

Supporting the self-employed during COVID-19

Joe Fallon & Gav Thompson | HMRC

EARL 2020

Self-Employment Income Support Scheme

The Challenge

- In response to the Covid-19 outbreak, the UK government launched a policy (SEISS – Self-Employment Income Support Scheme) aimed at supporting the self-employed.
- We were tasked with bringing together data from disparate systems to accurately determine the eligible population, and to model their income position, which in turn was used to determine their grant.

What did we do?

In 6 weeks developed a Reproducible Analytical Pipeline covering:

- rapid ingestion of customer data
- modelling of eligibility and grant
- generation of outputs to feed both the Contact Strategy but also the live Digital service.



Self-Employment Income Support Scheme

The Outcome

- HMRC were able to successfully launch the Self-Employment Income Support Scheme in a little over 6 weeks. This was built on the foundation of a robust and accurate database covering the entire eligible population.
- The scheme has paid out almost £7bn to the self-employed.

Feedback

- **Twitter:** “Best part of Monday so far, opening up my emails to find messages from self-employed clients confirming they've successfully claimed their SEISS and that I got my sums right when I estimated how much they'd get!”
- **A comment from a stakeholder:** Personally I have been hugely impressed by HMRC both on this project and more generally at how you have responded to the challenges of Covid-19. I'm guessing not many people realise the amount of work that goes into getting these schemes off the ground, but I think we all owe you a debt of gratitude. Thank you.



SEISS Directive

The profits condition

- 5.1 The profits condition is met if-
- (a) where the person is not subject to the loan charge, the person meets condition A, B or C, or
 - (b) where the person is subject to the loan charge, the person meets condition D or E.
- 5.2 Condition A is met if-
- (a) the person's trading profits of the tax year 2018-19 were £50,000 or less but were more than nil, and
 - (b) those profits are equal to or more than the person's relevant income in that tax year.
- 5.3 Condition B is met if-
- (a) the person carried on a trade in the tax years 2016-17, 2017-18 and 2018-19,
 - (b) the average amount of the person's trading profits of those tax years was £50,000 or less but was more than nil, and
 - (c) the sum of those profits is equal to or more than the sum of the person's relevant income for those tax years.
- 5.4 Condition C is met if-
- (a) the person carried on a trade in the tax years 2017-18 and 2018-19 but did not carry on a trade in the tax year 2016-17,
 - (b) the average amount of the person's trading profits of the tax years 2017-18 and 2018-19 was £50,000 or less but was more than nil, and
 - (c) the sum of those profits is equal to or more than the sum of the person's relevant income for those tax years.
- 5.5 Condition D is met if-
- (a) the person carried on a trade in the tax years 2016-17 and 2017-18,
 - (b) the average amount of the person's trading profits of those tax years was £50,000 or less but was more than nil, and

- (c) the sum of those profits is equal to or more than the sum of the person's relevant income for those tax years.

5.6 Condition E is met if-

- (a) the person did not carry on a trade in the tax year 2016-17,
- (b) the person's trading profits of the tax year 2017-18 were £50,000 or less but were more than nil, and
- (c) those profits are equal to or more than the person's relevant income for that tax year.

SEISS payment

6.1 The amount of the SEISS payment is the lower of-

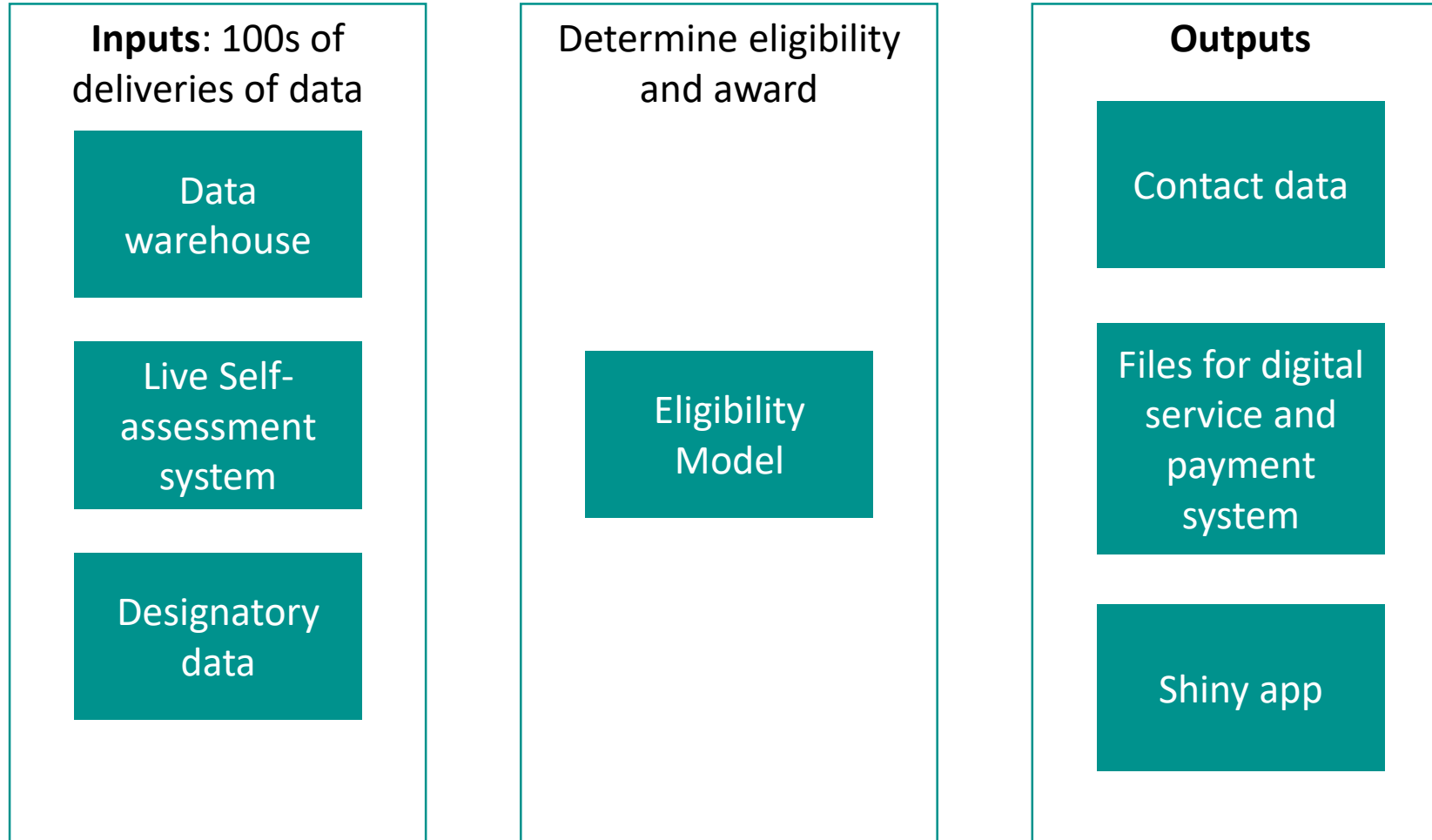
- (a) £7,500, and
- (b) $3 \times \left(\frac{TP}{12} \times 80\%\right)$.

6.2 In paragraph 6.1, TP is-

- (a) except where the person is subject to the loan charge, determined by the first to apply of the following paragraphs-
 - (i) if the person carried on a trade in the tax years 2016-17, 2017-18 and 2018-19, the average trading profits of those tax years,
 - (ii) if the person did not carry on a trade in the tax year 2016-17, the average trading profits of the tax years 2017-18 and 2018-19, and
 - (iii) if the person did not carry on a trade in the tax year 2017-18, the trading profits of the tax year 2018-19, or
- (b) where the person is subject to the loan charge, the average trading profits of the tax years 2016-17 and 2017-18 or, if the person did not carry on a trade in the tax year 2016-17, the trading profits of the tax year 2017-18.



Self-Employment Income Support Scheme (SEISS)



How did we cope?

Hundreds of
data deliveries

Developing model whilst
policy in development

Rapid turnaround
needed

What aspects of data science proved useful?

- Approaches
- R tips and tricks

Approaches

Data Lineage

Kept raw data, interim datasets, and output data.

Defined Roles

Each data scientist had well-defined responsibilities.

Version Control

Git was used for version control.
Made it easy to re-deploy to the cloud.

QA

Comparison between results calculated by tax professionals and the model after every major change.



Approaches

Pipelines

Series of processes starting from the raw data and ending with the output files.

Functions

Code split up into small logical chunks.
Separate scripts to run functions, from functions themselves.



Separating functions from scripts

```
230 #' Add address
231 #'
232 #' @param df Dataframe from sa_cid_name_address
233 #' @param db_path Path to seiss database
234 #' @param utr_col Column containing UTRs (needs to be numeric)
235 #'
236 #' @return Dataframe
237 add_address <- function(df, db_path, utr_col = "cwh_txp_utr"){
238
239   seiss_analysis = RSQLite::dbConnect(RSQLite::SQLite(), dbname = db_path, flags = RSQLite::SQLITE_RO)
240   on.exit(RSQLite::dbDisconnect(seiss_analysis))
241
242   df_address <- dplyr::tbl(seiss_analysis, "sa_cid_name_address") %>%
243     dplyr::collect()
244
245   df_out <- df %>%
246     dplyr::left_join(df_address, by = structure("cwh_txp_utr", names = utr_col)) %>%
247     add_rls_flag(db_path, utr_col) %>%
248     add_special_req_flag(db_path, utr_col) %>%
249     add_welsh_flag(db_path) %>%
250     add_overseas_flag(db_path, utr_col) %>%
251     add_no_address() %>%
252     format_address() %>%
253     format_name() %>%
254     add_nino(utr_col = utr_col) %>%
255     dplyr::select(NINO, !!rlang::sym(utr_col), full_name, name_line_1, name_line_2, starts_with("address"), everything())
256
257   return(df_out)
258 }
```



Separating functions from scripts

```
78 df <- readr::read_csv("data/other/seiss1_comms_invited.csv", col_names = "cwh_txp_utr")
79
80
81 df_contact <- df %>%
82   add_best_email(db_path) %>%
83   add_best_mobile_num(db_path) %>%
84   add_address(db_path)
85
```



Tips and tricks

RSQLite

R interface to the SQLite database engine

File in zip archive

```
readr::read_csv(unz("zip_archive.zip", "file.csv"))
```

Finding data about a given customer

Command line tools such as **grep**, **zipgrep**, **awk**, **sed** *etc.*

Tips and tricks

Scoped versions of mutate

```
1 library(dplyr)
2
3 # Convert all columns to character
4 iris %>%
5   dplyr::mutate(dplyr::across(dplyr::everything(), as.character))
6
7 # Multiple columns whose names start "Petal" by 10
8 iris %>%
9   dplyr::mutate(dplyr::across(dplyr::starts_with("Petal"), ~ .*10))
10
11 # Multiple all numeric columns by 100
12 iris %>%
13   dplyr::mutate(dplyr::across(where(is.numeric), ~ .*100))|
```



Tips and tricks

anti_join, semi_join

```
1 library(dplyr)
2
3 df_utrs_to_remove
4 df_utrs_to_keep
5
6 df %>%
7   dplyr::anti_join(df_utrs_to_remove, by = "UTR")
8
9 df %>%
10  dplyr::semi_join(df_utrs_to_keep, by = "UTR")
11
```

Q&A...

