**Summary**

Small numbers of responders, so conclusions that can be drawn from the survey are limited. Perhaps repeat after a future RAP champions meetings?

There were a range of responders from different departments, most were not part of a peer review scheme but wanted to be with a focus very much around R. About half the responders had something that they thought would benefit from peer review.

Further detail split by about responders, current peer review, interest in receiving peer review, willingness to provide peer review.

**Notes on methodology**

Some of the questions around current coding experience and types of coding work presumed an R-type coding (functional paradigm, use of packages etc). This isn’t an issue as the majority of responders said they used R.

**About the responders (Q1 to Q6)**

20 responses from a range of government departments and non-departmental bodies. Distribution not remarkable: 3 responses from the DfE, 2 responses from 4 departments/bodies and 1 from the remaining 9 department/bodies. Most (15/20) work in a clearly analytical function in their organisation (stats/data science/analysis department).

In response to the Q3 around current coding skills, over half of the responders stated they use version control (higher level coding). Most (7/20) responders put themselves in the ‘Code in project with version control’ category or ‘Code in project’ (5/20).

¾ of responders would be happy to be contacted by email with further questions.

**Current experience of peer review (Q7 to Q11)**

Less than a quarter (4/20) responders are currently part of a peer review scheme.

**Interest in receiving peer review scheme (Q12 to Q21)**

All those who weren’t part of an existing peer review scheme thought it would be useful. 2 of those who were part of one thought it would be neither useful or not useful, presumably because they are already part of a scheme. Broadly, the same answers were given for the usefulness of such a scheme for responders’ team but more tentatively so (‘very useful’s downgraded to ‘useful’).

Removing those 2, about half (10/18) responders had code that would benefit from peer review (Q13). Slightly confusingly, one of the 2 ‘neither’ responders had code that they thought would benefit from peer review.

Only about a quarter of those with code ready to review (2/10 or 3/11 with the rogue ‘neither’) had any timescales around when it would be useful for review to happen. All but 1 of these 11 responders used R in these project, with smaller amounts of SQL (5/11), python (3/11) and HTML (1/11). No other languages or HMTL were used. The picture was very similar when responders were asked about project review in their teams, with all projects using R (12/12), some SQL (5/12) or Python (4/12) and a small but slightly increased on induvial amount of HTML (3/12). One responder mentioned SAS as a ‘possibility’.

Looking at the numbers of code for review by type (Q14 individual & Q19 team) there were some surprisingly high numbers by 3 responders in the ‘ad hoc’ and ‘coding project’ type categories. It may be that these responders gave the number of pieces of code work rather than the number that would be ready for and benefit from peer review. One responder had 30 individual and 8 team ad hoc pieces of coding work for review, and 10 individual coding projects. Another 2 responders had 10 ad hoc projects, for themselves or themselves and their team. Aside from these responses, the numbers were between 1 and 6, with 1 being the modal value, and fairly evenly across the categories (ad hoc, code in project, code in project with VC, project with VC, functional project with VC, package with VC). In total, there were 19 non-zero responses in the individual categories and 22 in the team categories.

**Willingness to peer review (Q22 to Q25)**

Good news is that most (17/20) responders would be happy to peer review other people’s code (Q22), and over half of them (10/17) confident to look at version controlled code, and over a quarter (6/17) could review functional code (Q23).

In terms of time commitment, about half (9/17) would give 0 to 4 hours a month and the remaining (8/17) could provide 5 to 9.

All except 1 could provide R coding review (16/17), about half SQL (9/17), about a quarter python (4/17) and small numbers for HTML (2/17) and Java (1/17). No other languages were suggested by responders.