
Appendix A. Data appendix

We set out the methods we use to infer additional information about users and about transactions in the MDB data. Where relevant, we point out the kind of measurement errors that could be induced and so that need to be borne in mind when interpreting the results.

Business accounts

The first key split we identify is between accounts used for businesses and those used for personal use by households. This distinction is motivated by a concern about artificially inflating the incomes of the self-employed by mistaking their revenues for profits and their costs for consumption. Moreover, identifying business accounts allows us to separately analyse the way that the self-employed as a group are impacted by the COVID shock.

We use several indicators to identify accounts used for business purposes.

- **Business expenses:** Accounts where more than 20% of debits are tagged as 'business expenses' over the course of a year.
- **Tax bunching:** The tax-minimising strategy for withdrawing profits from an owner-managed firm is to pay a salary equal to the primary threshold in the National Insurance system and then to make any further withdrawals as dividends. We identify accounts from which debits tagged as either salary, transfers, business expenses or 'no tag' are made that are within a pound of the (monthly) primary threshold.
- **Account name:** Bank customers can register an 'account name' with their bank. We use an MDB-supplied list of accounts with a name including the words 'company', 'business' or 'limited' (or derivations).
- **Business account type:** We use an MDB-supplied list of bank accounts that are explicitly registered as a business account with the bank.
- **HMRC VAT transaction:** We use an MDB-supplied list of accounts that include a transaction where the merchant is HMRC and the transaction description includes the word 'VAT'.

Any MDB user with at least one of these flags on any of their accounts is then designated a business account.

Children

We use two approaches to identify households with children. These can further help identify whether a partner's accounts have been linked into the MDB account.

Child benefit (number of children)

Child benefit payments can be identified using the 'benefits' tag and the statutory child benefit amount. These then give us the number of children for whom the user receives child benefit.

Around 90% of those listed as the registered claimant of child benefit are women.¹ We therefore take instances of male MDB users receiving child benefit as a strong indication that the accounts of a partner have been linked to the MDB account.

Child-associated spending

Another potential indicator of a user having children is spending on child-specific items. We identify a user as likely to have a child where *either* of the following criteria is satisfied:

- the user has three or more transactions in a given year that are tagged as one of the following: 'child clothes', 'child toys, clubs, or other', 'children other', 'children's club fees', 'toys' or 'child, every day or childcare'; or
- the user has at least *one* transaction in a given year tagged as either 'childcare fees', 'nursery fees' or 'school fees'.

If a user has child-specific spending but does *not* receive child benefit, that is suggestive of not linking in a partner's accounts (except for those who have gross earnings above £60,000, who have no incentive to claim child benefit since they have to pay it back with higher taxes). We exclude from our analysis users who earn under £60,000, have child-associated spending and do not receive child benefit. This should largely be users who have not linked in a partner's account, but will also include the separated partners of lone parents, who might still be spending on their children, not receive child benefit and yet have no 'missing' partner's account in the MDB app.

Number of jobs per user

Within a given month, a pair of a user's salaries are deemed to be part of the same salary 'stream' if they are:

- paid exactly 7, 14 or 28 days apart from one another, or the working day following if that date is a bank holiday; and
- not more than 20% different from one another in amount.

The intuition is that two salary transactions that are exactly one, two or four weeks away from one another and of a similar amount are likely to arise from the same job; and conversely that a pair of salaries that do not meet these conditions are likely to be from different jobs. We use this method to identify the number of jobs per user.

There will be some measurement error using this approach. For example, users who only have one job but have very irregular incomes may appear to have two (or more) salary streams; and couples who happen to get paid at the same time (perhaps because they

¹ <https://www.gov.uk/government/statistics/child-benefit-statistics-annual-release-2019>.

work for the same firm) and are in similarly paid roles may appear to only have one stream.

Untagged transfers

Many transfers from one account to another (for example, from a savings to a current account) are tagged as transfers. We identify additional transfers that are not tagged as such in the following way. Among transactions over £50, we search for transaction *pairs* satisfying *all* the following criteria:

- the two transactions are for the *exact* same amount;
- one transaction is a credit and the other a debit;
- the transactions are associated with the same user, but different accounts;
- the transactions occurred no more than two days apart; and
- each pair is unique: any transaction assigned to a pairing cannot be assigned to another.

We consider these pairs highly likely to be transfers. We do not include transactions of £50 or under as the tendency for transactions to bunch around round numbers increases the likelihood that we identify pairs of transactions that happen to meet the above conditions but in fact are not transfers. We restrict to transactions that are no more than two days apart based on an examination of tagged transfers.² This approach will miss transfers to or from an unlinked account (though note that these are still likely to be *tagged* as transfers).

Salaries

MDB has its own algorithm to tag credits as salaries. Unfortunately, this algorithm has an error from March 2020, causing too few salaries to be tagged since then (MDB is working on fixing this error in the near future). Thus, throughout this report, we use our own algorithm to identify salary payments. We have confirmed in correspondence with MDB that our algorithm is similar to MDB's; the key difference is that the descriptions of large credit transactions have been redacted in our data, so (unlike MDB) we cannot use that information. Our algorithm works as follows:

1. We start with untagged credits over 16 times the minimum wage that are paid into a current account, are not identified as a transfer by the transfer flag, and are not an exact multiple of £100. We gather together all of these transactions in our sample periods (for example, May 2019 to May 2020).

² One might have thought that we could tighten this criterion to the credit appearing no more than two days *later* than the debit, but not allowing for pairs where the credit appears first. However, based on an examination of tagged transfers, it appears that credits do sometimes appear before debits.

-
2. We place transactions in the same account into calendar 'cycles'. The simplest cycle is weekly: we identify all transactions that take place on Mondays, all that take place on Tuesdays, etc. We also examine cycles of four weeks, one month and 'last day-of-week of the month' (for example, last Friday of the month).
 - a. If a day in a cycle falls on a bank holiday or weekend, the cycle is adjusted to instead include transactions on the working day before and the working day after that bank holiday or weekend. We account for differences in bank holidays across UK nations.
 - b. Things are more complex in December, when many workers get paid early, though not all at the same time and not in a way obviously related to their standard cycle. Based on examining tagged salaries, we classify the following as 'Christmas pay days': all working days up to a week before Christmas, the last Friday that is at *least* a week before Christmas, and the first working day after Christmas. If a cycle includes a day after any of these 'Christmas pay days', we adjust the cycle to include transactions that take place on all earlier Christmas pay days. We do not do this for weekly cycles – as these seem to be less affected by Christmas – although we do adjust for Christmas bank holidays as described above.
 - c. We adjust each cycle other than weekly cycles to include the working day before and the working day after the expected day of the cycle (we refer to these as '+- 1 day' cycles. This reflects the fact that it appears (from examining tagged salaries) that sometimes workers get paid slightly early or late for reasons unrelated to bank holidays, weekends or December.
 3. Transactions can fall into multiple cycles – for example, a pair of transactions on two Tuesdays four weeks apart from one another will fall into the same *three* '4 weeks +- 1 day' cycles: 'every fourth Monday +- one day', 'every fourth Tuesday +- one day' and 'every fourth Wednesday +- one day'. They will also fall into the same weekly cycle: 'every Tuesday'. And they may fall into common 'month +- 1 day' cycles.
 - a. Within these cycles, we separate transactions falling in them into sets, where each transaction in a set is no more than 20% larger than the next-lowest transaction. This groups transactions that are of a similar size, which salary payments usually are. We delete sets that have fewer than four transactions in them. For weekly cycles, the threshold is 10%, reflecting the fact that weekly cycles cover many more days and so are more prone to pick up other transactions. We note that those with truly weekly cycles have many transactions for obvious reasons, which also means that, if their pay fluctuates, they can still have a set that spans a wide range of amounts.
 4. We now have a series of cycle-set combinations, and a given transaction may fall into multiple such combinations (because they can fall into multiple cycles, as described above).
 5. We go through these combinations and find, for each account, the one that *includes the most transactions*. These transactions are placed into 'group 1'. We then repeat the exercise with the remaining transactions, finding the cycle-set combination that includes the most of the remaining transactions. Those in that cycle-set combination

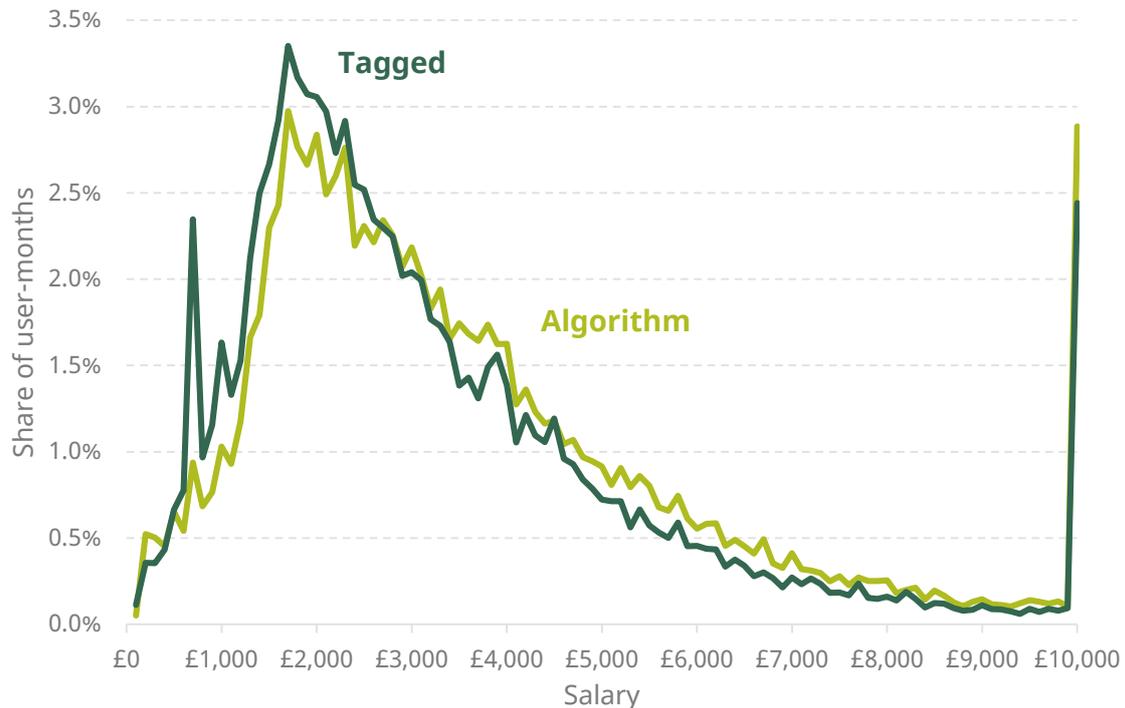
are placed into 'group 2', etc., until all transactions are assigned a group. Groups with fewer than four transactions are deleted.

6. This gives us a series of transactions that are regular in the sense that they appear with at least three other transactions over the sample period at a regular frequency and are of a regular amount.

To be clear, what this method does *not* do is:

- Track well users who change jobs or see a substantial change in earnings during the sample period (unless they get four payments in the new job/payment level in the sample period).
- Take any account of the transactions having a regular *gap*. For example, suppose a user gets five transactions all on the same day. Those transactions will all be put in the same cycles, and may end up in the same group. There is no check that, for example, transactions in a weekly cycle are all at least one week apart.
- Identify jobs with zero earnings – say those on zero hour contracts who did not get any work that month.

Figure A1. The distribution of actual tagged salaries and identified salaries using our algorithm



Note: The data shown are at the user-month level. The lines show the distribution of user-months with at least £100 of the respective kind of salary. Salaries are binned to the nearest £100. Those with a salary above £10,000 per month are grouped together.

Source: Authors' calculations using Money Dashboard data available 12 June 2020.

We **test our algorithm** by using it to classify transactions that include tagged salaries (over the period April 2019 to February 2020, when MDB's tagging algorithm was working). Specifically, we give the algorithm the tagged salaries and untagged transactions of all users with at least one tagged salary in our balanced panel, and ask it to classify which of these are salaries. Our algorithm identifies 89% of user-months as having some salary, compared with 83% of user-months that have some tagged salary. The total salary identified by the algorithm is the same as the total tagged salary 50% of the time. This rises to 58% for user-months that have both an algorithm-identified and a tagged salary.

On the intensive margin of salary amounts, the distribution of algorithm and tagged salaries is very close, as Figure A1 shows.