

# {nottshcData}

Development of tools for querying, tidying, and analysing data in R



## Lori Edwards Suárez

Clinical Analyst 🦄

Background: NHS Expert

Fastest Mario Kart Driver in the Team

*RiO Expert, Data Analysis*



## Oluwasegun Michael Apejaye

Data Science Intern 🤖

Background: Energy and environmental management

Enjoys working with health data

*Forecasting, text mining, AI*

## Chris Beeley

Senior Data Scientist 🧙

Background: Clinical Psychology

Owns two cats named Pavlov and Freud

*Shiny, Server Hosting*



## Zoë Turner

Data Scientist 🧐

Background: Philosophy

Knows too much about data

*SQL and data engineering*



## Andreas Soteriades

Data Scientist 🧠

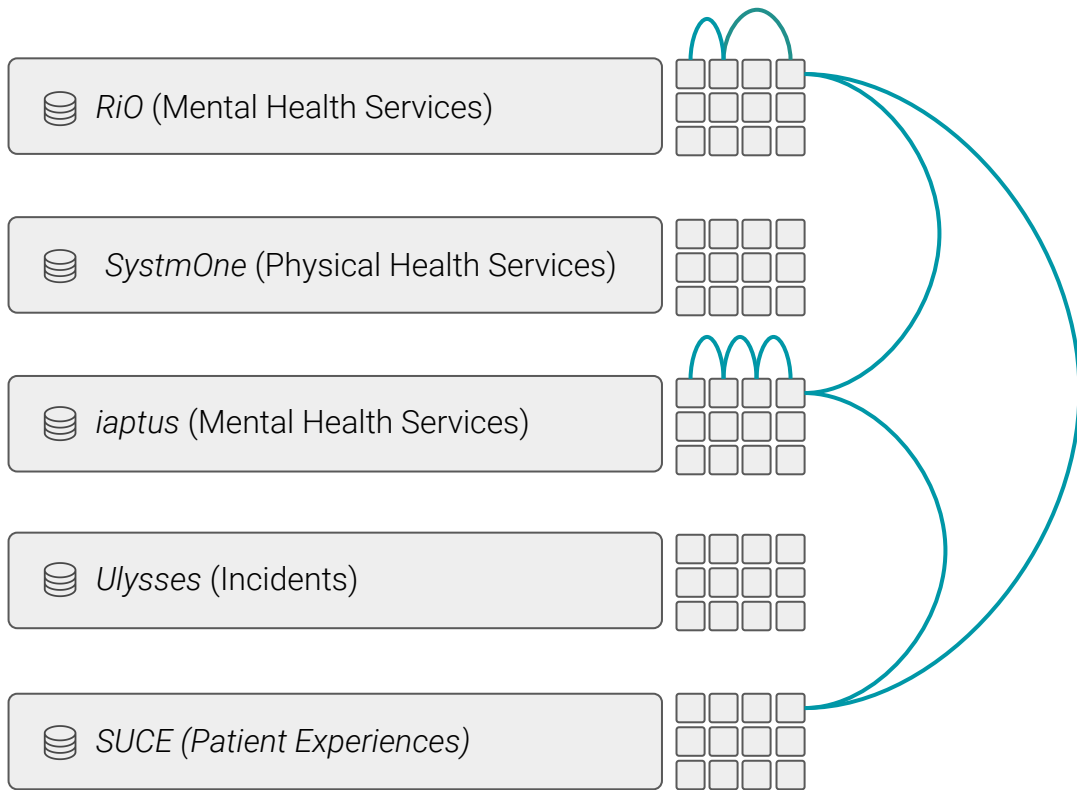
PAST MEMBERS

## Milan Wiedemann

Data Scientist 🐛



# THE CHALLENGE



GET DATA

TIDY DATA

ANALYSE / VIZ DATA

DOCUMENT CODE

TEST CODE



Get data

Tidy data

Analyse data

Visualise data

Communicate data

**{nottshcData}**



*RiO* (Mental Health Services)



*iaptus* (Mental Health Services)



*SystemOne* (Physical Health Services)



*Ulysses* (Incidents)



*SUCE* (Patient Experiences)

[{honos}](#)



Package for visualising and analysing the HoNOS (Health of the Nation Outcome Scales)

[{LSOApop}](#)



Package with Lower layer Super Output Area (LSOA) population estimates and Indices of Multiple Deprivation (IMD) for England

[{nottshcTeams}](#)



Helper funs to link clinical teams across different databases

[{nottshcMethods}](#)



Helper funs for data manipulations, calculations, and visualisations

[{pxttextmining}](#)

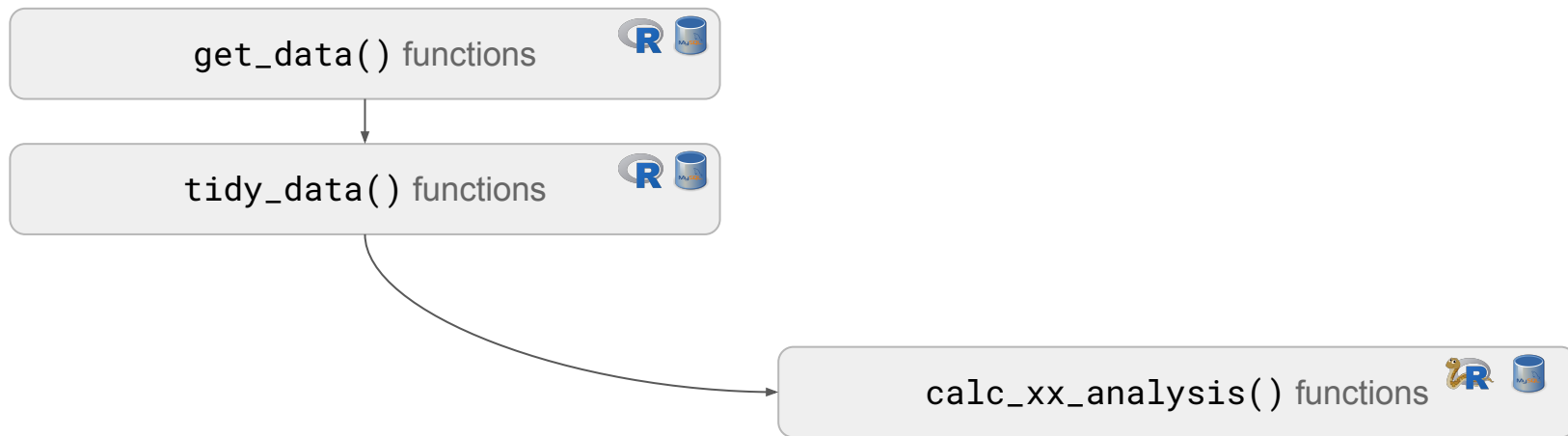
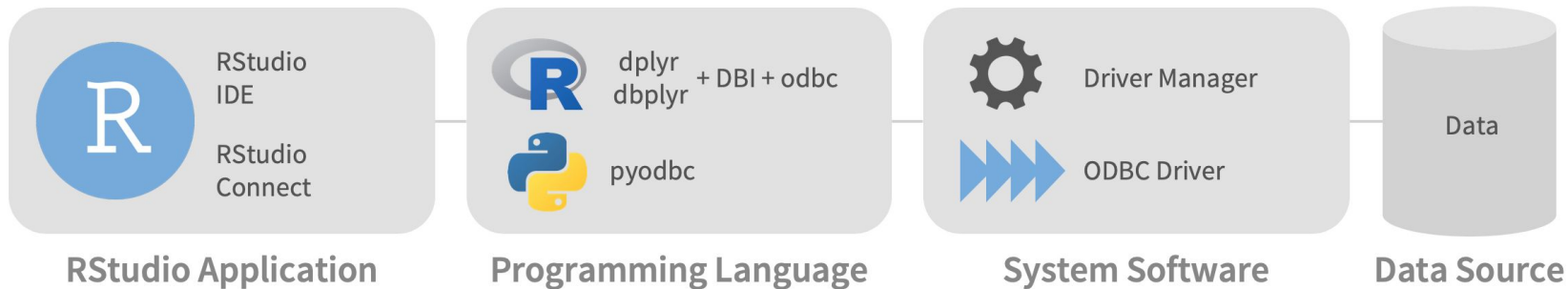


Package for text classification of NHS patient experiences

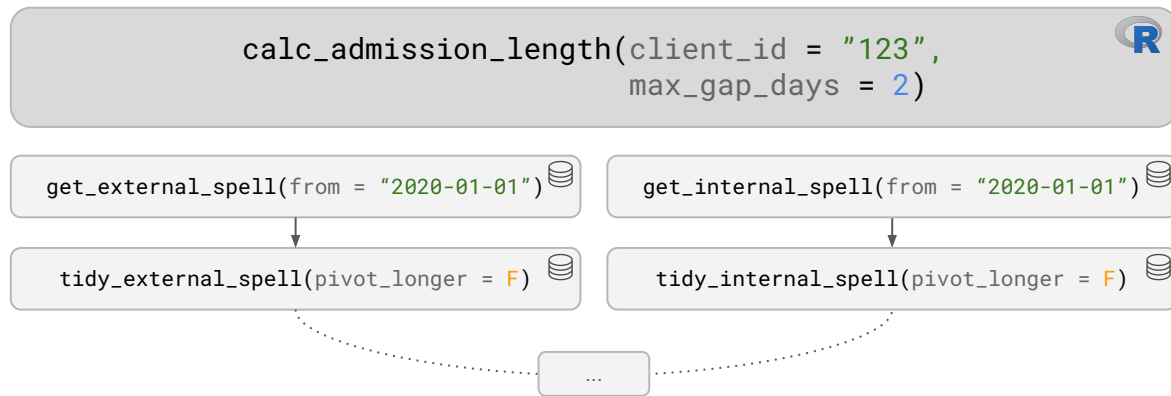
[{outcomesdashboard}](#)



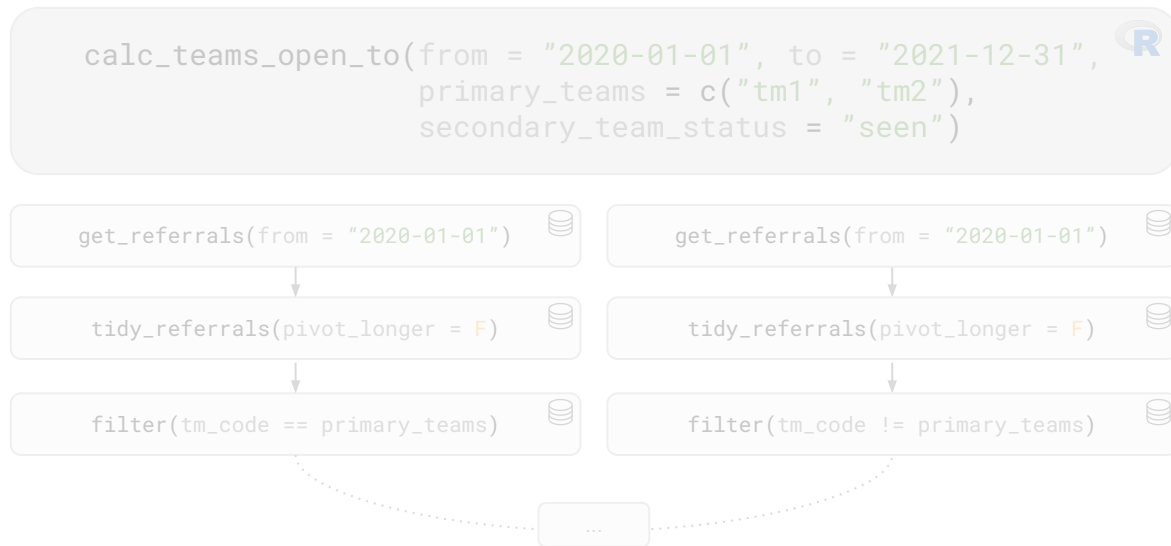
Interactive dashboard presenting results to a selected group of clinicians, managers, or the public.







What is the  
'total admission length'  
(internal and external)  
of a patient?



Which other teams  
are patients 'open to'  
while they are seen by  
one or more teams of  
interest?

```
calc_admission_length(client_id = "123",  
                      max_gap_days = 2)
```



```
get_external_spell(from = "2020-01-01")
```

```
get_internal_spell(from = "2020-01-01")
```

```
tidy_external_spell(pivot_longer = F)
```

```
tidy_internal_spell(pivot_longer = F)
```

...

```
calc_teams_open_to(from = "2020-01-01", to = "2021-12-31",  
                  primary_teams = c("tm1", "tm2"),  
                  secondary_team_status = "seen")
```



```
get_referrals(from = "2020-01-01")
```

```
get_referrals(from = "2020-01-01")
```

```
tidy_referrals(pivot_longer = F)
```

```
tidy_referrals(pivot_longer = F)
```

```
filter(tm_code == primary_teams)
```

```
filter(tm_code != primary_teams)
```

...

What is the  
'total admission length'  
(internal and external)  
of a patient?

Which other teams  
are patients 'open to'  
while they are seen by  
one or more teams of  
interest?

AUTOMATION OF RECURRING TASKS

EASY TO DISTRIBUTE ACROSS TEAM

CONSISTENT WORKFLOW ACROSS ANALYSTS

MODULAR APPROACH

DISAGGREGATION of DATA vs ANALYSIS

# BLOG POST

<https://cdu-data-science-team.github.io/team-blog/posts/2021-08-06-nottshcverse/>

```
calc_teams_open_to(from = "2020-01-01", to = "2021-12-31",
  primary_teams = c("tm1", "tm2"),
  secondary_team_status = "seen")
```

Which other teams  
are patients *'under  
the care of'* while they  
are seen by one or  
more teams of  
interest?

```
get_referrals(from = "2020-01-01")
```

```
tidy_referrals(pivot_longer = F)
```

```
filter(tm_code == primary_teams)
```

```
get_referrals(from = "2020-01-01")
```

```
tidy_referrals(pivot_longer = F)
```

```
filter(tm_code != primary_teams)
```

> Run on active connection | = Select block

```
1 SELECT *
2 FROM (SELECT COALESCE("LHS"."client_id", "RHS"."client_id") AS "client_id", "primary_referral_id", "primary_referral_datetime", "primary_team_code", "primary_team_desc",
3 FROM (SELECT "client_id", "primary_referral_id", "primary_referral_datetime", "primary_team_code", "primary_team_desc"
4 FROM (SELECT "client_id", "referral_id" AS "primary_referral_id", "referrals_referral_datetime" AS "primary_referral_datetime", "team_code" AS "primary_team_code", "team
5 FROM (SELECT "client_id", "referral_id", "referrals_referral_datetime", "team_code", "team_desc", "primary_referral", NULLIF("primary_referral_count", 0.0) AS "primary_r
6 FROM (SELECT "client_id", "referrals_referral_datetime", "team_code", "team_desc", "primary_referral", SUM("primary_referral") OVER (PARTITION BY "client_
7 FROM (SELECT "client_id", "referral_id", "referrals_referral_datetime", "team_code", "team_desc", COALESCE("primary_referral", 0.0) AS "primary_referral"
8 FROM (SELECT "client_id", "referral_id", "referrals_referral_datetime", "team_code", "team_desc", CASE
9 WHEN ("team_code" IN ('tm200')) THEN (1.0)
10 END AS "primary_referral"
11 FROM (SELECT "ClientID" AS "client_id", "ReferralNumber" AS "referral_id", "ReferralDateTime" AS "referrals_referral_datetime", "TeamReferredTo" AS "team_code", "TeamRef
12 FROM (SELECT *
13 FROM "REPORT_BASE"."Referrals"
14 WHERE ("ReferralDateTime" >= '2021-01-01')) "q01"
15 WHERE ("ReferralDateTime" <= '2021-06-01')) "q02") "q03") "q04") "q05") "q06"
16 WHERE (NOT(((("primary_referral_count") IS NULL)))) "q07") "LHS"
17 FULL JOIN [(SELECT "client_id", "referral_id", "referral_datetime", "firstappt_datetime", COALESCE("discharge_datetime", '2021-08-09') AS "discharge_datetime", "team_code
18 FROM (SELECT "client_id", "referral_id", "referrals_referral_datetime" AS "referral_datetime", "referrals_firstappt_datetime" AS "firstappt_datetime", "referrals_dischar
19 FROM (SELECT "ClientID" AS "client_id", "ReferralRowID" AS "referral_row_id", "ReferralNumber" AS "referral_id", "AppointmentSequenceID" AS "sequence_id", "AppointmentCont
20 FROM "REPORT_BASE"."Referrals") "q01"
21 WHERE (NOT("team_code" IN ('tm200')))) "q02") "RHS"
22 ON ("LHS"."client_id" = "RHS"."client_id")
23 ) "q03"
24 WHERE ("primary_referral_datetime" BETWEEN "referral_datetime" AND "discharge_datetime")
25 ORDER BY "client_id", "primary_team_code"
```

```
calc_teams_open_to(from = "2020-01-01", to = "2021-12-31",
  primary_teams = c("tm1", "tm2"),
  secondary_team_status = "seen")
```

Which other teams are patients 'open to' while they are seen by one or more teams of interest?

```
get_referrals(from = "2020-01-01")
```

```
tidy_referrals(pivot_longer = F)
```

```
filter(tm_code == primary_teams)
```

```
get_referrals(from = "2020-01-01")
```

```
tidy_referrals(pivot_longer = F)
```

```
filter(tm_code != primary_teams)
```

> Run on active connection | = Select block

```
1 SELECT *
2 FROM (SELECT COALESCE("LHS"."client_id", "RHS"."client_id") AS "client_id", "primary_referral_id", "primary_referral_datetime", "primary_team_code", "primary_team_desc",
3 FROM (SELECT "client_id", "primary_referral_id", "primary_referral_datetime", "primary_team_code", "primary_team_desc"
4 FROM (SELECT "client_id", "referral_id" AS "primary_referral_id", "referrals_referral_datetime" AS "primary_referral_datetime", "team_code" AS "primary_team_code", "team
5 FROM (SELECT "client_id", "referral_id", "referrals_referral_datetime", "team_code", "team_desc", "primary_referral", NULLIF("primary_referral_count", 0.0) AS "primary_r
6 FROM (SELECT "client_id", "referrals_referral_datetime", "team_code", "team_desc", "primary_referral", SUM("primary_referral") OVER (PARTITION BY "client
7 FROM (SELECT "client_id", "referral_id", "referrals_referral_datetime", "team_code", "team_desc", COALESCE("primary_referral", 0.0) AS "primary_referral"
8 FROM (SELECT "client_id", "referral_id", "referrals_referral_datetime", "team_code", "team_desc", CASE
9 WHEN ("team_code" IN ('tm200')) THEN (1.0)
10 END AS "primary_referral"
11 FROM (SELECT "ClientID" AS "client_id", "ReferralNumber" AS "referral_id", "ReferralDateTime" AS "referrals_referral_datetime", "TeamReferredTo" AS "team_code", "TeamRef
12 FROM (SELECT *
13 FROM "REPORT_BASE"."Referrals"
14 WHERE ("ReferralDateTime" >= '2021-01-01')) "q01"
15 WHERE ("ReferralDateTime" <= '2021-06-01')) "q02") "q03") "q04") "q05") "q06"
16 WHERE (NOT(((("primary_referral_count") IS NULL)))) "q07") "LHS"
17 FULL JOIN [(SELECT "client_id", "referral_id", "referral_datetime", "firstappt_datetime", COALESCE("discharge_datetime", '2021-08-09') AS "discharge_datetime", "team_code
18 FROM (SELECT "client_id", "referral_id", "referrals_referral_datetime" AS "referral_datetime", "referrals_firstappt_datetime" AS "firstappt_datetime", "referrals_dischar
19 FROM (SELECT "ClientID" AS "client_id", "ReferralRowID" AS "referral_row_id", "ReferralNumber" AS "referral_id", "AppointmentSequenceID" AS "sequence_id", "AppointmentCont
20 FROM "REPORT_BASE"."Referrals") "q01"
21 WHERE (NOT("team_code" IN ('tm200')))) "q02") "RHS"
22 ON ("LHS"."client_id" = "RHS"."client_id")
23 ) "q03"
24 WHERE ("primary_referral_datetime" BETWEEN "referral_datetime" AND "discharge_datetime")
25 ORDER BY "client_id", "primary_team_code"
```



## Overview

dbplyr is the database backend for [dplyr](#). It allows you to use remote database tables as if they are in-memory data frames by automatically converting dplyr code into SQL.

To learn more about why you might use dbplyr instead of writing SQL, see

`vignette("sql")`. To learn more about the details of the SQL translation, see

`vignette("translation-verb")` and `vignette("translation-function")`.