :=xlSortNormal
    .SortFields.Add Key:=
        Range(Cells(5, "C"), Cells(5, "C").Offset(dataLength - 1, 0)),
    SortOn:=xlSortOnValues, Order:=xlAscending, CustomOrder _
    :="Analyst,Consultant,Senior Consultant,Manager,Senior Manager,Director,Partner"
_     , DataOption:=xlSortNormal
    .SortFields.Add Key:=
        Range(Cells(5, "D"), Cells(5, "D").Offset(dataLength - 1, 0)),
    SortOn:=xlSortOnValues, Order:=xlAscending, DataOption _
    :=xlSortTextAsNumbers
    .SetRange Range(Cells(5, "A"), Cells(5, "A").Offset(dataLength - 1, dataWidth + 3))
    .Header = xlGuess
    .MatchCase = False
    .Orientation = xlTopToBottom
    .SortMethod = xlPinYin
    .Apply

    ' bycols
    .SortFields.Clear
    .SortFields.Add Key:=
        Range(Cells(1, "E"), Cells(1, "E").Offset(0, dataWidth - 1)), SortOn:=xlSortOnValues,
    Order:=xlAscending, DataOption:= _
    xlSortNormal
    .SortFields.Add Key:=
        Range(Cells(3, "E"), Cells(3, "E").Offset(0, dataWidth - 1)), SortOn:=xlSortOnValues,
    Order:=xlAscending, CustomOrder _
"Analyst, Consultant, Senior Consultant, Manager, Senior Manager, Director, Partner"

```vbscript
On Error Resume Next
Dim data, dataWidth, dataLength, Cells, Range, Key, SortFields, DataOption
Dim xlSortNormal, xlSortOnValues, xlAscending, xlSortTextAsNumbers

data = Cells(4, "E")
Cells(4, "E").Offset(0, dataWidth - 1)
SortOn = xlSortOnValues
Order = xlAscending
DataOption = xlSortTextAsNumbers

End Sub
```

**R CODE**

```r
data <- read.csv("Sample Link Matrix.csv")
matr <- as.matrix(data)
#install.packages('gtools')
library(gtools)
nrows <- length(matr[,1])
ncols <- length(matr[1,])
besti <- 0
bestj <- 0
# slope of diagonal line
slope <- (nrows-1)/(ncols-1)
bestDis <- 0
totDis <- 0
for (m in 1:nrows){
  for (n in 1:ncols){
    if (matr[m,n] == 1){
      # perpendicular distance
      bestDis = bestDis + (Mod(slope*(n-1) - Mod(m-nrows) + (nrows-1))/sqrt(slope^2 +1))
    }
  }
}
x <- c(1:nrows)
y <- c(1:ncols)
permsx <- permutations(nrows, nrows, x)
permsy <- permutations(ncols, ncols, y)
# going through permutations
```
for (i in 1:length(permsx[,1])){
  for (j in 1:length(permsy[,1])){
    matr <- as.matrix(data)
    matr <- matr[permx[i,], permy[j,], drop=FALSE]
    # going through matrix of current permutation
    totDis <- 0
    for (m in 1:nrows){
      for (n in 1:ncols){
        if (matr[m,n] == 1){
          # perpendicular distance
          totDis = totDis + (Mod(slope*(n-1) - Mod(m-nrows) + (nrows-1))/sqrt(slope^2 +1))
        }
      }
    }
    if(totDis<bestDis){
      besti <- i
      bestj <- j
      bestDis <- totDis
    }
  }
}
View(matr[permx[besti,],permy[bestj,]])

matr <- matrix(c(1,1,0,0,1,0,0,1,1,
  0,1,0,0,0,0,1,1,1,
  0,0,1,1,0,1,0,1,1,
  1,1,0,1,0,0,1,0,0,
  0,1,0,1,0,0,0,1,1,
  1,0,1,0,1,0,0,0,0,
  1,1,0,1,0,1,0,0,0,
  0,0,0,1,0,0,0,1,0,
  0,1,0,1,0,0,0,0,0)
9,9,byrow = TRUE)
colnames(matr) <- c("A1","A2","A3","A4","A5","A6","A7","A8","A9")
rownames(matr) <- c("M1","M2","M3","M4","M5","M6","M7","M8","M9")
columns <- colnames(matr)
rows <- rownames(matr)
#-----------------------------------------------
bertinplot(matr, highlight = TRUE, options = list(panel=panel.squares, spacing = 0))
o <- seriate(matr, method = "BEA", control = list(rep = 11))
bertinplot(matr, o, highlight = TRUE, options = list(panel=panel.squares, spacing = 0, reverse = TRUE))
bertinplot(matr,o)
matr <- matr[get_order(o),]
matr <- t(matr)
#-----------------------------------------------
x <- seriate(matr, "BEA")
ser_permutation(matr,)
View(matr)

?seriation

hmap(matr)
pimage(realMatr)
bertinplot(matr,options=list(pop=FALSE))
bertin_cut_line(matr)
matr[get_order(o),c(6,3,1,8,2,5,7,9,4)]

getwd()
setwd("C:/Users/Pauline/Documents/Ciaran M. Doyle/4th Year/Modules/FYP/Data/Excel Sheets/")
data <- read.csv("AnalystsApp=9.csv")
rwnames<- data[,1]
data<-data[,1]
clnames<- colnames(data)
realMatr <- as.matrix(data)
nrows <- length(realMatr[,1])
ncols <- length(realMatr[1,])
rownames(realMatr)<- rwnames
colnames(realMatr)<- clnames
matr<-realMatr
rr<- realMatr[get_order(seriate(realMatr, "ME")),c(6,3,1,8,2,5,7,9,4)]
write.csv(matr, file = "C:/Users/Pauline/Documents/Ciaran M. Doyle/4th Year/Modules/FYP/Data/Excel Sheets/Brio2.csv")

#CAseriation
matr<- matrix(c(1,1,0,0,1,0,0,1,1,
    0,1,0,0,0,0,0,1,1,
    0,0,0,1,1,0,1,0,1,
    1,1,1,0,0,0,1,0,0,
    0,1,0,0,1,0,0,0,1,
    1,0,1,0,0,1,0,0,0,
    1,1,0,0,1,0,1,0,0,
    0,0,0,1,0,0,0,1,0,
    0,1,0,1,0,0,0,0,0),
   9,9,byrow = TRUE)

colnames(matr)<- c("A1","A2","A3","A4","A5","A6","A7","A8","A9")
rownames(matr)<- c("M1","M2","M3","M4","M5","M6","M7","M8","M9")
columns <- colnames(matr)
rows <- rownames(matr)
check.ca.plot(matr)
sort.table(matr,1)
plot.clusters.rows(matr,1,4)
plot.clusters.cols(matr,1,1)

data("perfect_seriation")
check.ca.plot(perfect_seriation,1,2)
sort.table(perfect_seriation,1)
plot.clusters.rows(perfect_seriation,1,2)
plot.clusters.cols(perfect_seriation,1,2)
getwd()
setwd("C:/Users/Pauline/Documents/Ciaran M. Doyle/4th Year/Modules/FYP/Data/Excel Sheets")
data <- read.csv("ConsultantsApp=9.csv")
rwnames<- data[,1]
data<-data[,,-1]
clnames< colnames(data)
realMatr <- as.matrix(data)
nrows <- length(realMatr[,1])
ncols <- length(realMatr[1,])
rownames(realMatr)<- rwnames
colnames(realMatr)<- clnames
matr<-realMatr

o <- seriate(matr, method = "BEA", control = list(rep = 194))
matr <- matr[get_order(o),]

orderedCols <- c()
for(i in 1:nrows){
  orderedCols <- c(orderedCols,which(matr[i,]%in% 9))
}
matr<- matr[,orderedCols]
write.csv(matr, file = "C:/Users/Pauline/Documents/Ciaran M. Doyle/4th Year/Modules/FYP/Data/Excel Sheets/finitoConsults.csv")

plot(hclust(as.dist(matr)))
hmap(matr)
hma

x.hc <- hclust(as.dist(1-cor(matr)))
## Run complementary hierarchical clustering.
x.chc <- compHclust(matr,x.hc)
xp <- x.chc$x.prime
x.gi <- x.chc$gene.imp
xp.hc <- hclust(as.dist(1-cor(xp)))
xp.gi <- compHclust(xp,xp.hc)$gene.imp
compHclust.heatmap(matr,x.hc,x.gi,d.title="Initial Clustering",d.ht= .25)
## The complementary clustering.

```r
compHclust.heatmap(xp.xp.hc.xp.gi.d.title="Complementary Clustering")
```

```r
n <- cutree(x.hc,k=5)
n <- sort(n)
compHclust(x = matr, xhc )
dendroPlot(matr[1,])
d.mar
```
REFERENCES


DELOITTE
Performance Appraisal Scheduling

March 2015

Prepared by: Ciarán Doyle

Supervisor: Prof. Frank Bannister
**TABLE OF CONTENTS**

<table>
<thead>
<tr>
<th>NO.</th>
<th>SECTION</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>BASIC FEATURES</td>
<td>1</td>
</tr>
<tr>
<td>1.1</td>
<td>Enabling Macros</td>
<td>1</td>
</tr>
<tr>
<td>1.2</td>
<td>Home Screen</td>
<td>2</td>
</tr>
<tr>
<td>1.3</td>
<td>Ordering</td>
<td>5</td>
</tr>
<tr>
<td>1.4</td>
<td>Viewing Groups</td>
<td>6</td>
</tr>
<tr>
<td>2.</td>
<td>EDITING GROUPS</td>
<td>7</td>
</tr>
<tr>
<td>3.</td>
<td>OTHER INFORMATION</td>
<td>9</td>
</tr>
<tr>
<td>3.1</td>
<td>System Requirements</td>
<td>9</td>
</tr>
</tbody>
</table>
1. **BASIC FEATURES**

1.1 **Enabling Macros**

Excel macros must be enabled to use this program. If the program is being opened for the first time, the macros may be disabled. This is the default for Excel on most computers.

To enable the macros:

1. If the following Security Warning appears at the top of the page, when the Excel file is first opened, click the “Enable Content” button.

![Security Warning](image1.png)

2. If the security warning did not appear, go to the “File” tab in the top left hand corner of the screen and in the Info tab on the left, look for a security warning. If there is no warning return to the “Home” tab.

![Excel Interface](image2.png)

3. If there is a warning, click the “Enable Content” button. A drop down list will be displayed.
4. Click the “Enable All Content” button.

1.2 Home Screen

Open Files

When the program is first opened, the only button that is displayed is the “Open Files” button. Select this and you will be prompted to select the folder that contains the data files. (Note: You are selecting the folder that contains the files not the files themselves)

Select this folder and click “OK”.
Warning Messages

If the selected folder does not contain the correct files an error message will show.

If this occurs, please ensure that the two files named “Sample Link Matrix.csv” and “Sample Staff List.csv” are in the selected folder.

Similar warning messages are shown if the files are not in the correct format. Sample files are provided to be used as templates when creating the data files.

If there are any appraisers in the Sample staff list that have been assigned to a Junior not in the same department, a warning message will be shown and you will be brought to the “Sample Staff List” worksheet.
These conflicts will be highlighted in red in the “Sample Staff List” sheet.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Name</td>
<td>Level</td>
<td>Service List Appraiser</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Analyst 2</td>
<td>Analyst</td>
<td>Analytics Senior Consultant 13 (Analytics)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Analyst 2</td>
<td>Analyst</td>
<td>Analytics Senior Consultant 14 (Analytics)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Analyst 2</td>
<td>Analyst</td>
<td>Analytics Manager 15 (Analytics)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Analyst 2</td>
<td>Analyst</td>
<td>Analytics Manager 15 (Analytics)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Analyst 2</td>
<td>Analyst</td>
<td>Analytics Manager 15 (Analytics)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Analyst 2</td>
<td>Analyst</td>
<td>Analytics Manager 15 (Analytics)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Analyst 2</td>
<td>Analyst</td>
<td>Analytics Manager 15 (Analytics)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Analyst 2</td>
<td>Analyst</td>
<td>Analytics Manager 15 (Analytics)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Consultan</td>
<td>Consultant</td>
<td>Analytics Manager 16 (Analytics)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Consultan</td>
<td>Consultant</td>
<td>Analytics Manager 17 (Analytics)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Senior Co</td>
<td>Senior Co</td>
<td>Analytics Manager 18 (Analytics)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Senior Co</td>
<td>Senior Co</td>
<td>Analytics Manager 18 (Analytics)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Manager 1</td>
<td>Manager</td>
<td>Analytics Manager 19 (Analytics)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>Manager 1</td>
<td>Manager</td>
<td>Analytics Senior Manager 2 (T)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Manager 1</td>
<td>Manager</td>
<td>Analytics Senior Manager 9 (Analytics)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>Manager 1</td>
<td>Manager</td>
<td>Analytics Senior Manager 9 (Analytics)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>Manager 1</td>
<td>Manager</td>
<td>Analytics Senior Manager 9 (Analytics)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>Senior M</td>
<td>Senior M</td>
<td>Analytics Director 8 (Analytics)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>Director 8</td>
<td>Director</td>
<td>Analytics Director 8 (Analytics)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>Analyst 2</td>
<td>Analyst</td>
<td>EA Manager 8 (EA)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>Analyst 2</td>
<td>Analyst</td>
<td>EA Manager 8 (EA)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>Analyst 2</td>
<td>Analyst</td>
<td>EA Manager 10 (Analytics)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>Analyst 2</td>
<td>Analyst</td>
<td>EA Manager 9 (EA)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>Analyst 2</td>
<td>Analyst</td>
<td>EA Manager 9 (EA)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>Analyst 2</td>
<td>Analyst</td>
<td>EA Manager 10 (EA)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Manually change these and select the “Update” button or return to the Home screen using the “Home” button. (Note: It is only recommended to change these entries but it is not a necessity)
1.3 Ordering

Before the program can perform any functions the data must be ordered. To do this:

1. Select the “View” button on the home screen
2. A full view of the data is given with data entries highlighted in black and appraisers highlighted in green.

3. Make sure this data is consistent with the data that was imported. Then click the “Order” button.

The data is ordered by department, then by employment level and then by employee. The departments are highlighted in yellow, employees of similar level are highlighted in orange, appraisers are highlighted in green and the remaining data is highlighted in black.
1.4 **Viewing Groups**

Different departments and levels can be viewed using the dropdown lists on the home screen.

Select what department or level you wish to view in more detail and click the “View” button.

A department and level can be selected at the same time.
2. **EDITING GROUPS**

Each level in every department can be split into a specified number of smaller groups. To do this:

1. Select a department and a level from the dropdown lists on the home screen.
2. Click the “View” button.
3. Click the “Edit” button. A pop up window will require you to enter the number of groups you wish to split the group into. (Note: This number must be greater than 1 and less than the number of rows or columns)

4. Click the “OK” button. The group will be split into the specified number. These sub groups are highlighted in orange.

![Excel dialog box](image)

Three buttons appear. Click the “Exit Edit” button at any time if you wish to undo all changes that have been made.

5. Click the “Sort” button. This will reorder the rows and columns of the table in an attempt to get the most amount of black and green squares in the orange groups.

![Table image](image)
6. The “Sort” button can be clicked until the groups are acceptable or if it stops making changes.

7. Manual edits can be made by changing the number in the group column (or row) to the group number you desire. (Note: It must be an existing group number)

8. When you are content with the groups click the export button. This will save the new groups into a new spreadsheet and remove them from the original spreadsheet. The new spreadsheet is named after the department and level you edited.

(Note: green squares represent appraisers. These must be in the orange groups)

You will be returned to the original spreadsheet. Any time you search, using the home screen dropdown lists, for the department and level you edited you will be brought to the new spreadsheet that contains the edits. A new spreadsheet is made for every edit you make.
3. OTHER INFORMATION

1.1 System Requirements

The system uses Visual Basic to manipulate Microsoft Excel Spreadsheets. This code is embedded in the Microsoft Excel. Therefore the system requirements are the same system requirements to run Microsoft Excel 2010:

- **Memory:** 512 megabytes of RAM or higher
- **Hard Disk:** 1 gigabyte available disk space
- **Computer Processor:** 500 megahertz processor or higher
- **Display:** 1024 × 576 resolution monitor or higher
- **Operating System:** Windows XP, Windows Vista, Windows 7