**Question 1**

**a.**

One of the potential benefits of using a spreadsheet solution for the Baby Green Collection is that spreadsheets are easy to use. Creating a spreadsheet and performing the standard statistical analysis can be done in minutes. Spreadsheets are also easy to understand by humans because of their tabular nature of data representation.

Spreadsheets have several potential disadvantages though that make them not suitable for storing larger amounts of data like the full dataset of Baby Green Collection. When spreadsheets get larger, they become harder to understand which leads to mistakes. Maintaining a spreadsheet is also more difficult when the document grows. Important other disadvantages of spreadsheets are the lack of data integrity (it’s harder to enforce that the data is well-formed), data redundancy (the same entries occur in different places in a single spreadsheet), performance (searching large spreadsheets requires a lot of CPU power) and multi-user issues/security (it’s hard to give certain users only access to a part of the spreadsheet).

Database solutions are initially harder to use than spreadsheets but are more suitable for storing larger amounts of data. Using a database management system (DBMS), it is possible to access all of the data using a single query. For example, you can get the total number of sold items from one product in December of last year with one query. Another advantage of using a DBMS over a collection of spreadsheets is that it is possible to run queries that reference different tables in the database. An example is a query that references the customers table and the orders table, to see what orders a single customer has placed in the past.

The possibility to maintain data integrity by defining constraints upon the stored data is a third advantage of using databases over spreadsheets. The Baby Green Collection can for example enforce that every order needs to have an order ID that is an integer of 10 characters. A fourth advantage of using a database over spreadsheets is the possibility to normalize the data, which makes sure that every piece of data exists only in one place in the database to minimize data redundancy. The chance of an error is smaller if for example a product id only needs to be changed in one place, instead of four places.

A database management system gives administrators control over the security of the data and access to the data. This can be used by the Baby Green Collection to limit the access to personal data of employees and customers to only users that need it (for example sales and client support) to comply with government regulations regarding privacy. The company can also protect its data by giving only certain users rights to add, delete or update data. Securing data and access control are not easy to implement using spreadsheets.

**b.**

An example of one document in the sales collection in a MongoDB database (including production data):

{ 'product\_code' : '0001',

'product\_category': 'T-shirt'

'description': 'Newborn Boy'

'age': '1m'

'size': 0,

'color': 'blue',

'production': {'estimated\_production\_2019': 900,

'quantity\_produced\_upto\_jan2019': 540,

'cost\_per\_unit': 3.76,

'total\_cost': 3384.20

'center\_of\_production': 'Bangladesh'

'main\_component': 'Cotton'},

'price': 8.74,

'quantity\_sold': 867 }

**c.**

**Staff**

staff\_id CHAR(4) (PK)

staff\_name VARCHAR(70)

staff\_role VARCHAR(30)

*Explanation and assumptions:*

* I chose staff\_id as the primary key, because each staff ID uniquely identifies a record in the Staff table.
* Because of the leading zero’s, I chose CHAR instead of INT as the type of the staff ID.
* I gave the staff name a maximum of 255 characters, to make sure that new employees with long names will not encounter any problems.
* I assumed based on the current staff roles that the name of a staff role never exceeds 30 characters based.

**Contacts**

staff\_id CHAR(4) (PK, FK)

customer\_id CHAR(4) (PK, FK)

contact\_date DATE (PK)

contact\_method VARCHAR(50)

contact\_details VARCHAR(50)

*Explanation and assumptions:*

* I added staff\_id and customer\_id (instead of the names) and made them foreign keys to connect this table to the Staff and Customer tables.
* I chose the combination of staff\_id, customer\_id, and contact\_date as the primary key because they together uniquely identify each record in the Contacts table. I assumed that if a customer has more than once contact with a staff member on the same day it will still be recorded in only one record.
* I assumed that the contact method and the contact details never exceed 50 characters.

**Customer**

customer\_id INT(6) (PK)

customer\_name VARCHAR(255)

customer\_street VARCHAR(70)

customer\_city VARCHAR(70)

customer\_postal\_code VARCHAR(8)

customer\_country CHAR(2)

customer\_freq\_of\_purchase\_feb\_may INT(3)

customer\_min\_purchase DOUBLE(3, 2)

customer\_max\_purchase DOUBLE(3, 2)

*Explanation and assumptions:*

* I chose customer\_id as the primary key because each customer ID uniquely identifies a record in the Customer table.
* I assumed that: a postal code is never more than 8 characters, a country code is always 2 characters, a customer name never exceeds 255 characters, and street and city both never exceed 70 characters. Also, I assumed that no customer buys more than 999 products worth in total £999.99.

**Production**

product\_code CHAR(5) (PK)

product\_category VARCHAR(20)

product\_description VARCHAR(20)

product\_age VARCHAR(4)

product\_size DOUBLE(2, 2)

product\_color VARCHAR(10)

product\_est\_production\_2019 INT(4)

product\_quantity\_produced\_upto\_jan2019 INT(4)

product\_cost\_per\_unit\_2019 DOUBLE(2,2)

product\_total\_cost DOUBLE(5,2)

product\_center\_of\_production VARCHAR(35)

product\_main\_composition VARCHAR(30)

*Explanation and assumptions:*

* I chose product\_code as the primary key, because each product\_code uniquely identifies a record in the Product table.
* I assumed that: the category and description never exceed 20 characters, the colour never exceeds 10 characters, and the age, estimated production, and the produced quantity never exceed 4 characters,
* The size 1 ½ must be added as 1.5 and size 1 ¾ as 1.75 to comply with the constraint of DOUBLE(2,2).
* I assumed also that the cost per unit never exceeds 99.99 and the total cost never exceeds 99999.99.

**Sales**

product\_code CHAR(5) (PK, FK)

product\_price DOUBLE (3, 2)

product\_quantity\_sold INT(4)

*Explanation and assumptions:*

* I removed all of the columns that are also in the Production table to minimize data redundancy.
* I chose product\_code as the primary key, because each product\_code uniquely identifies a record in the Sales table.
* I made product\_code also foreign key, to connect this table to the Production table.
* I assumed, based on the current numbers, that no price ever exceeds more than 999.99 and that no product will be sold more than 9999 times.

d.

Document databases have a flexible or no schema and therefore permit the storage of complex data that does not conform to a structured format like the images and letters that Baby Green receives. In contrast, relational databases have a rigid schema and do not support the storage of data without a structured format, which make them less suitable for Baby Green.

Because document databases have no rigid schema, they can easily deal with changes in the world outside the database like when database definitions need to be changed to encompass changes in the data structures being held. The updated schema can be applied when new documents are added, or when existing documents are updated. This is a benefit for Baby Green, because changes in the data structures, especially in the communication with suppliers, are likely going to happen. They need to be able to receive documents (e.g., Word, PDF) and images (e.g., JPEG, PNG) in different formats.

In relational databases, changes to a table’s schema are harder to implement. The database is unlikely to be available during this transition, which potentially means that Baby Green cannot record any sales during that time. Changing database definitions will also be expensive because every record in the table must be rewritten to comply with the changed requirements. This is therefore not desirable for a smaller company like Baby Green.

The lack of a rigid schema in document databases comes also with disadvantages for Baby Green, like the fact that these databases are not able to enforce data integrity by implementing constraints. Constraints can only be enforced by external applications. If there are several of those applications, each of them has to know about and enforce those rules without the support of the DBMS. This is not ideal and potentially leads to mistakes in the data, especially if the rules change.

Relational databases are able to enforce data integrity by implementing constraints. The DBMS will take steps to ensure that changes are in accordance with the rules, which leads to fewer mistakes in the data.

Both document databases and relational databases support database-using programs like database management systems (DBMS) to access and process the data. MongoDB is a popular DBMS for document databases, where PostgreSQL is a popular DBMS for relational databases.

Because of the ability of document databases to store complex data without a structured format, like the images and letters that Baby Green receives, and because they are able to adapt to changes in those data structures, a document database is the best option for Baby Green. Relational databases have the advantage that they can enforce data integrity by implementing constraints, but Baby Green can rely on an external application to have that.

**e.**

The data that I think is missing is actual customer orders. Currently, the data shows only the frequency of purchases and the minimum and maximum purchase of each customer, but not what products a customer bought and when. This can be done by creating an extra orders table where customer\_id and (a list of) product\_id’s can be linked to an order\_id and a date. The advantage of storing this information is that it is easier to help customers if there is something wrong with the product they bought. An employee of Baby Green could check in the database if and when the customer bought the product at Baby Green. The employee can then tell the customer for example if they still can send the product back for a refund.

Other data that is incomplete is the address info of the customers. Customers currently have no house number, which makes it very hard to ship products to their house. It will be a challenge to get that info, because there is also no contact info of customers, like an email address or a telephone number, recorded in the spreadsheets. Only the customers that had contact with one of the salespeople have an email address or telephone number in the contacts spreadsheet.

**Question 2**

**a.**

* The English Indices of Deprivation 2015 are measures of multiple deprivation at Lower-layer Super Output Area (LSOA) level.
* The indices are based on administrative data from tax year 2012/2013 and in some cases census data.
* Development of the Indices follows extensive exploration and quality assurance of data

sources and a methodology review.

* The index is based on seven domains: Income, Employment, Education, Health, Crime, Housing, and Living Environment.
* There are 32,844 LSOAs in England. The most deprived is ranked as 1, the least deprived as 32,844. The ranked LSOAs are divided into ten equal deciles.

*(100 words)*

**b.**

See notebook mk8978\_TMA02\_Question2b.ipynb.

**c.**

**The link between employment and education**

**Aims and objectives**

This study investigates the link between education and employment in England. It is based on one dataset that ranks the 32,844 Lower-layer Super Output Areas (LSOAs) in England by deprivation, where the LSOA (1500 residents) with a rank of 1 is the most deprived.

This study considers two questions: does deprivation of education correlate with the deprivation of employment and if so, is this correlation as strong in the cities as in the rural areas?

**Background**

Research in the past has been clear: more education leads to better prospects for employment (UK Government, 2014). This investigation uses the English Deprivation Indices to see if we can confirm this strong link between education and employment.

A downside of this dataset is that the choice of components and the weighting of those components that leads to the ranking is unavoidably subjective.

**Sources of data**

The deprivation data used in this report was originally obtained from the Gov.uk website (Department for Communities and Local Government, 2015). The original Excel-file was converted into a CSV file and underwent some minor cleaning before it was imported as a data frame into Jupyter Notebook using Python.

**Analysis pipeline**

I started looking on a national level for a correlation between education and employment by filtering in a data frame on the three most employment-deprived deciles. I then created a bar chart to see if most of those LSOAs also belonged to the three most education deprived deciles, which was the case.

Then I created a scatter plot of the unfiltered data frame, with the employment rank on the y-axis and the education rank on the x-axis. This graph also showed a correlation. I used Pearson’s R2 test to confirm this.

On the local level, I repeated the above steps to compare the city of Manchester with the more rural County Durham. Finally, I used Pearson’s test to compare the three cities Liverpool, Birmingham, and Westminster with the three rural areas Shropshire, Lancashire, and Cheshire.

**Findings**

The first charts I made to compare the education deprived deciles in the three most employment deprived deciles showed a clear correlation between education and employment.

Chart

Description automatically generated

*Figure 1: Most LSOAs that belong to the three most employment deprived deciles also belong to the three most education deprived deciles.*

Chart, scatter chart

Description automatically generated

*Figure 2: A scatter plot of the employment deprivation ranks against the education deprivation ranks shows a correlation.*

Pearson’s statistical test confirms the correlation with an r of 0.825, which is considered strong.

To see if this pattern is as strong in cities as in rural areas, I compared the city of Manchester with County Durham, a more rural area.

Chart, scatter chart

Description automatically generated

*Figure 3: A scatter plot of the employment deprivation ranks against the education deprivation ranks in County Durham.*

Chart, scatter chart

Description automatically generated

*Figure 4: A scatter plot of the employment deprivation ranks against the education deprivation ranks in Manchester.*

Both areas show a correlation, but it is stronger in County Durham. Pearson’s statistical test confirms this with an r of 0.906 for Durham and an r of 0.743 for Manchester.

When also comparing the correlation in cities Liverpool, Birmingham, and Westminster with the correlation in rural areas Shropshire, Lancashire, and Cheshire, we see that the average correlation of the rural areas was stronger (r = 0.875) than in the cities (r = 0.753).

**Conclusions**

The conclusion can be drawn that, based on this dataset, there is a strong correlation between education and employment. If this correlation is stronger in rural areas compared to cities cannot be concluded based on this small sample size. Further research is needed for that.

(600 words)

**References**

* Department for Communities and Local Government (2015) *The English indices of deprivation 2015*. Available at <https://www.gov.uk/government/statistics/english-indices-of-deprivation-2015> (Accessed 5 March 2021)
* Kleiweg, M. (2021), Notebook Data exploration TMA 02 Question 2b, submitted to The Open University as part of TM351 assessment
* UK Government (2014), Strengthening links between education and employment. Available at <https://www.gov.uk/government/speeches/strengthening-links-between-education-and-employment> (Accessed 5 March 2021)

**Question 3**

**a.**

i.

I choose the UK census report dataset as an additional dataset. The data is clear, complete, and fairly easy to understand. I can think of tons of interesting questions that can be answered using this dataset. Also, there is a clear way to connect this dataset with the deprivation index dataset using the output area lookup table.

I therefore will not use the GCSE results dataset. I do think it is an interesting dataset, but it is narrower in terms of who it includes (school results of young people) and I therefore cannot think of as many interesting questions.

ii.

A question that can be answered using this additional dataset:

What are the five Middle Layer Super Output Areas (MSOA) in England with the highest percentage of people between 16 and 24 years old who have never worked?

Note: this does not include full-time students, because they are their own category in this dataset (and are not considered

This dataset can answer this question, because it has for every MSOA the exact number of people under 24 years old who have never worked. The total number of people under 24 years old in each MSOA can be found by calculating the sum of each row. With that number, the percentage of the people that have never worked can be calculated. By comparing the percentages of all the different MSOAs, the five MSOAs with the highest percentage can be found.

iii

A question that can be answered using both datasets:

Is there a correlation between the percentage of people that have never worked and the employment deprivation ranking of the area where they live?

This question can be answered by combining the census dataset and the deprivation dataset. The percentage of people who never worked, regardless of age, can be found by combining the four CSV files. Then you can calculate the sum of each row to get the total people in each MSOA. Adding up the cells of each row that contain the number of people that never worked will give the number of people in each MSOA who never worked.

Both numbers can be used to calculate the percentage in each MSOA. Then this dataset can be combined with the deprivation index dataset, by matching the MSOA codes with the LSOA codes in the deprivation index. This can be done by using the provided output area classification table.

**b.**

*First question*

To answer the first question, I will start with opening the age 16-24 dataset in OpenRefine and do some data cleaning if needed. Then I import the data into a data frame in a Notebook using Python. I will use tools like head() and describe() to get a feel for the data.

To get a column with the total population of each MSOA in this age category, I can use df.sum(axis=1) in Python.

To get a column with the percentage of the people who never worked in each MSOA, I divide the ‘L14.1 Never worked’ column by the ‘total population’ column and then multiply by 100.

After that, I create a new data frame with only four columns: the MSOA name, the MSOA code, and the ‘percentage never worked’ column. Then I can sort the ‘percentage never worked’ column descending to see the five highest percentages at top of the column.

I can use that same data frame to visualise the data in a histogram or on a map with the help of matplotlib.

*Second question*

To answer the second question, I will again check all the CSV files in OpenRefine to see if there is any data cleaning needed. Then I will import the four provided CSV files of the census dataset

Into a data frame, which I will examine using head() and describe().

I can then use this data frame to get a column (using df.sum(axis=1)) with the total 16+ population of each MSOA.

Next, I combine the four columns with the ‘never worked’ totals for each age category in another data frame, together with the MSOA code column. I can use these columns to get a ‘total never worked’ column (using df.sum(axis=1)). After that, I use the ‘total population’ column and the ‘total never worked’ column to get a ‘percentage never worked’ column (see the first question).

Subsequently, I need to import the deprivation index CSV file into a data frame and examine it using head() and describe(). To combine this dataset with the data frame that contains the ‘percentage never worked’ column, the MSOA codes in the data frame need to be matched with the LSOA codes in the deprivation index data frame. I can do this by using the provided output area classification table and data mining tools.

Next, I can create a scatter plot of the employment deprivation rank against the never worked percentage to see if there is a correlation. Statistical tests like Pearson’s R2 test can be used to confirm the correlation and if confirmed, how strong it is. I can also visualise the ‘never worked’ percentage or the strength of the correlation of each MSOA on a map.

**c.**

See mk8978\_workplan.docx and mk8978\_project\_diary.ipynb.