www.advance-vaccines.eu

Accelerated Development of VAccine beNefit-risk Collaboration in Europe

Grant Agreement n°115557

D5.7 Proof-of-Concept studies Phase 1: Results on near real-time monitoring

WP5 – Proof-of-concept studies of a framework

to perform vaccine benefit-risk monitoring

V1.8
[Final]

Date: xx April 2019
Nature: Report
Dissemination level: PU

Disclaimer: *The results described in this report are from the proof of concept studies conducted as part of the IMI ADVANCE project with the aim of testing the methodological aspects of the design, conduct and reporting of studies for vaccine benefit-risk monitoring activities. The results presented herein relate solely to the testing of these methodologies and are not intended to inform regulatory or clinical decisions on the benefits and risks of the exposures under investigation. This warning should accompany any use of the results from these studies and they should be used accordingly.*

TABLE OF CONTENTS

TABLES

FIGURES

DOCUMENT INFORMATION

Grant Agreement Number	115557	**Acronym**	ADVANCE
Full title	Accelerated Development of VAccine beNefit-risk Collaboration in Europe		
Project URL	http://www.advance-vaccines.eu		
IMI Project officer	Tek-Ang Lim (Tek-Ang.Lim@imi.europa.eu)		

Deliverable	**Number**	5.7	**Title**	Work Package 5 Near real-time monitoring study
Work package	**Number**	5	**Title**	Proof-of-Concept studies Phase 1: Results on near real-time monitoring

Delivery date	**Contractual**	Month 66	**Actual**	xx
Status	Current version / V1.8		Draft ☐ Final x	
Nature	Report x Prototype ☐ Other ☐			
Dissemination Level	Public x Confidential ☐			

Authors (Partner)	**Name (Partner)** Kaatje Bollaerts (P95), Tom de Smedt (P95), Chris Mcgee (Surrey) Hanne-Dorthe Emborg (SSI), Marco Villa (ASCLR), Maria Alexandridou (P95) Talita Duarte-Salles (IDIAP), Rosa Gini (ARS), Claudia Bartolini (ARS), Simon de Lusignan (Surrey), Myint TinTin Htar (Pfizer), Lina Titievsky (Pfizer) Miriam Sturkenboom (P95), Vincent Bauchau (GSK)			
Responsible Author	Kaatje Bollaerts		**Email**	Kaatje.bollaerts@p-95.com
	Partner	P95	**Phone**	

Description of the deliverable	Proof-of-concept study to test the feasibility of near real-time monitoring using European electronic health record databases
Key words	vaccine, benefit-risk, near real-time monitoring

DOCUMENT HISTORY

PROTOCOL (see EU PAS Register - 26809)

NAME	DATE	V	DESCRIPTION
Vincent Bauchau, Kaat Bollaerts, Miriam Sturkenboom	7-01-2017		This document is based on the POC1.2 outline, final version
Kaatje Bollaerts, Vincent Bauchau, Lina Titievsky, Miriam Sturkenboom	4-08-2017		The POC outline was further worked out in the POC1.2 synopsis, final version
Kaatje Bollaerts	21-09-2017	0.1	First draft protocol
Vincent Bauchau, Lina Titievsky, Daniel Weibel, Hanne-Dorthe Emborg	23-10-2017		Comments
Miriam Sturkenboom	30-10-2017		Comments
Kaatje Bollaerts	02-11-2017	0.2	Second draft
Hanne-Dorthe Emborg	8-11-2017		Comments
Miriam Sturkenboom	30-11-2017		Comments, Databases added
Kaatje Bollaerts	6-12-2017	0.3	Addressing comments
TinTin Htar, Vincent Bauchau, Hanne-Dorthe Emborg, Daniel Weibel	8-12-2017		Review, providing requested input
Kaatje Bollaerts	13-12-2017	1.0	Finalization
Patrick Mahy, Xavier Kurz, Jorgen Bauwens, Lieke van der Aa, Lina Titievsky, Linda Levesque, Sonja Banga	5-01-2018		SC review
Rosa Gini, Miriam Sturkenboom, Vincent Bauchau, Lina Titievsky, Kaatje Bollaerts	16-1-2018		Specifications success measures and ENCePP compliance

STATISTICAL ANALYSIS PLAN

NAME	DATE	V	DESCRIPTION
Kaat Bollaerts, Tom De Smedt, Lina Titievsky, Miriam Sturkenboom, Vincent Bauchau	31-07-2017		The SAP is based on the data specification plan v0.4,
Kaatje Bollaerts	14-02-2018	0.1	First draft SAP
Vincent Bauchau	16-02-2018		Comments
Miriam Sturkenboom, Vincent Bauchau, Daniel Weibel, Hanne-Dorthe Emborg, Tom De Smedt, Kaatje Bollaerts, Lieke van der Aa	21-02-2018		Discussion during TC
Kaatje Bollaerts	28-02-2018	0.2	Updating SAP
Vincent Bauchau	01-03-2018		Comments
Kaatje Bollaerts	04-03-2018	0.3	Updating SAP
Miriam Sturkenboom	05-03-2018		Comments
Kaatje Bollaerts	08-03-2018	0.4	Updating SAP
Miriam Sturkenboom	12-03-2018		Comments
Tom De Smedt	13-03-2018	0.5	Updating SAP to local creation of data tables
Kaatje Bollaerts	15-03-2018		Review
Tom De Smedt	15-03-2018	0.6	Updating following feedback
Kaatje Bollaerts	17-03-2018	V1.0	Final

REPORT

NAME	DATE	V	DESCRIPTION
Kaatje Bollaerts (P95)	14.12.18	0.1	First draft
Miriam Sturkenboom (P95), Vincent Bauchau (GSK)			comments
Kaatje Bollaerts (P95)	17.01.19	1.0	Second draft
Simon de Lusignan (Surrey), Miriam Sturkenboom (P95), Rosa Gini (ARS), Hanne-Dorthe Emborg (SSI), Tin Tin Htar (Pfizer)			Comments
Hanne-dorthe Emborg (SSI), Chris Mcgee (Surrey), Marco Villa (ASCLR)	14.03.19	0.1	Input databases
Kaatje Bollaerts			
Margaret Haugh (Pfizer)	22.03.19	0.2	Editing
Kaatje Bollaerts (P95)	20.04.19		Final edits made after ADVANCE Steering Committee review

DEFINITIONS

Abbreviations used in the report

- **ADVANCE** Accelerated Development of VAccine beNefit risk Collaboration in Europe
- **aP** acellular pertussis
- **B/R** benefit risk
- **CDC** Center of Disease Control
- **CDM** common data model
- **CIs** confidence intervals
- **eHR** electronic health record
- **EU** Europe
- **HHE** hypo-responsive episodes
- **IMI** Innovative Medicines Initiative
- **NRT** Near real time
- **POC** proof of concept
- **UK** United Kingdom
- **VAC4EU** Vaccine monitoring Collaboration for Europe
- **VCD** Vaccine Safety Datalink
- **wP** whole cell pertussis

Participants of the ADVANCE Consortium are referred to herein according to the following codes:

- **ARS** Agenzia Regionale di Sanità
- **ATSVP** Agenzia di Tutela della Salute Val Padana
- **AUH** Aarhus Universitets Hospital
- **BIFAP** Base de Datos parala Investigación Farmacoepidemiológica en Atencion Primaria
- **RCGP RSC** Royal College of General Practitioners Research and Surveillance Centre
- **SIDIAP** Information System for Research in Primary Care
- **SSI** Statens Serum Institut
- **THIN** The Health Improvement Network
- **EMC**. Erasmus Universitair Medisch Centrum Rotterdam (Netherlands)
- **UNIBAS**. Universitaet Basel (Switzerland) - **Managing entity of the IMI JU funding**
- **EMA.** European Medicines Agency (United Kingdom)
- **ECDC.** European Centre for Disease Prevention and Control (Sweden)
- **SURREY.** The University of Surrey (United Kingdom)
- **P95.** P95 (Belgium)
- **SYNAPSE.** Synapse Research Management Partners, S.L. (Spain)
- **OU.** The Open University (United Kingdom)
- **LSHTM.** London School of Hygiene and Tropical Medicine (United Kingdom)
- **PEDIANET.** Società Servizi Telematici SRL (Italy)
- **KI.** Karolinska Institutet (Sweden)
- **ASLCR.** Azienda Sanitaria Locale della Provincia di Cremona (Italy)
- **AEMPS.** Agencia Española de Medicamentos y Productos Sanitarios (Spain)
- **AUH.** Aarhus Universitetshospital (Denmark)
- **UTA.** Tampereen Yliopisto (Finland)
- **WIV-ISP.** Institut Scientifique de Santé Publique (Belgium)
- **MHRA.** Medicines and Healthcare products Regulatory Agency (United Kingdom)
- **SSI.** Statens Serum Institut (Denmark)
- **RCGP.** Royal College of General Practitioners (United Kingdom)
- **RIVM.** Rijksinstituut voor Volksgezondheid en Milieu * National Institute for Public Health and the Environment (Netherlands)

- **GSK.** GlaxoSmithKline Biologicals, S.A. (Belgium) – EFPIA Coordinator
- **SP.** Sanofi Pasteur (France)
- **NOVARTIS.** Novartis Pharma AG (Switzerland)
- **SP MSD.** Sanofi Pasteur MSD (France)
- **CRX**. Crucell Holland BV (Netherlands)
- **PFIZER**. Pfizer Limited (United Kingdom)
- **TAKEDA**. Takeda Pharmaceuticals International GmbH (Switzerland)

1. ABSTRACT

Background

The Accelerated Development of VAccine beNefit-risk Collaboration in Europe (ADVANCE) is a public-private partnership aiming to develop and test a system for rapid benefit-risk (B/R) monitoring of vaccines using European electronic health record (eHR) databases. This proof-of-concept study aimed to test the feasibility of near real-time (NRT) monitoring of vaccination coverage, benefits and risks based on multiple European eHR databases, using acellular pertussis vaccination in children aged <6 years as test case.

Methods

A qualitative feasibility assessment on NRT monitoring was carried out using a survey and face-to-face discussion with ADVANCE data partners. Subsequently, a dynamic cohort study was conducted containing two distinct observation periods: a first period to establish a baseline (Jan 2014 to Mar 2018) and a subsequent 3-month period to test the actual feasibility of weekly NRT monitoring. Data collection delays were calculated and an interactive web-application facilitating the visual monitoring of vaccination coverage, benefits and risks was developed.

Results

Nine databases from four countries (Denmark: SSI and AUH; Italy: ARS, ATSVP and PEDIANET; Spain: BIFAP and SIDIAP; UK: THIN and RCGP RSC) participated in the qualitative feasibility assessment. Five databases (SSI, ARS, ATSVP, SIDIAP, RCGP RSC) provided baseline data for the cohort study. Three databases (SSI, ATSVP and RCGP RSC) also participated in the NRT monitoring, providing data extractions on an almost weekly basis for 374,161 (SSI), 2,492 (ATSVP) and 6,362 (RCGP RSC) children. For vaccination events, the median data latency (time between event date and data release date) were 2 (SSI), 1 (ATSVP) and 2 (RCGP RSC) weeks. For the benefit and risk events, the median latencies were 2 (SSI), 16 (ATSVP) and 1 (RCGP RSC) weeks.

Conclusion

Several European eHR databases successfully demonstrated the feasibility of providing data for weekly NRT monitoring, with short data latencies during three months.

2. INTRODUCTION

Post-marketing near real-time (NRT) monitoring is used to detect vaccine safety signals as some adverse events might have gone undetected or unconfirmed during pre-licensure clinical trials as they are rare and/or have a delayed onset or because they are restricted to specific subpopulations not considered in the trials (1). Such NRT monitoring is usually implemented shortly after a new vaccine is released, when switching vaccine brand or expanding the target population. Apart from monitoring safety, the post-marketing surveillance of vaccination coverage and the benefits of vaccination is often undertaken to understand the uptake/vaccine utilization and impact on the vaccine preventable disease.

For vaccine monitoring purposes, electronic health record (eHR) databases are increasingly used as they allow to study real-world effects, including rare events, on a representative and wide geographical scale without the need to initiate prospective data collections. The Vaccine Safety Datalink (VSD) from the US Center of Disease Control (CDC) is a pioneering example of large scale vaccine monitoring focused primarily on safety (2). In Europe, most of the monitoring of vaccine coverage, benefit and risk is done nationally (ref paper 1), with the few explorations of NRT vaccine safety monitoring being based on a single national database (3, 4), or through enhanced surveillance by collecting additional data from patients (5).

To improve vaccine monitoring on a European scale, the Innovative Medicines Initiative (IMI) funded the Accelerated Development of VAccine beNefit-risk Collaboration in Europe (ADVANCE) project. ADVANCE is a public-private partnership aiming to develop and test a system for rapid benefit-risk (B/R) assessment and NRT B/R monitoring of vaccines in the post-market setting using a distributed network of European electronic health record (EU eHR) databases (6).

A first set of four proof-of-concept (POC) studies were conducted in 2016 to test the system and workflows for generating the required data to perform the B/R assessment of vaccines in Europe. Particularly, these studies assessed the feasibility for generating data for pertussis vaccination coverage (7), benefits (8) and risks (9), and for synthesizing the obtained evidence through B/R modeling (10). As test-case for these POC studies, it was assessed if the initial B/R profile of pertussis vaccines in children (<6 yrs) was maintained after the switch from whole cell pertussis (wP) vaccines to acellular pertussis (aP) vaccines.

ADVANCE also piloted the use of an interactive dashboard for vaccine B/R monitoring based on simulated data mimicking the introduction of rotavirus vaccination in the UK (11). Although the potential of such a dashboard was recognized by various potential end-users, the main concern was the availability of appropriate EU data for monitoring, requiring frequent data updates and small time-lags between the occurrence of the events and their release date (11).

Building upon previous work (7-9, 11), this paper addresses the actual feasibility of NRT B/R monitoring using real world data on pertussis vaccination coverage, benefits and risks from EU eHR databases. The objectives of the current study are twofold: 1) to explore the capacity of EU eHR databases to perform NRT monitoring and 2) to demonstrate the practical potential of B/R monitoring using an interactive dashboard populated with real-world evidence.

3. MATERIALS AND METHODS

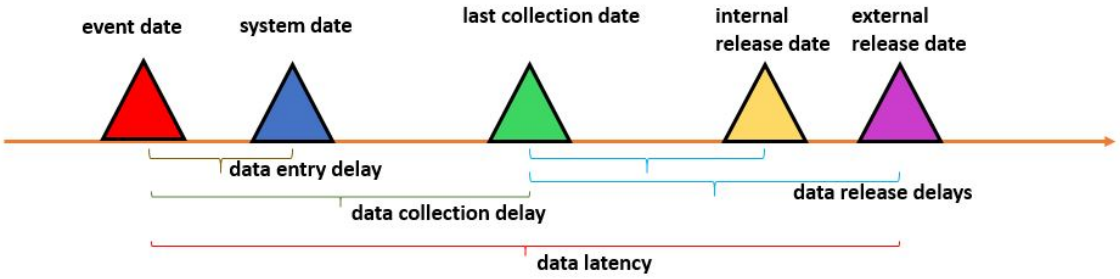
2.1 Participating databases

The nine ADVANCE eHR databases who successfully participated to the fit-for-purpose assessment (ref paper 3), were invited to participate to this study. The databases were based in four countries: Denmark, Italy, Finland, Spain and the UK. For Denmark, there was one regional (AUH: Aarhus Universitets Hospital) and one national (SSI: Statens Serum Institut) population-based hospital discharge database linked to vaccination registries. For Italy, there were three databases: two were administrative databases with data from the population from a geographical area (Agenzia Regionale di Sanità, Tuscany region (ARS); Agenzia di Tutela della Salute Val Padana (ATSVP), a subarea of Lombardy region) and one was a primary care paediatric database linked to the Veneto vaccine registry (PEDIANET). For Spain, there was one multi-regional primary care (Base de Datos para la Investigación Farmacoepidemiológica en Atencion Primaria (BIFAP)) and one regional primary care and hospital medical record database (Information System for Research in Primary Care (SIDIAP)). Finally, for the UK, there were two national primary care medical record databases (The Health Improvement Network (THIN), and the Royal College of General Practitioners (RCGP) Research and Surveillance Centre (RSC)) (12).

2.2 Qualitative feasibility assessment

The feasibility of conducting NRT monitoring of vaccination coverage, benefits and risks was first assessed using a survey. The survey contained questions on the time required for the different data processing steps between the actual date of an event and releasing the data for statistical analysis (or monitoring). The following steps were distinguished: event date (i.e. date of assumed diagnosis for events recorded at primary care, date at hospital admission for events recorded at the hospital or vaccination date), system date (i.e. date on which the information is electronically recorded using the medical software), data collection date (i.e. data lock point or the cut-off point for data to be entered in the database), internal and external release date (i.e. date on which the data are ready for querying internally or by external parties) (Figure 1). In addition, questions were asked regarding data delays and main barriers to implementing NRT monitoring, if any. The following data delays were distinguished; data entry delay (i.e. difference in time between the event date and system date), data collection delay (i.e. difference in time between the event date and data collection date), data release delay (i.e. difference in time between last collection date and release date(s)) and the total data latency (i.e. difference in time between the event date and the release date(s)) (Figure 1). The survey responses were discussed during a face-to-face meeting among the participating ADVANCE data partners.

Figure 1: *Chronological steps from event (exposure or outcome) to releasing data for statistical analysis (or monitoring).*



2.3 Proof-of-concept study on NRT monitoring

Based on results of the qualitative feasibility assessment, databases were invited to participate to the study on NRT monitoring of coverage, benefits and risks using aP vaccination as test case. The study protocol can be found at the EU PAS Register (EUPAS26809) and builds further upon the study protocols of the earlier conducted ADVANCE POC studies on pertussis vaccination coverage (EUPAS13908), benefits (EUPAS13766) and risks (EUPAS13779) (7-9).

2.3.1 Study design & period

This study was a retrospective dynamic cohort study, with the observation period divided over two distinct periods. The first period started arbitrarily from 01 January 2014 until the start of the NRT monitoring (March 2018), which has the objective of establishing a baseline. The second period started immediately after the first period and lasted about three months, which has the objective of testing the actual feasibility of weekly NRT monitoring.

2.3.2 Study population

The source population consisted of children in the participating EU eHR databases from the start of the study period or first entry in the database (whichever occurred last) until administration of the pre-school-entry booster, their sixth birthday, death, transfer out of the database or last data collection date (whichever occurred first). Children with missing information on month and year of birth and date of start of follow-up were excluded.

2.3.3 Vaccination and health outcome events

The exposure of interest was vaccination with any aP-containing vaccine (either as a single component or part of a multivalent vaccine product) by dose (dose 1, dose 2 or dose 3). The risk outcomes of interest were febrile convulsions/seizures, fever, hypotonic hypo-responsive episodes (HHE), somnolence and persistent crying. The benefit outcome of interest (i.e. the vaccine preventable disease) was confirmed or probable pertussis.

For the risk outcomes, events were identified that occurred within pre-defined risk windows post-vaccination (febrile convulsions/seizures and fever: 0-3 days; HHE and somnolence: 0- 2 days and persistent crying: 0-1 day post-vaccination) or that occurred within the baseline window of 10-15 days post-vaccination. The case definitions, risk windows and code lists were generated as part of the previous POC studies on vaccination coverage (7), benefits (8) and risks (9) (refs papers 4-5-6 to be checked/updated).

2.3.4 Local data processing and sharing of aggregated data

All ADVANCE data partners worked in a distributed manner, following a common protocol, common data model (CDM) and common data transformation scripts (ref paper 2). In summary, individual-level data were extracted and transformed locally into three study-specific CDM files; a 1st file containing information on the patients, a 2nd file containing information on the vaccinations and a 3rd file containing information on the benefit and risk outcomes of interest. Subsequently, these CDM files were locally transformed into aggregated data outputs using common scripts.

The aggregated data outputs contained information on the number of administered vaccine doses by calendar time (in weeks) and age (in weeks); the total number and the number of vaccinated subjects within year-month birth cohorts by age (in weeks); the number of pertussis events by calendar time (in weeks) and the number of risk outcomes within the risk windows and within the baseline windows by calendar time (in weeks) as well as the corresponding person time information.

The aggregated data outputs were subsequently transferred from the local databases to a central server for further statistical analysis to generate input for the interactive dashboard. During the 3-month period of NRT monitoring, the process of data collection and transformation into the CDM was repeated on a weekly basis. Each time data were collected from the start of the study (1st January 2014) until the last data collection date.

2.3.5 Interactive dashboard

Data from the entire study period (i.e. baseline period and NRT period) were used for the interactive dashboard, which was developed to facilitate the visual monitoring of vaccination coverage, benefits and risks. The dashboard contains over 20 interactive graphs and is freely accessible upon registration from https://advance.p-95.com/brpertussis/.

To create input for the interactive dashboard, the total number of administered doses n_{ij} (dose1, dose2 and dose3) during week i in age group j was calculated for each database. In addition, the coverage at week i for birth-month cohort j was calculated by dividing the number of vaccinated subjects n_{ij} by the total number of subjects still under follow-up at week i for subjects with start of follow-up no later than 6 weeks. The benefits were monitored by calculating the weekly pertussis incidence rate (per 100,000 person-years), complemented with exact Poisson 95% confidence intervals (CIs). To monitor safety, incidence rates (per 1000 person-years) within the pre-defined outcome-specific risk windows and within the baseline risk windows were calculated cumulatively over time, combining all data from the start of the study period until week i. Each time, exact Poisson 95% CIs were obtained.

2.3.6 Data latency

Data collected during the 3-month period of NRT monitoring were used to calculate data collection delays. In particular, data collection delays were calculated based on subsequent data extracts ≤ 1 week apart. For each data extract i, we only kept the records referring to the newly added events (i.e. events present in data extract i, but not yet present in the previous data extract i-1). For each newly added event, the data collection delay was calculated as the difference between the event date and the data collection date of the data extract i when the event appeared the first time. In addition, we assessed the time required from data collection to locally transforming the CDM files into aggregated data outputs for transfer to the central server.

2.3.7 Software

Depending on the database, different softwares (e.g. SQL server, DB visualizer, SQL management server, MSAccess, SAS) were used to locally extract the data and transform to the CDM files. For all subsequent data transformation steps R3.4.0 was used. The dashboard was developed using Shiny R 3.4.4 (13).

2.3.8 Ethics

The study was a continuation of the earlier conducted ADVANCE POC studies on vaccination coverage, benefits and risks (7-9). The current study was in accordance with the ADVANCE Code of Conduct (14) for the vaccination, benefit and risks events and with the ENCePP Code of Conduct (15) for the vaccination and risk events.

4. Results

4.1 Qualitative feasibility assessment

Nine databases (AUH, SSI, ARS, ATSVP, PEDIANET, BIFAP, SIDIAP, THIN and RCGP RSC) participated to the qualitative feasibility assessment, based on which four databases (AUH, PEDIANET, BIFAP, THIN) decided to not participate to this POC study. AUH, the regional Danish database, did eventually not participate as the procedure for data access changed before the start of the study, making it unlikely to get approval in due time. PEDIANET could not provide data within the study period on subjects < 6 years of age as they only have vaccination data on birth cohorts from 2006 and 2007. BIFAP decided to not participate as the database is only updated annually. Finally, the initial access provider for THIN within the ADVANCE consortium (Erasmus Medical University Centre) was no longer having a license for THIN at the time of study conduct (Table 1). The remaining five databases (SSI, RCGP RSC, ATSVP, ARS and SIDIAP) agreed to participate to this POC study.

Table 1: *Main results database survey on the feasibility of NRT monitoring*

Country - database	**Expected time required between**			**Is NRT monitoring feasible?**	**What is needed to implement NRT monitoring or further reduce data latencies?**
	System date and data collection date	**Data collection date and internal release date**	**Data collection date and external release date**		
Denmark					
AUH	Up to 3 months	1 day	N.A.	Yes, but with delays	Registries are presently updated monthly. A more frequent update would reduce data latencies.
SSI	<2 days	1 day	N.A.	Yes, but with delays.	Monitoring is resource intensive. Additional programs and procedures should be developed.
Italy					
ARS	2 to 4 weeks	1 to 3 months	N.A.	Yes, but with delays. All data (with exception of exposure events) are received roughly on a monthly basis and with 45 days delay on average. Vaccination data was incomplete before 2018, because vaccinations administered in primary care paediatrician practices were not included in the available data.	To further decrease delays, an agreement with the regional healthcare system is needed to be able to receive weekly updates.
ATSVP	4 weeks	<1 week	N.A.	Yes, pending feasibility	Political endorsement and

					dedicated personnel
PEDIANET	1 day	1 week	N.A.	Yes, but only for selected health outcomes. Vaccination data is not yet available.	Time consuming activity, for which approval is needed.
UK					
THIN Access provided through Erasmus University Medical Centre	Not provided	Not provided	>6 months	No, not under the current licensing agreement with the Erasmus University Medical Centre	Change the licensing agreement to allow for weekly updates.
RCGP RSC Access provided through University of Surrey	<1 week	3 days	1 week	Yes	Weekly monitoring is possible, Delays in data recording where vaccination takes place outside general practice. Access to GPs may limit immediacy of reporting.
Spain					
BIFAP	Annual updating of information from Jan 1st to Dec 31st.	1 to 3 months	1 to 3 months	No, not under the current agreements and governance model of the database	More frequent update would be necessary for NRT monitoring, However it is not possible under the current agreements and governance model of the database.
SIDIAP	2-14 months (annual updating of information from Jan 1st to Dec 31st)	2-3 months	N.A.	No, not under the current agreement with the Catalan Institute of Health.	The Catalan Institute of Health receives data continuously. We would need to collaborate closely with them to get NRTdata. Resources are also needed to set up a system.

N.A. not applicable as data are not available for analysis by third party

4.2 Proof-of-concept study on NRT monitoring

The five databases participating to this POC study varied in size, with SSI providing data on 374,161 subjects (0-6 years) at the start of the observation period, RCGP RSC on 6,362 subjects, ATSVP on 2,492 subjects, ARS on 84076 subjects and SIDIAP on 28089 subjects. All databases provided baseline data for the vaccination and risk events whereas only four databases provided baseline data for the benefit events. ARS did not provide data for the benefit events as ARS can only participate in studies compliant with the ENCePP Code of Conduct (15), prohibiting partners from industry to be the principal investigator, which was the case for the benefit events. Four databases decided to participate to NRT monitoring. SIDIAP decided to not do so as the database is only updated once a year. ARS could eventually also not participate as the anticipated improvement in their access to the vaccination registry was delayed beyond the period of this study. The three databases that could successfully participate to the NRT monitoring, provided 11 (SSI), 7 (ATSVP) and 11 (RCGP RSC) data extractions during the 3-month period of NRT monitoring. All three databases were able to provide data extractions on a weekly basis (See Appendix Table 1 for data extraction dates).

3.2.1 Interactive dashboards

As the dashboard was developed for system testing, the provided visualizations should not be used to inform clinical or regulatory decision making. For illustrative purposes, we here give graphical representations of the weekly number of administrations of dose 1 by age group and the incidence rate of fever (per 1,000 person-years) estimated cumulatively over time, within the risk window of 0-3 days post-vaccination after dose 1 and within the baseline window.
For all databases except SSI (Denmark), the first dose of aP is mostly given to children below 12 weeks of age (Figure 2). For SSI, the majority of doses is administered to children older than 12 weeks. The fever incidence rate within the risk window after dose 1 was not different from the baseline incidence rate, with the exception of the UK. In the RCGP RSC database, an increase in the fever incidence after dose 1 (Figure 3) and dose 3 (see dashboard) but not dose 2 (see dashboard) could be clearly seen from March 2016 onwards, which is likely explained by the co-administration of Meningitis B, which was introduced in the UK routine childhood vaccination program on the 1st of September 2015 (16, 17).

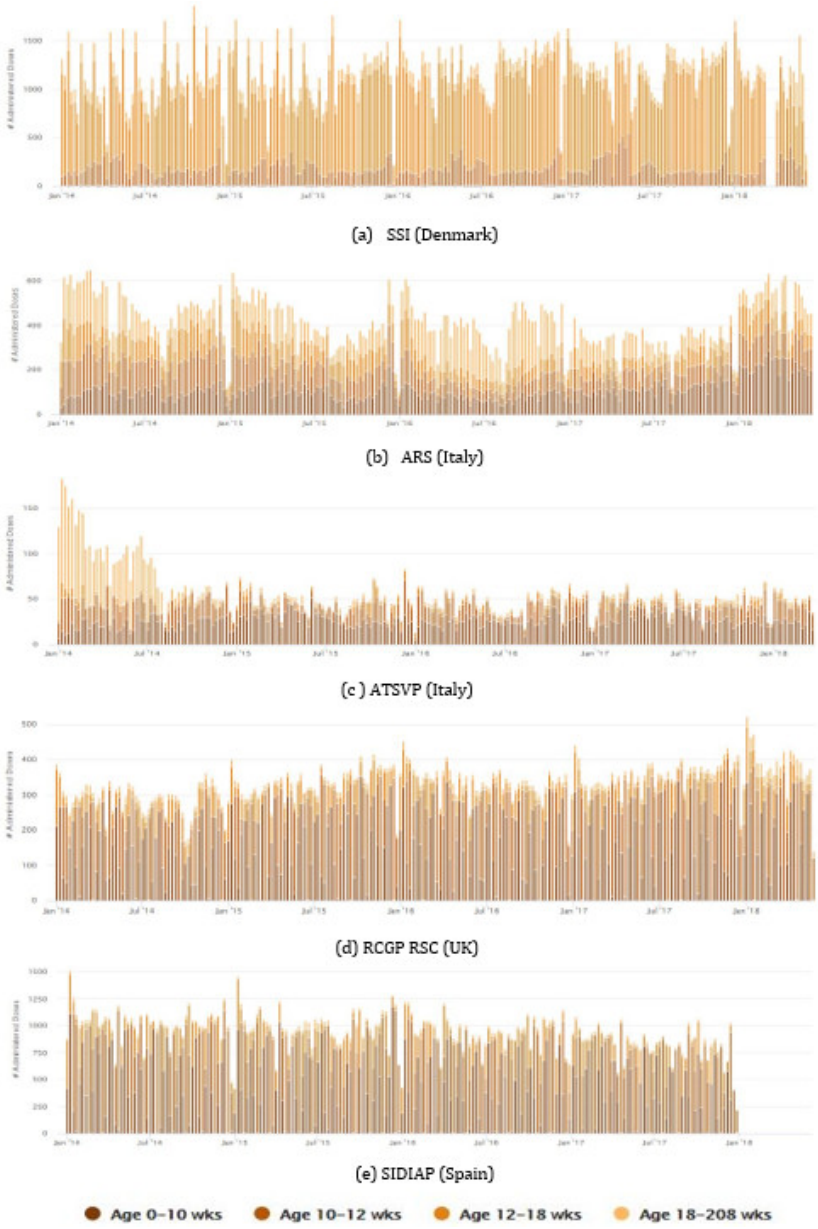


***Figure 2**: An example of dashboard visualizations on vaccination: number of administered dose 1 by age group, by calendar time (in weeks)*

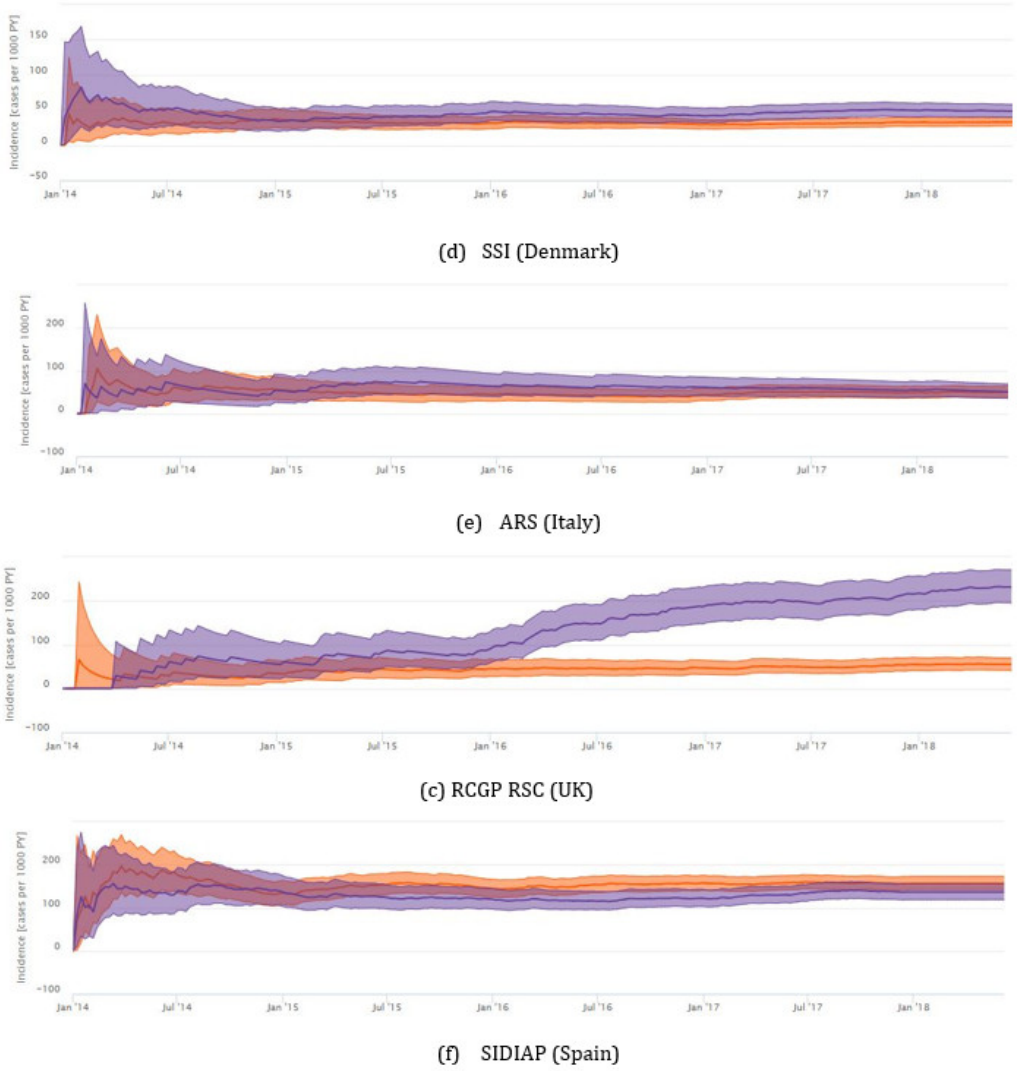


Figure 3: *An example of dashboard visualizations on safety: incidence rate per 1000 person years (95% CI) within the risk window of 0-3 days post-vaccination (purple) after dose 1 and within the baseline window (orange). ATSVP is not shown as fever is poorly reported in this database with only very few events within the risk and baseline windows.*

3.2.2 Data latency

For vaccinations irrespective of dose, the majority of data collection delays were short for all three databases (Figure 4, Table 2 which only shows dose 1 as dose 2 and 3 are similar). The median collection delays for vaccination events (all doses) were 2 (SSI), 2 (ATSVP) and 1 (RCGP RSC) weeks (Table 2). For the outcomes, the collection delays varied more widely (Figure 4) with median delays of 2 (SSI), 16 (ATSVP) and 1 (RCGP RSC) weeks (Table 2).

For SSI and RCGP RSC, over 70% of the vaccination, risk and benefit events were collected within 4 weeks.
The time t (i.e. time required for internal pre-processing and quality checks, mapping to the CDM files, transforming to aggregated data outputs and transferring to the central server) is short, with on average 1 to 2 days for SSI, 1 day for ATSVP and 3 days for RCGP RSC. For this POC study, uploading the aggregated data outputs to the dashboard was not automated and took approximately 1 hour.

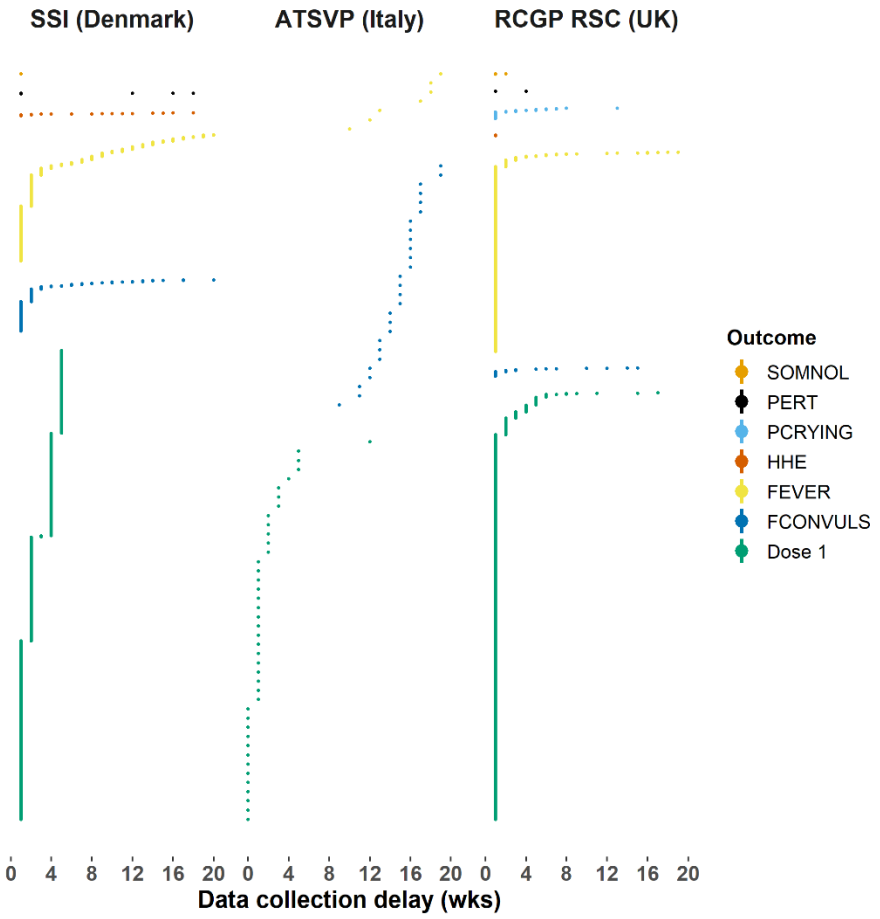


Figure 4: *Data collection delays (in weeks) ≤20 weeks, by event type and database (dose 2 and dose 3 not shown).*
SOMNOL = somnolence, PERT = pertussis, FCONVULS = febrile convulsions, PCRYING = persistent crying, HHE = hypo-tonic hypo-responsive episodes.

Table 2: *Data extraction delays for different types of outcomes and vaccination doses, by database*

	SSI (Denmark)						**ATSVP (Italy)**						**RCGP RSC (UK)**					
	N	**Min**	**Med**	**≤4wk (%)**	**≤8wk (%)**	**≤20wk (%)**	-						**N**	**Min**	**Med**	**≤4wk (%)**	**≤8wk (%)**	**≤20wk (%)**
Somnolence	2	1	1	100%	100%	100%	-						6	1	1	83.3%	83.3%	83.3%
Pertussis	7	1	12	42.9%	42.9%	85.7%	-						3	1	1	100%	100%	100%
Persistent crying	-						-						65	1	1	78.5%	96.9%	98.5%
HHE	29	1	3	55.2%	62.1%	96.6%	7	10	17	0	0	100%	3	1	1	66.7%	66.7%	66.7%
Fever	1577	1	2	62.7%	68.9%	83.1%	27	9	15	0	0	100%	1231	1	1	95.6%	96.9%	97.7%
Febrile convulsions	544	1	1	86.0%	91.7%	97.6%	34	9	15	0	0	100%	54	1	1	77.8%	87.0%	94.4%
All outcomes	2159	1	2	68.5%	74.5%	87.0%							1362	1	1	94.0%	96.4%	97.5%
							60	0	2	63.3%	68.3%	70.0%						
aPE dose 1	4924	1	2	81.5%	99.1%	99.1%	111	0	1	82.9%	82.9%	85.6%	2584	1	1	96.8%	99.5%	99.7%
aPE dose 2	3649	1	4	76.0%	97.8%	97.8%	95	0	1	93.7%	93.7%	93.7%	2409	1	1	97.3%	98.9%	99.4%
aPE dose 3	3823	1	1	79.6%	98.4%	98.4%	266	0	1	82.3%	83.5%	85.0%	2441	1	1	97.7%	98.2%	98.9%
All doses	12396	1	2	79.3%	98.5%	98.5%	-						7434	1	1	97.2%	98.9%	99.3%

N = number of events, Min = minimum, Med = medium

HHE = hypo-tonic hypo-responsive episodes. aPE = acellular pertussis vacc

5. Discussion

With this study, we demonstrated the actual feasibility of NRT monitoring of vaccination coverage, benefits and risks using some EU eHR databases. Out of the nine ADVANCE databases that participated to the survey-based qualitative feasibility assessment, five databases (SSI from Denmark, ARS and ATSVP from Italy, SIDIAP from Spain and RCGP RSC from the UK) participated to the dynamic cohort study providing baseline data. We additionally developed an interactive dashboard to facilitate the monitoring of vaccination coverage, benefits and risks, for which all five databases provided data. Three databases (SSI, ATSVP and RCGP RSC) also performed NRT monitoring, proving data extractions on a weekly basis during 3 months. These data extractions were subsequently used to calculate data collection delays (i.e. time between event date and last collection date). For the three participating databases, the vaccination events showed a median delay of 1 to 2 weeks, whereas differences across databases were observed for the health outcome events. In particular, ATSVP (Italy) showed longer delays for the health outcomes compared to other two databases. For all databases and all event types, the vast majority of data collection delays were ≤ 8 weeks. These data collection delays are a good proxy of the total data latencies (i.e. time between event date and release date) as the data release delays (i.e. time between data collection and data release date) were maximum a few days.

However, the data collection delays were calculated based on a limited number of events collected during a short three month period of NRT monitoring. The period of NRT monitoring was deliberately kept short as providing weekly data extractions is resource-intensive. The here reported data collection delays are therefore indications only, and do not represent a proper distribution of the possible delays. The data collection delays depend on the database and type of event. The delays for the vaccination events may also vary within databases/countries depending on the type of vaccination. Particularly school-based vaccinations or travelers vaccines are less likely to be (timely) recorded in many primary care databases depending on the national regime for vaccine administration.

The theoretical feasibility of using EU eHR databases for NRT monitoring has been studied before. Leite et al studied recording delays (i.e. difference in time between the assumed date of diagnosis and the system date) from events of interest for vaccine safety monitoring in the CPRD (Clinical Practice Research Datalink) (3), a UK primary care database with currently over 5 million active patients (18). They found that over 70% of the events they studied had recording delays ≤ 30 days, which is in line with the data collection delays we found for SSI and RCGP RSC. In subsequent research, the same authors assessed the statistical power and time to signal safety outcomes using continuous sequential tests and concluded that there is limited power to timely detect small to moderate increases in risk using the CPRD only and that larger sample sizes are required (4).

The three databases that participated to the NRT monitoring of this POC study (SSI, ATSVP and RCGP RSC) have currently a total of above 11.3 million active patients. Given the main reasons of not participation were rather due to licensing and initial agreement than technical feasibility, we would expect to get more substantial numbers from additional database partners.

For instance, when adding ARS and SIDIAP, the total number of active patients would be above 22 million. Having such numbers available would greatly improve the performance of NRT monitoring systems provided that heterogeneity in databases are well studied and accounted for. Continued capacity building and adequate resources are required to further develop and enhance a sustainable vaccine monitoring system in Europe with a wide geographical coverage and sufficient sample size to monitor also rare events. To this end, VAC4EU (Vaccine monitoring Collaboration for Europe) was launched in March 2019. VAC4EU is a multi-stakeholder international association intended to be the sustainability solution of the ADVANCE project. (https://vac4euorgdev.wpengine.com/).

6. References

[1] Lopalco PL, Johansen K, Ciancio B, De Carvalho Gomes H, Kramarz P, Giesecke J. Monitoring and assessing vaccine safety: a European perspective. Expert Rev Vaccines. 2010;9:371-80.

[2] McNeil MM, Gee J, Weintraub ES, Belongia EA, Lee GM, Glanz JM, et al. The Vaccine Safety Datalink: successes and challenges monitoring vaccine safety. Vaccine. 2014;32:5390-8.

[3] Sturkenboom M, Bahri P, Chiucchiuini A, Grove Krause T, S. H, Khromava A, et al. Why we need more collaboration in Europe to enhance post-marketing surveillance of vaccines. Vaccine. 2018;Paper 1 in ADVANCE supplement.

[4] Leite A, Andrews NJ, Thomas SL. Assessing recording delays in general practice records to inform near real-time vaccine safety surveillance using the Clinical Practice Research Datalink (CPRD). Pharmacoepidemiol Drug Saf. 2017;26:437-45.

[5] Leite A, Thomas SL, Andrews NJ. Implementing near real-time vaccine safety surveillance using the Clinical Practice Research Datalink (CPRD). Vaccine. 2017;35:6885-92.

[6] de Lusignan S, Ferreira F, Damaso S, Byford R, Pathirannehelage S, Yeakey A, et al. Enhanced passive surveillance of influenza vaccination in England, 2016-2017 - an observational study using an adverse events reporting card. Hum Vaccin Immunother. 2019.

[7] Sturkenboom M, van der Aa L, Bollaerts K, Emborg HD, Ferreira G, Gino R, et al. The ADVANCE distributed network system for evidence generation on vaccines coverage, benefits and risks based on electronic health care data. Vaccine. 2018;Paper 2 in supplement.

[8] Emborg HD, Berensci K, Braeye T, Bauwens J, Bollaerts K, Correa A, et al. ADVANCE system testing: can coverage of pertussis vaccination be estimated in EU countries using electronic health data: an example. Vaccine. 2018;Paper 4 in Supplement.

[9] Tin Tin Htar M, de Ridder M, Braeye T, Correa A, McGee C, de Lusignan S, et al. ADVANCE system testing: vaccine benefit studies using multi-country electronic health data - the example of pertussis vaccination. Vaccine. 2018:Paper 5 in this supplement.

[10] Weibel D, Dodd C, Mahaux O, Haguinet F, de Smedt T, Duarte-Salles T, et al. ADVANCE system testing: can safety studies be conducted using electronic health data: an example with pertussis vaccination? Vaccine. 2018;Paper 6 in Supplement.

[11] Bollaerts K, Ledent E, de Smedt T, Weibel D, Emborg HD, Correa A, et al. ADVANCE system testing: benefit-risk analysis of a marketed vaccine using multi-criteria decision analysis and cohort modelling. Vaccine. 2018;Paper 9 in this supplement.

[12] Bollaerts K, De Smedt T, Donegan K, Titievsky L, Bauchau V. Benefit-risk monitoring of vaccines using an interactive dashboard: a methodological proposal from the ADVANCE Project. Drug Saf. 2018;41:775-86.

[13] Sturkenboom M, Weibel D, van der Aa L, Braeye T, Gheorge M, Becker B, et al. ADVANCE database characterization and fit for purpose assessment for multi-country studies on the coverage, benefits and risks of vaccinations. Vaccine. 2018;Paper 3 in Supplement.

[14] Winston Chang JC, JJ Allaire, Yihui Xie and Jonathan McPherson. Shiny: Web Application Framework for R. 2016.

[15] Kurz X, Bauchau V, Mahy P, Glismann S, van der Aa LM, Simondon F, et al. The ADVANCE Code of Conduct for collaborative vaccine studies. Vaccine. 2017;35:1844-55.

[16] Gini R, Fournie X, Dolk H, Kurz X, Verpillat P, Simondon F, et al. The ENCePP Code of Conduct: A best practise for scientific independence and transparency in noninterventional postauthorisation studies. Pharmacoepidemiol Drug Saf. 2019.

[17] Ladhani SN, Campbell H, Parikh SR, Saliba V, Borrow R, Ramsay M. The introduction of the meningococcal B (MenB) vaccine (Bexsero(R)) into the national infant immunisation programme--New challenges for public health. J Infect. 2015;71:611-4.

[18] Harcourt S, Morbey RA, Bates C, Carter H, Ladhani SN, de Lusignan S, et al. Estimating primary care attendance rates for fever in infants after meningococcal B vaccination in England using national syndromic surveillance data. Vaccine. 2018;36:565-71.

[19] Herrett E, Gallagher AM, Bhaskaran K, Forbes H, Mathur R, van Staa T, et al. Data Resource Profile: Clinical Practice Research Datalink (CPRD). Int J Epidemiol. 2015;44:827-36.

7. Disclaimer

The results described in this publication are from the proof of concept studies conducted as part of the IMI ADVANCE project with the aim of testing the methodological aspects of the design, conduct and reporting of studies for vaccine benefit-risk monitoring activities. The results presented herein relate solely to the testing of these methodologies and are not intended to inform regulatory or clinical decisions on the benefits and risks of the exposures under investigation. This warning should accompany any use of the results from these studies and they should be used accordingly.
The views expressed in this article are the personal views of the authors and should not be understood or quoted as being made on behalf of or reflecting the position of the agencies or organisations with which the authors are affiliated.

8. Funding source

The Innovative Medicines Initiative Joint Undertaking funded this project under ADVANCE grant agreement n° 115557, resources of which were composed of a financial contribution from the European Union's Seventh Framework Programme (FP7/2007-2013) and in kind contributions from EFPIA member companies.

9. Declaration of potential conflicts of interest

Tom de Smedt, Hanne-Dorthe Emborg, Talita Duarte-Salles declared no potential conflicts of interest.
Kaatje Bollaerts and Maria Alexandridou received consultancy fees from GSK unrelated to this work.
Simon de Lusignan is Director of the RCGP RSC within his academic role at the University of Surrey. He is also (though his University) member of Sanofi and Seqirus advisory boards. He has investigator led funding (relevant to vaccination) from Seqirus, Takeda, and GSK.
Vincent Bauchau is an employee of the GSK group of companies and holds company shares
Myint Tin Tin Htar is an employee of Pfizer Inc and holds company shares. Miriam Sturkenboom declared that she has received grants from Novartis, CDC and Bill & Melinda Gates Foundation, for work unrelated to the submitted work.

10. Acknowledgements

The authors would like to thank Margaret Haugh, MediCom Consult, Villeurbanne, France for editorial services; Johnny Kahlert (Aarhus Universitets Hospital), Consuelo Huerta (BIFAP), Lara Tramontan and Gino Picelli (PEDIANET) for providing information on their databases and Rachel Byford, Mariya Hriskova, Filipa Ferreira and Ivelina Yonova (RCGP RSC) for support to the RCGP RSC database.

Appendix 1: statistical analysis plan

LIST OF ABBREVIATIONS

ADVANCE	Accelerated Development of VAccine beNefit-risk Collaboration in Europe
aP	acellular (pertussis vaccine)
B/R	Benefit/ Risk
CDM	common data model
DLP	data lock point
HHE	hypotonic-hyporesponsive episode
IMI	Innovative Medicines Initiative
POC	proof-of-concept
SAP	statistical analysis plan

RESPONSIBLE PARTIES

Main Author(s) of the SAP

Name	***Institution***	***Role***	***Contributions***
Kaatje Bollaerts	P95	Statistician	First draft v0.1 and updates until last version
Tom de Smedt	P95	Data analyst	Substantial input data management plan
Vincent Bauchau	GSK	Epidemiologist	Substantial comments and discussions
Miriam Sturkenboom	P95	WP5 lead	Substantial comments and discussions

Note: Sections that are taken from POC1 protocols and POC1 SAPs are indicated as such. These reference documents can be found at the ADVANCE sharepoint: Work Package 5 – POC 1 – Study teams working documents library :

https://publication.wiv-isp.be/workspaces/advance/SitePages/Home.aspx

Study protocol version	**Read and approved by (name)**	**Role**	**Signature**	**Date**
V1.0	Kaatje Bollaerts	Principal investigator		29/11/2018
V1.0	Hanne-Dorthe Emborg	Principal investigator		
V1.0	Myint Tin Tin Htar	Principal investigator		
V1.0	Daniel Weibel	Principal investigator		

RATIONALE AND BACKGROUND

At the core of the mission of ADVANCE and many of its stakeholders is the concept of vaccine Benefit/Risk (B/R) monitoring. Monitoring should be understood as a periodic assessment of several key parameters including, coverage, incidence of adverse events, and incidence of the vaccine preventable disease to trigger an alert if and when there is an indication that the B/R profile in the population is different from what is expected (based on clinical trials, observational studies or similar products). This alert would generate a subsequent and possibly more formal assessment and analysis of the vaccine. Monitoring should, in principle, start as soon as a new vaccine is introduced in a given country and continue throughout the vaccine's lifecycle, and would also be applied to vaccines with established B/R profiles.

B/R monitoring requires information that is available in a timely fashion. Hence, the goal is to have access to near real-time information, which is defined as either weekly or monthly refresh of data that is only a few days old. Visual monitoring of the component parameters of B/R as well as two composite B/R measures has been recently prototyped in a dashboard using simulated data on rotavirus (http://apps.p-95.com/BRMonitor/). This approach was found useful and informative. Subsequently, the dashboard was further developed to incorporate the retrospective real world data on pertussis from the first proof of concept study (POC1), demonstrating good acceptability.

The current POC1.2 will be a real-world test case of near real-time visual monitoring of coverage, benefits and risks using electronic healthcare data. POC1.2 will be designed as a continuation of POC1 to maximally leverage previous work. The dashboard developed as additional analyses to POC1 as described above will be used for monitoring. The expected learnings from POC1.2 include:

1. Identifying databases with proven fast and frequent data refresh that can be used for near real time monitoring of vaccination coverage, benefits and risks rates (on top of the theoretical ability as identified based on a survey), and understanding the reasons, if any, why some databases cannot be used for monitoring.
2. Testing the requirements for data flow and data processing to ensure near real-time monitoring;
3. Using a dashboard for near real-time monitoring.

RESEARCH QUESTION AND OBJECTIVES

The overall objective of POC1.2 is to determine the feasibility of periodic and rapid assessments of vaccine coverage, benefits and risks using electronic healthcare databases.

The specific objectives include:

1. Evaluate the ability of databases to generate periodic data with limited latency to secure near real-time monitoring on a preferably weekly basis (or bi-weekly/monthly if weekly would not be feasible)
2. Visually monitor vaccination coverage, benefits and risks using an interactive dashboard

The test case will be the monitoring of acellular pertussis vaccination in Europe, mimicking the launch of a new vaccine.

STUDY METHODS

Several sections below are taken directly from the POC1 protocol and SAPs because POC1.2 will follow similar processes as POC1. To create a stand-alone document, but to avoid repetition for those familiar with POC1, we indicated the sections from POC1 with a *. Readers familiar with POC1 might skip these sections.

Study design and study period

This is a dynamic cohort study, containing two distinct observation periods. The first period starts from 01 January 2014 until the start of the near real-time monitoring upon approval of the protocol and has as objective to establish a baseline. The second period has the objective of near real-time monitoring and will cover a few months. This period will start upon protocol approval until last periodic data lock point (DLP) (i.e. the time the last periodic database extract is produced) and will cover a few months.

Population

The target population are all children from their start of follow-up in the database until school-entry pertussis booster, 6 years of age or any periodic data lock point within the participating ADVANCE databases.

Participating databases

In total, six European electronic health record (EHR) databases will participate to POC1.2. These databases are ASLCR (Italy), ARS (Italy), PEDIANET (Italy), RCGP (UK), SIDIAP (Spain), SSI/AUH (Denmark). Their selection and description are given in the POC1.2 study protocol.

Exposure

The exposure of interest in this study is vaccination with any acellular pertussis-containing vaccines which could be identified in the participating databases during the study period. The ATC codes listed in Table S1 will be used to define exposure.

Details on the definition of exposure are given in Sections 6.1.2 and 6.2.2. We used the *VaccO Selector* web application to identify ATC codes related to Pertussis. The *VaccO Selector* application analyses a user-provided vaccine coding system, and enables the user to select codes based on their VaccO vaccine properties.

The VaccO Selector is available at https://euadr.erasmusmc.nl/VaccO/#!/selection and the process is described in:Alignment of vaccine codes using the VaccO ontology. Benedikt Becker, Jan Kors, Erik Mulligen, Miriam Sturkenboom. *Submitted*, 2018

Table S1 Acellular pertussis-containing vaccines of interest - ATC code for defining the exposure in any database (From POC1 coverage SAP)

ATC code	ATC name
J07AG52	haemophilus influenzae B, combinations with pertussis and toxoids
J07AJ	Pertussis vaccines
J07AJ02	pertussis, purified antigen
J07AJ52	pertussis, purified antigen, combinations with toxoids
J07CA02	diphtheria-pertussis-poliomyelitis-tetanus
J07CA05	diphtheria-hepatitis B-pertussis-tetanus
J07CA06	diphtheria-haemophilus influenzae B-pertussis-poliomyelitis-tetanus

J07CA09	diphtheria-haemophilus influenzae B-pertussis-poliomyelitis-tetanus-hepatitis B
J07CA11	diphtheria-haemophilus influenzae B-pertussis-tetanus-hepatitis B
J07CA12	diphtheria-pertussis-poliomyelitis-tetanus-hepatitis B
J07CA13	diphtheria-haemophilus influenzae B-pertussis-tetanus-hepatitis B-meningococcus A + C

Outcomes

The outcomes of interest are listed in the Table S2, along with their respective risk windows. The benefit outcome is confirmed or probable pertussis. The code lists for all different outcomes are given in Table S3.

Table S2: Outcomes and definition of risk windows, day 0 is day of vaccination.
(From POC1 risk protocol).

	Outcome	***Risk Window After Vaccination***
Risks		
	Fever	0-72 hours
	Somnolence	0-48 hours
	Persistent crying, irritability	0-24 hours
	Febrile convulsions/seizures	0-72 hours
	HHE	0-48 hours
Benefits		
	Pertussis	

Table S3: Code lists of health outcomes of interest (From POC1 risk and benefit SAP)

Outcome	Data source(s)	Operational definition (any of the codes would qualify)
Fever	All participating healthcare databases	ICD9:078.2, 780.6 ICD10: P81.9, R50 READ-CTV3: 1653. 2E34. ,2EZ.., A782., R006.,R0060, R0061, R0062, R0063, R006z, X76Df, X76Di, X76Dk, X76Dl, X76EF,X76EI, X76EI, XM05S, XM09q, XM0yv, XM0yw, XM1AX, Xa9sd, READ-v2: 165.., 165..,2E...,A782.,R006. ICPC: A03
Somnolence	All participating healthcare databases	ICD-9:780.09, 780.54 ICD-10: G47.1, R40.0 READ-CTV3: R0000, R0001, R0054,X007w, X007x, XM06R, Xa2bY, XaC0p READ-v2: 1B67., 1BX1.,2234.,E2743, R0000,R0001, R0054 ICPC: none (text extractions)
Persistent crying, irritability	All participating healthcare databases (except SSI/AUH)	ICD9: 780.92, 780.95 ICD10: R45.83, R68.11 READ-CTV3: 1B1I0, Xa2lv, Xa9zv READ-v2: 1B1I0 1B1P. ICPC: A15
Febrile convulsions	All participating healthcare databases	ICD-9: 780.31; 780.32 ICD10: R56.0 READ-CTV3: R0030; XM03l READ-v2: 1B6B.; R0030 ICPC: none
Hypotonic hypo-responsive episode (HHE)	All participating healthcare databases	ICD-9: 770.88; 782.5; 782.61; 799.02 ICD10: P84; R09.02; R23.0; R23.1 READ-CTV3: R025.; R0260; R2y01; X76qQ; XM041; XM07N; XM09U; XM09U;Xa9E7 READ-v2: R025.; R0260; R2y01 ICPC: none

Outcome	Data source(s)	Operational definition (any of the codes would qualify)
Pertussis	All participating healthcare databases	ICD-9: 033.9; 484.3 ICD-10: A37 READ-CTv3: A33y. A33yz; A33z.; Ayu39; Ayu3A; H243.; X70I8; XE0Qw; XE0Qw; XM00D READ v2: A33.. Ayu39; Ayu3A; H243. ICPC: R71

Data management

The processing of data will take place at the local level to the extent possible. Aggregated data tables with counts, person time and denominator information by calendar week (and possible age in weeks) will be transferred to the dashboard server as illustrated in Figure 1. These data tables contain the minimal information needed to produce the monitoring visualizations. Some additional data processing will take place at the dashboard server to make the dashboard interactively. All data transformation files will be programmed in R 3.4.0.

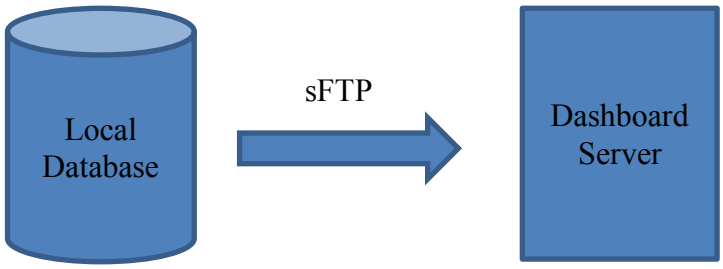


Figure 5 - POC1.2 data management flow

Phase 1: Extraction & transformation of local data to Common Data Model* (from POC1 protocols)

This entails the extraction of study specific individual data from the original databases into study specific common data model (CDM) input files. This will be done by the database custodians locally. The following three CDM input files will be created:

1. ***Patients.txt:*** This input file should contain the information of persons recorded in the database, including patient identifier, date of birth, gender, start and end registration in the database.

2. ***Vaccinations.txt:*** This input file contains information about Pertussis vaccinations, including date of administration, dose recorded and dose derived. The file also includes the patient identifier.
3. ***Events.txt:*** This input file contains information about comorbidity and diagnostic events of interest. The file also includes date of event and patient identifier.

The CDM files stay locally and have the same structure for all databases. Details on the CDM files are given in Section 6.1.

Phase 2: Transformation of CDM data files into the analytical dataset* (from POC1 protocols)

The study specific CDM data will be transformed into the analytical dataset.

The different database teams will run this data transformation file locally to create the database-specific analytical datasets. Testing of the data transformation programs will be done as in POC1. Once the codes are tested, they will be shared with the local database teams, who will run these programs locally, and ensure all needed documentations (log files, recording of site-specific changes to the code, and all versions of the code if there are modifications) are retained and archived.

Phase 3: Transformation of the analytical dataset into data tables

The analytical datasets will be further manipulated locally to create data tables with aggregated data that will serve as input to the dashboard.

To be able to export these data tables to the server environment that will be used for the dashboard, the sFTP protocol will be used to send the data tables directly to server used for the dashboard. The system administrator will verify that the files do not contain any restricted data following the ADVANCE policy.

Phase 4: Web-application server

As described above, the data tables with aggregated data are then transferred from the local databases to a Shiny server application on a server provided by P95. On this server, a Shiny web

application is served. The data tables with aggregated data will be further transformed – based on the user-defined input parameters – to epidemiological summary estimates (e.g. incidence rates, coverages) for the visual monitoring of vaccination coverage, benefits and risks. The additional data transformations based on user-defined input parameters at the Shiny server allows the dashboard to be interactive. Functions in R will be written to perform these additional data manipulations. The resulting plots will be accessible through a web page. Details on the additional data manipulations are given in Section 6.4.

The web-application is a server under the full control of P95, with the following specifications:

- Two vCPUs
- 4GB RAM
- Ubuntu 16.04 Operating system
- Shiny Server based on R 3.4.4
- SSL-certificate from Let's Encrypt (HTTPS connection to the web app)
- User-authentication based on auth0
- Up-to-date virus and malware protection

DATA MANIPULATION AND ANALYSIS

Phase 1: Extraction & transformation of local data to CDM*

The POC1.2 CDM files are the same as for POC1, except that only records on persons from birth cohort 2008 or later are retained. Details are given below.

Patients.txt* (from POC1 SAPs)

This input file might be limited to the information of all persons recorded in the database with year of birth 2008 or later; patients with an invalid date variable for Date of Birth, Start Date or End Date (YYYYMMDD) cannot be included. Patients with rounded date variables can be included. The number of excluded patients shall be counted by the data custodians and depicted using an attrition diagram. For each patient, there is one record (in the case of patients' reentering the population either a suffixed ID number shall be used or the most relevant follow-up period shall be chosen) in the *Patients.txt* input file containing the following variables:

Table S4: Patients.txt

Name of the Variable	Description of the Variable	Format / Possible Values	Description of the Values
PatientID	Patient Identifier	any string maximum length: 32 characters	
Birthdate	Date of birth	YYYYMMDD.	
Gender	Gender	F M	Female Male
Startdate	Date from which the patient is eligible to be included in the study. This is typically the date the patient is entered into the registration system (date of registration with insurance/region, date GP started to collaborate). Run in periods should not be applied when defining StartDate	YYYYMMDD.	
Enddate	Date after which the patient is no longer eligible for inclusion in the study (e.g. end of registration with GP/database, insurance, moving out, death, last data draw down (whichever is earliest)).	YYYYMMDD.	

Vaccinations.txt*(from POC1 SAPs)

This input file contains information about vaccinations for all products of interest. The vaccinations are preferably restricted to vaccinations administered to individuals included in the Patients.txt. For this POC study, only information on acellular pertussis vaccines is required. Vaccinations with an invalid PatientID, Date variable and Dose cannot be included. Patients with rounded date variables can be included. The number of excluded vaccinations shall be counted by the data custodians and depicted using an attrition diagram.

For each vaccination, there is one record in the *Vaccinations.txt* input file containing the following variables:

Table S5: Vaccinations.txt

Name of the Variable	Description of the Variable	Format / Possible Values	Description of the Values
PatientID	Patient Identifier	any string maximum length: 32 characters	
Date	Date of administration	YYYYMMDD.	
Brand	Product name of the vaccine	any string	
VacType	Type of vaccination	Will contain components of the vaccine separated by hyphens.	Pertussis vaccines used in the POC1.2 study will be acellular pertussis containing vaccines (aPE)
ATC	ATC code of the vaccine	7 characters long string	May be shorter if not full ATC
DoseRecorded*	Dose received as specified in the database	P1, P2, etc. B1, B2, etc.	For priming doses For booster doses
DoseDerived*	Defined as dose per recommendation, as determined by the database custodian based on knowledge of the local immunization schedule and as documented in an analytic variable generated for the study	P1, P2, etc. B1, B2, etc.	For priming doses For booster doses

* At least one of them is mandatory. Databases may choose which dose variable to prioritize based upon local expertise and results of the vaccination fingerprint. Both can be filled in at the same time as well.

Events.txt* (from POC1 SAPs)

This input file contains information about comorbidity and diagnostic events (events) of interest for the persons in *Patients.txt*; events with an invalid PatientID or Date variable cannot be included. The number of excluded events shall be counted by the data custodians and depicted using an attrition diagram. The events of interest for POC1.2 are convulsions (CONVULS), febrile convulsions (FCONVULS), hypotonic-hyporesponsive episodes (HHE), persistent crying (PCRYING), somnolence (SOMNOL), and confirmed or probable pertussis (PERT). See POC1.2 study protocol for a rationale of the event selection.

For every event of interest, there is one record in the *Events.txt* input file containing the following variables:

Table S6: Events.txt

Name of the Variable	Description of the Variable	Format / Possible Values	Description of the Values
PatientID	Patient Identifier	any string maximum length: 32 characters	
Date	Date of event	YYYYMMDD.	
Eventtype *	Type of event	CONVULS	CONVULS
		FCONVULS	FCONVULS
		FEVER	FEVER
		HHE	HHE
		PCRYING	PCRYING
		SOMNOL	SOMNOL
		PERT	PERT
Code	Describes database specific code that was used for extraction and the components if an algorithm component analysis will be performed. Code of the variable or one of the variables that will be used to classify the event after harmonization step is completed		

Phase 2: Transformation of CDM data files into the analytical dataset

In this phase, the CDM data will be transformed into the analytical datasets containing minimal anonymized data. Data are minimal in the sense that the analytical dataset contains no more data than

minimally needed for the conduct of this study. For every participating database, analytical datasets will be created.

Study population: Population.txt

The CDM file *Patients.txt* (Table S4) will be extended with additional variables to define the study population and the time at follow-up:

Table S7: Population.txt

Name of the Variable	Description of the variable	Format	Transformation rule
PopulationStart	Start of the patient follow-up for the study	YYYYMMDD	Latest of Startdate (from CDM file patients.txt) and start study period (1st January 2014)
PopulationEnd*	End of the patient follow-up for the study	YYYYMMDD	Earliest of Enddate (from CDM file patients.txt) and end study period (1st March 2018)
PopulationEnd**	End of the patient follow-up for the study	YYYYMMDD	Earliest of Enddate (from CDM file patients.txt) and data lock point***

* First period for establishing baseline

** Second period for near real-time monitoring

*** Data lock point (DLP) is defined as the cut-off date for data to be included in the weekly monitoring. The DLP is the date at which the CDM files are locally generated.

Persons for which person time (calculated as PopulationEnd minus PopulationStart) is negative will be discarded as they do not contribute follow-up time to the study. The new file is named *Population.txt*. The number of patients in *Patient.txt* and in *Population.txt* will be reported in an attrition diagram.

Cleaning the vaccination file: VaccinationsClean.txt (from POC1 coverage SAP)

The CDM file *Vaccinations.txt* (Table S5) will be extended with additional variables to clean the recording of the vaccinations:

Table S8: VaccinationsClean.txt

Name of the Variable	Description of the Variable	Format/ Values	Transformation Rule
VacTypeCombined	Type of vaccination	aPE	Only records with VacType containing aPE are retained.
DoseCombined	Dose of the vaccination	P1, P2, etc. B1, B2, etc.	Depending on the option chosen by the database custodian, DoseRecorded or DoseDerived will be used. The following options were chosen: DoseRecorded: Pedianet, SIDIAP DoseDerived: ASLCR, ARS, RCGP, SSI/AUH
DoseSequence	Sequence number of the vaccination dose	integer	Derived by chronologically ordering and subsequently ranking the different vaccination doses.

- Removing observations with missing date at vaccination
- MinimumNecessaryDistance is implemented and allows for deleting vaccinations that were administered too shortly after each other. (That is, if $Date_{P2} < Date_{P1}$ + MinimumNecessaryDistance, then P2 is deleted.). The value is set to 1 day.
- In case of multiple records with the same value for DoseCombined, the one with the earliest date is kept, the other records are deleted.
- In case of multiple records with the same date at vaccination but different values for DoseCombined, the record with earliest DoseCombined is kept; the other records are deleted.
- If case the dates of the ordered values for DoseCombined (P1 < P2 < P3 < B1 < B2) are not in a chronological order, the dates are swapped to the right chronological order.

The new file is named *VaccinationsClean.txt*. Vaccinations removed by the different steps of the vaccination cleaning are recorded in an attrition diagram.

Cleaning the event file: EventsClean.txt (from POC1 risk SAP)

Events within less than 7 days from an event of the same type will be discarded from the CDM file *Events.txt* (Table S6). Events will be considered recurrent if they are at least 7 days apart, irrespective of other events of the same type in between. For instance, a subject with fever on day 0, day 5 and day 8 will be considered having experienced 2 fever events, one starting on day 0 and one starting on day 8; in case of occurrence of a same event on two different dates for a same patient for which the dates are less than 8 days apart (i.e. date of event2 – date of event1 ≥ 8) then the second occurrence is deleted.

The new file is named *EventsClean.txt*. Events removed by the different steps of the event cleaning are recorded in an attrition diagram.

Analytical datasets

The analytical datasets are created by merging the files *Population.txt*, *VaccinationsClean.tx*t and *EventsClean.txt* by PatientId. Only records for patients in *Population.txt* are retained. The analytical dataset is as described in Table S9. A new anonymized patient identifier will be generated and replaces the original Patientid.

Table S9: Dummy table: analytical dataset with date information for analyzing reporting delays

AnonId	DoB	Cohort Start	Cohort End	EventId	EventDate
P10000	03-03-2014	01-04-14	11-01-18	aP_P1	07-05-2014
P10000	03-03-2014	01-04-14	11-01-18	aP_P2	06-06-2014
P10000	03-03-2014	01-04-14	11-01-18	FEVER	08-05-2014
P10000	03-03-2014	01-04-14	11-01-18	FEVER	08-06-2014
P10000	03-03-2014	01-04-14	11-01-18	ISR	08-05-2014
P10001	26-09-2015	15-01-16	03-03-18	aP_P2	14-01-2016
P10001	26-09-2015	15-01-16	03-03-18	FEVER	23-02-2016
P10001	26-09-2015	15-01-16	03-03-18	FCONVULS	27-08-2016

Phase 3: Transformation of the analytical data into data tables

The analytical datasets created in Phase 2 will be further manipulated locally to create data tables with summary statistics that will serve as input to the B/R dashboard. These aggregated data tables will later be send to the dashboard server and used as inputs to the B/R dashboard.

Data table 1: number of doses

For every database, summary tables with number of administered doses per dose within the primary series (dose 1, dose 2 and dose 3) by calendar time (in weeks) and age at vaccination (in weeks) will be created (Table S10). The following data manipulations steps will be performed:

- Select vaccination events per dose (based on the variable EventId in the analytical dataset)
- Calculate age at vaccination (in weeks) through dividing age at vaccination (in days) by 7 and rounding to the nearest smaller integer
- Calculate time by week and year
- Calculate number of administered doses by calendar time x age x dose

This table will be used as input to the vaccination coverage component of the dashboard (see section 6.4). The data files will be named *POC12_ndose_dbname_dnumber_date.RData.*

Table 3: Dummy data table 1: number of doses given by age at vaccination x calendar time. A table should be generated for every database and every dose.

Dose	Time (wks)*	Age (wks)	N
P1	22	8	1010
P1	22	9	1020
P1	22	10	832
P1	23	9	1006
	23	10	845
	23	11	621
	….		

*weeks since start study (01 January 2014), with first week being denoted week 1

Data table 2: Number of subjects

For every database, summary tables will be created, with the number of children of a given age (in weeks) by year-month birth cohort. The total number of children as well as the children vaccinated with dose 1/dose 2/dose 3 will be obtained (Table S11). The following data manipulations steps will be performed:

- Select vaccination events per dose (based on the variable EventId in the analytical dataset)
- Calculate total number of subjects under follow-up by age (in weeks) and by year-month birth cohort.
- Calculate the number of children vaccinated at age week *i*, or earlier, by dose

This table will be used as input to the vaccination coverage component of the dashboard (see section 6.4). The data files will be named *POC12_ CovbyYMBirth _dbname _date.RData.*

Table S11: **Dummy data table 2: number of children and vaccinated children by age x calendar time. A table should be generated for every database.**

Database	YM_birth	Time (wks)*	N (total)	cumN (dose 1)**	cumN (dose 2)**	cumN (dose 3)**
RCGP	01/05					
RCGP	01/05					
RCGP	01/05					
…	…					
RCGP	02/05					

*weeks since start study (01 January 2014), with first week being denoted week 1

**number of children vaccinated at age in weeks *i* or earlier

Data table 3: Number of risk events

For every database and every risk event of interest (HHE, fever, febrile convulsions, persistent crying, somnolence) summary tables will be created, with the number of risk events within the risk windows after vaccination and outside the risk windows by calendar time (in weeks). The length of the risk window is event-specific (see Table S2). For the number of events outside the risk windows, person time at the time of vaccination is selected, excluding the risk window period and the period 1 week before vaccination as children experiencing one of the risk events are less likely to be vaccinated (healthy

vaccinee bias). An example table is given in Table S12. This table will be used as input to the risk component of the dashboard (see section 6.4). The data files will be named *POC12_nevent_dbname_eventname.RData*.

Table S12: Dummy data table 3: number of events in- and outside the risk windows by calendar time. A table should be generated for every database and risk outcome of interest*.

Time (wks)**	In risk window (dose 1)	In risk window (dose 2)	In risk window (dose 3)	Out risk window (d 1)	Out risk window (d 2)	Out risk window (d3)
23						
24						
...						

*Risk outcomes of interest are: HHE, fever, febrile convulsions, persistent crying and somnolence.

**Weeks since start study (01 January 2014), with first week being denoted week 1

Data table 4: Number of pertussis events

For every database, a summary table will be created with the number of pertussis events (Table S13). This table will be used as input to the benefit component of the dashboard (see section 6.4). The data files will be named *POC12_nevent_dbname_PERT.RData*.

Table S13: Dummy data table 4: number of pertussis events by age group X calendar time. A table should be generated for every database.

Time (wks)*	n
22	3
23	2
25	0
...	0

*weeks since start study (01 January 2014), with first week being denoted week 1

Data table 5: Person time information

For every database, a summary table will be created with the person time relevant for pertussis and the risk events in- and outside the risk windows by calendar time (in weeks). The unique length of risk windows are 1, 2 and 3 days (see Table S2). An example table is given in Table S14. This table will be used as input to the risk and benefit component of the dashboard (see section 6.4). The data files will be named *POC12_py_dbname_date.RData.*

Table S14: Dummy data table 5: number of events in- and outside the risk windows (of length 1, 2, 3 and 7 days) by calendar time. A table should be generated for every database.

Time (wks)*	In risk window of length 1 day (dose 1)	In risk window of length 1 day (dose 2)	In risk window of length 1 day (dose 3)	Out risk window of length 1 day (dose 1)	…	Out risk window of length 3 days (dose 3)	Overall
22							
23							
25							
…							

*weeks since start study (01 January 2014), with first week being denoted week 1

Data transfer to the dashboard server

After the data transformation steps described above, the aggregated data tables need to be send to the dashboard server by sFTP protocol. In particular the files having one of the following naming patterns need to be transferred:

- *POC12_ndose_dbname_dnumber_date.RData* (content as in Table S10)
- *POC12_ CovbyYMBirth _dbname _date.RData* (content as in Table S11)
- *POC12_nevent_dbname_eventname.RData* (content as in Table S12)
- *POC12_nevent_dbname_PERT.RData* (content as in Table S13)
- *POC12_py_dbname_date.RData* (content as in Table S14).

Phase 4: Web-application server

The different files sent by the database custodians to the dashboard servers are used to populate the different visualizations in the dashboard. The only data transformation these aggregated tables still undergo on the dashboard server are transformation with regards to the specific user-inputs a dashboard user has inputted on the dashboard (f.e. the definition of the age groups).

The interactive dashboard will be the one currently under development for the POC1 data. The dashboard contains several tabs: one for visualizing coverage, one for benefits and one for risks. The different countries will be presented next to each other with drop menus to select the type of information to be plotted (e.g drop down menu to select the dose, the risk event or age group of interest). The following visualizations will be created:

Coverage: bar charts with weekly number of administered doses extrapolated to the whole population

The total number of doses n_{ij} (for dose 1, dose 2 and dose 3) given during week i in age group j will be calculated from each database. Then, the total number of doses will be extrapolated to the whole population as follows:

$$Ntot_{ij} = (N_{ij}/pop_j)^{-1} n_{ij} = w_{ij}\, n_{ij},$$

where N_{ij} is the number of active subjects in the database at week i and age group j and where pop_j is the number of subjects in the total population of same age group j. The number of subjects in the total population will be obtained from the National Offices of Statistics.

Coverage: line plots with vaccination coverage (%) by year-month birth cohorts over time

For every birth-month cohort, we will evaluate the vaccination coverage by dose over time (i.e. at every start of the week/month depending on the refresh frequency of the database). The coverage at week i for birth cohort j will be calculated by dividing the number of vaccinated subjects n_{ij} by the total number of subjects still under follow-up at week i (N_{ij}), expressed as a percentage.

Benefits: line plots with pertussis incidence and 95% confidence intervals over time

The monthly pertussis incidence (/100.000 person-years) will be calculated as the number of pertussis events divided by the total person-time at risk multiplied with 100.000. Exact Poisson 95% confidence intervals will be calculated.

Risks: line plots with incidence of risk outcomes inside and outside the risk window and 95% confidence intervals over time using accrued data

The incidence rate (per unit person- years) for the risk outcomes within the pre-defined, outcome-specific risk windows will be estimated by week. In addition, the incidence rates will be estimated cumulatively over time, using all data accrued from the start of the study period until week i or

$$inc_I = \left(\sum_{i=1}^{I} n_i \Big/ \sum_{i=1}^{I} py_i \right) \times unit\ person\ years$$

where n_i is the number risk events of interest that happened during week i and where $py_i$is the amount of person-time (in years) within that week. Incidence rates outside the risk windows in vaccination comparable age groups will be calculated as well. Each time, exact Poisson 95% confidence intervals will be calculated.

Near real-time monitoring

Repeated data extracts

During the period of near real-time monitoring, several data extracts will be made (as described in Sections 5.6 and 6), which will be used to populate the dashboard. Each time, data will be extracted from start of the study (1st January 2014) till DLP. The analytical datasets will be locally created as described in Section 6.2.4. The files *POC12_OUT_date.txt* will be used to calculate the reporting delays and will stay local. Note that the file name will include the date of file generation.

New events

Every two subsequently generated data files *POC12_OUT_date.txt*. (data extract *i* and *i + 1*) will be compared. For data extract *i + 1*, only the records referring to newly added events (i.e. events present in data extract *i + 1*, but not present in data extract *i*) and their event dates will be retained.

Reporting delays

On the file with newly added events, the following data manipulations will be performed;

- Delay 1: calculate time between dates of newly added events and end of follow-up dates (in days)
- Delay 2: calculate time between end of follow-up dates and analytical dataset generation (in days)
- Delete variables gender, date of birth, cohortstart and eventdate

The new file is named *POC12_OUTdelay_dbname_date.txt*.

Table S15: Dummy table: analytical dataset with date information for analyzing reporting delays (in days)

Database	AnonId	EventId	Delay1	Delay2
UK_RCGP	P10000	aP_P1	100	7
UK_RCGP	P10000	aP_P2	15	7
UK_RCGP	P10000	FEVER	22	7
UK_RCGP	P10000	FEVER	14	7
UK_RCGP	P10000	ISR	33	7
UK_RCGP	P10001	aP_P2	22	7
UK_RCGP	P10001	FEVER	11	7
UK_RCGP	P10001	FCONVULS	15	7

The generation date will be created automatically. This file will be created locally and will then be transferred by the database custodians to the server used for the dashboard (see Section 5.6.2). The generated output will be as in Table S15. Then, the distribution of delays will be obtained by event type or group of events (events recorded at hospital vs. primary care; exposure events, health outcome events) and will be summarized using measures of central tendency (average or median, whichever is most appropriate) and measures of spread (standard deviation or inter-percentile ranges, whichever is most appropriate).

Finally, we will also calculate

- Delay 3: calculate time between analytical dataset generation and first display of the data within the dashboard

Summing the delays 1, 2 and 3, will give the time between event and first display of the data in the dashboard.

HANDLING OF MISSING DATA

POC1.2 will use the same approach to missing information regarding as POC1.

- Missing dose will be derived based on sequence in administration date, and depending on age at vaccination and country schedule as far as possible. Dose derivation algorithms may vary by country and will be documented.

QUALITY CHECK AND BENCHMARKING

All programs will be programmed per agreed coding standards. R version 3.4.0 will be used for extraction, transformation and statistical analyses. The POC1.2 programming file to transform the CDM files to the analytical datasets is a modification of the POC1 programming file to transform the CDM files to the analytical datasets used for the risk study. The POC1 programming files were double coded in R and SAS.

The POC1.2 results will be compared to the validated results from POC1.

Appendix 2: R scripts

STARTSCRIPT

```
################################################################################
####ADVANCE: POC 1.2 Startscript ####

### Date: 19/02/2018 - latest update 15/03/2018
### Author(s): Tom De Smedt & Kaatje Bollaerts
### Affiliation: P95
### Email: tom.desmedt@p-95.com
### Phone: +32472216565
###
################################################################################

#################
###Preparation###
#################

# Check whether additional packages exist, install them if not, and load them in
if (!("lubridate" %in% installed.packages())) install.packages("lubridate")
if (!("data.table" %in% installed.packages())) install.packages("data.table")
if (!("reshape2" %in% installed.packages())) install.packages("reshape2")
library("lubridate")
library("data.table")
library("reshape2")

# Variables to be set by the user
db_name <- "UK_RCGP"
folder_in = "~/ADVANCE/POC 2018/POC 1.2/201820"          # Fill in the path of the input
subfolder, a "results" folder will be created whitin the folder
folder_out  <- "~/ADVANCE/POC 2018/POC 1.2/201820/results/20180904"      # Fill in the
path of the output subfolder
scripts = "~/ADVANCE/POC 2018/POC 1.2/Scripts"                # Fill in the path of the
scripts subfolder
patients_file = "Patients201820.csv"              # Fill in the name of the patients CDM file,
include the extension (.csv or .txt)
```

```
vaccination_file = "Vaccinations201820.csv"          # Fill in the name of the vaccination
CDM file, include the extension (.csv or .txt)
events_file = "Events201820.csv"                     # Fill in the name of the events CDM file,
include the extension (.csv or .txt)
date_format = "YYYYmmdd"                             # Define the date format (options:
"YYYYmmdd", "YYYY-mm-dd", "YYYY/mm/dd")
date_extraction = "20190207"                         # Fill in the data of extraction of the dataset being
analysed

# Additional study variables
startstudy <- as.Date("2014-01-01")
endstudy <- as.Date("2020-12-31")
eventlist <- c("FCONVULS", "FEVER", "HHE", "SOMNOL", "PCRYING", "PERT")
rwdlist <- c(3,3,2,2,1,NA)
rwdlist2 <- c(0,1,2,3)
seqwk <- as.Date(seq(startstudy, endstudy, by = 'week'))
seqwk <- seqwk[1: (length(seqwk)-1)]                 # date start of every week, from start
study till last week before end study
n_wk <- length(seqwk)                                # total number of study weeks
wk_id <- seq(1, n_wk, 1)
wk_id <- data.table(wk_id)

# Additional function to replace missing values
f_dowle3 = function(DT) {
  for (j in seq_len(ncol(DT)))
    set(DT,which(is.na(DT[[j]])),j,0)
}

# Load in data transfromation scripts
setwd(scripts)
source("f_import_events_population.R")
source("f_import_exposure.R")
source("f_transformation_poc12.R")
source("f_dtrans_ndose.R")
source("f_dtrans_denom.R")
source("f_dtrans_nrisk.R")
source("f_dtrans_pyrisk.R")

# Phase 2 Data Transformation - From CDM to Analytical Dataset
# Import & cleaning of events CDM-file
events.import <- events.import.function(folder_in, events_file, date_format)
# Import & cleaning of patients CDM-file
```

```
population.import <- population.import.function(folder_in, patients_file, date_format)
# Import & cleaning of vaccinations CDM-file
vaccination.import <- vaccination.import.function(folder_in, patients_file, vaccination_file,
date_format)
# Merge and subsetting of imported and cleaned events, population and vaccinations file
poc12_results <- poc12.function(folder_in, events.import, population.import,
vaccination.import, dbname = db_name)

# Phase 3 Data Transformation - From Analytical Dataset to Aggregated Datatables
# From long to wide Analytical Dataset file
poc12_temp <- poc12_results
print(str(poc12_results))
poc12_temp$EventDate <- as.character(poc12_temp$EventDate)
poc12_temp <- poc12_temp[poc12_temp$EventId %in% c("P1", "P2", "P3"),]
poc12_temp <- data.table(poc12_temp)
poc12_wide <-  dcast(poc12_temp,
                AnonId + DoB + startfu + endfu ~ EventId, value.var = "EventDate")
poc12_wide$date.P1 <- as.Date(poc12_wide$P1)
poc12_wide$date.P2 <- as.Date(poc12_wide$P2)
poc12_wide$date.P3 <- as.Date(poc12_wide$P3)
nameWide <- names(poc12_wide)
nameWide <- gsub(pattern = "EventDate", replacement = "date", x = nameWide)
names(poc12_wide) <- nameWide

poc12_wide <- data.frame(poc12_wide)
poc12_results <- data.frame(poc12_results)


# Calculate Number of Doses per Agegroup and Calendar Week Datatable
f_dtrans_ndose(poc12_wide, folder_out, paste0("POC12_ndose_", db_name,"_d"), startstudy)

# Calculate YearMonth Birthcohort Coverage Datatable
f_dtrans_denom(poc12_wide, folder_out, paste0("POC12_CovbyYMBirth_", db_name,"_"),
startstudy, endstudy)

# Calculate Number of Events (for events in eventlist) per Agegroup and Calendar Week
f_dtrans_nrisk(poc12_results, poc12_wide, folder_out, paste0("POC12_nevent_", db_name,
"_"), eventlist, rwdlist, startstudy, wk_id)

# Calculate Persontime In- and Outside of the Riskwindows defined in rwdlist2
f_dtrans_pyrisk(poc12_wide, folder_out, paste0("POC12_py_", db_name, "_"), rwdlist2,
startstudy, endstudy)
```

```
# Wrap all Aggregated Datatables in One Zipfile
files2zip <- dir(folder_out, full.names = TRUE)
print(files2zip)
date_extraction2 <- file.info(paste0(folder_in, "/", patients_file))$ctime
date_extraction2 <- gsub(pattern = ":", x = date_extraction2, replacement = "")
date_extraction2 <- gsub(pattern = " ", x = date_extraction2, replacement = "-")
zip(zipfile = paste0('POC12_Results_', db_name, '_', date_extraction, date_extraction2, '_',
Sys.Date(), '.zip'), files = files2zip)
```

SEE STATISTICAL ANALYSIS PLAN
Phase 2: Transformation of CDM data files into the analytical dataset

---

MERGE VACCINATION¸ EVENT and PATIENT files and DEFINE COHORT

```
################################################################################
#### WP5 – ADVANCE f_transformation_poc12 ####

### Date: 12/01/2018
### Author(s): Tom De Smedt & Kaatje Bollaerts
### Affiliation: P95
### Email: tom.desmedt@p-95.com
################################################################################


#########################
###### Risks ###########
#########################

####Create the function

#dbname <- "RCGP"
#folder <- folder_out
#events.import = events.import2
#population.import = population.import2
```

```
#vaccination.import = vaccination.import2


poc12.function=function(folder, events.import, population.import, vaccination.import,
dbname){

setwd(folder)
output_folder=paste(folder,"results_POC12",sep="/")
dir.create(output_folder)


cat('.....\n')
cat('R code for the POC 1.2\n')
cat('.....\n')



#####################################
######Getting the input data########
#####################################
cat('\n')
cat('Import module should be ran first\n')
cat('This will import the necessary input files into the workspace\n')
cat('\n')

cleanvaccinations <- vaccination.import
events <- events.import
population <- population.import


#####################################
##Transformation of the input data##
#####################################

cat('\n')
cat('Transformation of the input data\n')
cat('\n')

### Merging cleanvaccinations, events-file and patient-file
cat('Preparing population file, exposure file and events file (eventtype="FEVER",
"SOMNOL", "PCRYING", "CONVULS", "HHE", "PERTUSSIS) for merge')
cat('\n')
```

```
##---------------------------------
## 1. Population
#Startdate and enddate of the study
cat('Population: adding variables, introduce rules for the follow-up time')
cat('\n')
cat('Setting start (01-01-2014) and end (31-12-2020) date of the study period\n')
cat('\n')

start.study <- as.Date("20140101","%Y%m%d")
end.study <- as.Date("20201231","%Y%m%d")
#population <- population[population$birthdate>=start.study,]
#Add variables
population$startfu  <- pmax(population$startdate, start.study, population$birthdate)
population$endfu    <- pmin(population$enddate, end.study, population$birthdate+2191)
population$age.at.start     <- population$startfu - population$birthdate
population$age.at.end       <- population$endfu - population$birthdate
persontime                  <- population$endfu - population$startfu + 1
#Select: positive follow-up
cat('Removing persons from population file with negative follow-up')
cat('\n')
population    <- population[!is.na(persontime),]
population    <- population[persontime > 0,]
# keep relevant variables only
print(dim(population))
population.s  <- subset(population, select = c("patientid", "birthdate", "gender", "startfu",
"endfu"))


##2. Vaccinations (exposure)
cat('Vaccinations: Cleaned...')
cat('\n')
# print("TEST MERGE POP VAC1")

# # Removing those persons with negative follow up time
# cat('Removing persons from vaccination file with negative follow-up age at start < age at
end')
# cat('\n')
print(str(cleanvaccinations))
cleanvaccinations.s  <- subset(cleanvaccinations, select = c('patientid', 'date.P1', 'date.P2',
'date.P3', 'date.B1', 'vactypecombined.P1', 'vactypecombined.P2', 'vactypecombined.P3',
'vactypecombined.B1'))  # endfu
```

```
print("Clean Vacc after subset")


## 3. Risk events
cat('Events: Subsetting to "FEVER", "SOMNOL", "PCRYING", "CONVULS", "HHE",
"PERT" cases\n')
cat('\n')
# selecting risk events
events.outc <- events[events$eventtype %in% c("FEVER", "SOMNOL", "PCRYING",
"CONVULS", "HHE", "FCONVULS", "PERT"),]
patientid   <- events.outc$patientid
eventtype   <- events.outc$eventtype
date        <- events.outc$date
events.outc <- data.table(patientid, eventtype, date)

# events are recurrent if they are at least 7 days apart, for 1 event with multiple records keep
only the first date

setkey(events.outc, patientid, eventtype)
#events.outc[order(patientid, eventtype, date)]
setorder(events.outc, patientid, eventtype, date)
events.outc[, diff := c(NA,diff(date)),by = list(patientid, eventtype)]

events.outc <- events.outc[(is.na(diff) | diff > 7)]

# reshape wide, keeping one line with risk event dates for every patient
events.outc[ , `:=`(cnt = 1:.N ) , by = list(patientid, eventtype)]
setkey(events.outc, cnt)
eventtype.cnt <- paste0(events.outc$eventtype, as.character(events.outc$cnt))
patientid     <- events.outc$patientid
date          <- events.outc$date

events.outc   <- data.frame(patientid, eventtype.cnt, date)
events.outc   <- reshape(events.outc, idvar = "patientid", timevar = "eventtype.cnt", direction
= "wide")


##---------------------------------
## merge population, exposure and risk events files
cat('Merging the population, vaccination and risk events file')
cat('\n')
```

```
#rm(list = ls()[!ls() %in% c("population.s","cleanvaccinations.s", "events.outc",
"output_folder", "dbname")])
gc()
DAT.poc12.merge <- merge(merge(population.s, cleanvaccinations.s, by = 'patientid', all.x =
T), events.outc, by = 'patientid', all.x = T)
print('AFTER MERGE')
print(dim(DAT.poc12.merge))


#dat.merge2 <- DAT.poc12.merge


cat('writing "DAT.poc12.merge" to output folder \n')
write.table(DAT.poc12.merge, paste(output_folder, "DAT_risk_merge.txt", sep = "/"),sep =
",", row.names = F)
cat('..... \n')
cat('\n')

##############################################################################
##
##Transformation of input files into analytical datasets that can be shared##
##############################################################################
##

cat('Transformation of input files into analytical datasets\n')
cat('\n')


### Exclude exposure and risk events that happended outside the FU time defined as [startfu -
endfu]
cols.date1 <- c('date.P1', 'date.P2', 'date.P3', 'date.B1')
cols.date2 <- c(names(events.outc)[2:dim(events.outc)[2]])
out <- matrix(FALSE, nrow = dim(DAT.poc12.merge)[1], ncol = dim(DAT.poc12.merge)[2])
colDates <- vector('numeric')
for(i in 1:dim(DAT.poc12.merge)[2]){
#   if(class(DAT.poc12.merge[,i]) == "Date" & names(DAT.poc12.merge)[i] %in% cols.date1
) {
#     colDates <- c(colDates, i)
#     # print(sum(is.na(DAT.poc12.merge[,i])))
#     # print(sum(is.na(DAT.poc12.merge[,i])))
#     out[,i] <- DAT.poc12.merge[,i] < (DAT.poc12.merge$startfu-7) | DAT.poc12.merge[,i] >
DAT.poc12.merge$endfu | DAT.poc12.merge[,i] < as.Date("2014-01-01")
```

```
#   }
  if(class(DAT.poc12.merge[,i]) == "Date" & names(DAT.poc12.merge)[i] %in% cols.date2)
{
    colDates <- c(colDates, i)
    # print(sum(is.na(DAT.poc12.merge[,i])))
    # print(sum(is.na(DAT.poc12.merge[,i])))
    out[,i] <- DAT.poc12.merge[,i] < DAT.poc12.merge$startfu | DAT.poc12.merge[,i] >
DAT.poc12.merge$endfu | DAT.poc12.merge[,i] < as.Date("2014-01-01")
  }
}
# out <- (DAT.poc12.merge  < DAT.poc12.merge$startfu | DAT.poc12.merge  >
DAT.poc12.merge$endfu)
# cat(str(out))
# idm <- which(names(DAT.poc12.merge) != cols.date)
# cat(str(idm))
# out[,idm] <- FALSE
print(dim(DAT.poc12.merge))
DAT.poc12.merge[which(out==TRUE, arr.ind=TRUE)] <- NA
print(dim(DAT.poc12.merge))

#### Reshape from wide to long format
rm(list = ls()[!ls() %in% c("DAT.poc12.merge", "output_folder", "dbname", "events.outc")])
gc()
DAT.poc12.merge   <- melt(DAT.poc12.merge, id.vars = c("patientid", "birthdate", "gender",
"startfu", "endfu"), measure.vars=c("startfu", "endfu", "date.P1", "date.P2",
"date.P3","date.B1", names(events.outc)[2:dim(events.outc)[2]]))  # , measure.vars =
names(DAT.poc12.merge)[colDates]
# DAT.poc12.merge   <- reshape(data = DAT.poc12.merge, idvar = c("patientid", "birthdate",
"gender", "startfu"), timevar = "event", v.names = "date",
#                              varying = names(DAT.poc12.merge)[colDates], times =
names(DAT.poc12.merge)[colDates], direction = "long")
print("TEST CASE")
# print(DAT.poc12.merge[DAT.poc12.merge$patientid == 'a6643_05eI', ])
print(dim(DAT.poc12.merge))
DAT.poc12.merge <- DAT.poc12.merge[!is.na(DAT.poc12.merge$value),]
#DAT.poc12.merge$Anonid <- paste0(dbname ,"_", match(DAT.poc12.merge$patientid,
unique(DAT.poc12.merge$patientid)))
DAT.poc12.merge$Anonid <- DAT.poc12.merge$patientid

DAT.poc12.merge$gender <- as.character(DAT.poc12.merge$gender)
DAT.poc12.merge$variable <- as.character(DAT.poc12.merge$variable)
print(str(DAT.poc12.merge))
```

```
tempVar <- strsplit(DAT.poc12.merge$variable, ".", fixed = T)
tempVar <- unlist(lapply(tempVar, tail, n = 1L))
AE <- c("FEVER", "SOMNOL", "PCRYING", "CONVULS", "HHE", "FCONVULS",
"PERT")
tempAE <- rep(NA, 1000 * length(AE))
teller <- 1
for(i in 1:length(AE)){
  for(j in 1:1000) {
    tempAE[teller] <- paste0(AE[i], j)
    teller <- teller + 1
  }
}
for(i in 1:length(tempVar)) {
  if(tempVar[i] %in% tempAE) {
    tempVar[i] <- gsub("[[:digit:]]","", tempVar[i])
  }
}

DAT.poc12.merge$variable <- tempVar

PatKey <- data.frame(PatID = DAT.poc12.merge$Anonid, AnonID =
DAT.poc12.merge$patientid)
PatKey <- PatKey[!duplicated(PatKey$AnonID),]
DAT.poc12.merge <- subset(DAT.poc12.merge, select = -patientid)

cat('writing analytical dataset POC12 "DAT.poc12.merge" to output folder \n')
write.table(DAT.poc12.merge, paste(output_folder,paste0("POC12_OUT_", Sys.Date(),
".txt"),sep="/"),sep=",",row.names=F)
cat('writing patient id anonid key dataset \n')
write.table(PatKey, paste(output_folder,paste0("PatIdKey_", Sys.Date(),
".txt"),sep="/"),sep=",",row.names=F)
cat('..... \n')
cat('\n')
names(DAT.poc12.merge) <- c("DoB", "gender", "startfu", "endfu", "EventId", "EventDate",
"AnonId")
return(DAT.poc12.merge)
}
```

IMPORT VACCINATION DATA AND CLEAN

```
##############################################################################
```

```
######ADVANCE: pertussis POC 1.2 f_import_exposure #######
(Script by Toon Braeye – re-used from POC1)
#########################################################################


###################
##Import the data##
###################

cat('\n')
cat('Importing the data\n')
cat('\n')

cat('\n')
cat('The import module will create 3 functions; \n')
cat('events.import.function, population.import.function and vaccination.import.function,\n')
cat('for importing and cleaning data.\n')
cat('This module should be ran first and the workspace should not be cleaned afterwards\n')
cat('\n')


###############
#-Vaccination-#
###############


vaccination.import.function = function(folder,patients_file ,vaccination_file,date_format){
####Importing and Cleaning the vaccination_file

 #checking parameters
 if (length(folder)==0){
  throw("No work space specified")
 }else{
  setwd(folder)
 }

 if (length(date_format)==0){
  warning("No date format specified try YYYYmmdd")
  date_format="%Y%m%d"
 }
```

```
  if (date_format=="YYYYmmdd"){
    date_format="%Y%m%d"
  }


  if (date_format=="YYYY-mm-dd"){
    date_format="%Y-%m-%d"
  }

  if (date_format=="YYYY/mm/dd"){
    date_format="%Y/%m/%d"
  }


  #Create output folder
  output_folder=paste(folder,"results_risk_doserecorded",sep="/")
  dir.create(output_folder)#,showwarnings=T,recursive=T)
  if (!dir.exists(output_folder)){                    #Unclear why this is here, folder was
just created?
    output_folder=paste(folder,"/", sep="")
  }
  ##READ IN DATA
  cat('Read in data')
  #Patientfile and Vaccinations files
  patients=read.table(patients_file,header=T,sep=",", fill=T, na.strings = c("","NA"))
#Added fill=True
  vaccination=read.table(vaccination_file,header=T,sep=",", fill=T, na.strings = c("","NA"))
  colnames(patients)=tolower(colnames(patients))
  colnames(vaccination)=tolower(colnames(vaccination))

  vaccination$doserecorded<-as.character(vaccination$doserecorded)
  vaccination$doserecorded<-ifelse(vaccination$doserecorded=="D1", "P1",
ifelse(vaccination$doserecorded=="D2", "P2", ifelse(vaccination$doserecorded=="D3", "P3",
vaccination$doserecorded)))
  vaccination$dosederived<-as.character(vaccination$dosederived)
  vaccination$dosederived<-ifelse(vaccination$dosederived=="D1", "P1",
ifelse(vaccination$dosederived=="D2", "P2", ifelse(vaccination$dosederived=="D3", "P3",
vaccination$dosederived)))

  #Remove incomplete data
  cat('Records with absent patientid will be removed\n')
```

```
  cat(sprintf("patients: patientid is not specified: %d\n",sum(is.na(patients$patientid))),
"Removed from database\n")
  patients<-patients[!is.na(patients$patientid),]
  cat(sprintf("vaccination: patientid is not specified: %d\n",sum(is.na(vaccination$patientid))),
"Removed from database")
  vaccination<-vaccination[!is.na(vaccination$patientid),]
  cat('duplicates from the patients-file will be removed. N=',
sum(duplicated(patients$patientid)))
  patients=patients[!duplicated(patients$patientid),]     #Deleting duplicated patient-records
before the merge


  #Set dates in date format
  cat("correct date formats if necessary...\n")
  patients$birthdate=as.Date(as.character(patients$birthdate),format= date_format)
  patients$startdate=as.Date(as.character(patients$startdate),format= date_format)
  patients$enddate=as.Date(as.character(patients$enddate), format= date_format)
  vaccination$date=as.Date(as.character(vaccination$date),format=date_format)

  #CHECKING IF THE STRUCTURE IS CORRECTLY SPECIFIED
  #str(patients)
  #str(vaccination)

  #checking on the patient ids

number_of_patients_in_vaccinationfile_but_not_in_patientsfile=length(unique(vaccination$p
atientid))-length(intersect(patients$patientid,vaccination$patientid))

number_of_patients_in_patientsfile_but_not_in_vaccinationfile=length(unique(patients$patie
ntid))-length(intersect(patients$patientid,vaccination$patientid))

  cat("Number_of_patients_in_vaccinationfile_but_not_in_patientsfile:",
number_of_patients_in_vaccinationfile_but_not_in_patientsfile,"\n")
  cat("Number_of_patients_in_patientsfile_but_not_in_vaccinationfile:",
number_of_patients_in_patientsfile_but_not_in_vaccinationfile,"\n")

  #removing those Patientids from vaccinations file  that are not in the Patients file
  #cat('Records of patients in vaccination but not in patients_file will be removed\n')
  #vaccination=vaccination[(vaccination$patientid %in% patients$patientid),]
  #inlude only pertussis vaccinations
  cat('vaccination file is limited to pertussis vaccinations (uPE, aPE and wPE)\n')
  vaccination$vactype <- as.character(vaccination$vactype)
```

```
  vaccination$vactype[grepl("aPE", vaccination$vactype, fixed = TRUE)]<-"aPE"
  vaccination$vactype[grepl("wPE", vaccination$vactype, fixed = TRUE)]<-"wPE"
  vaccination$vactype[grepl("uPE", vaccination$vactype, fixed = TRUE)]<-"uPE"
  print(str(vaccination))
  vaccination=vaccination[(vaccination$vactype %in% c('aPE')),]
  print(str(vaccination))
  cat('Records left in vaccination file:', nrow(vaccination),'\n')
  cat('Persons left in vaccination file:', length(unique(vaccination$patientid)),'\n')

  #deleting duplicated patient-records and merge patients and vaccination file
  cat("Adding neccsessary patient details in the vaccination file (merge).....\n") #TO BE
ABLE TO DO THE CHECKS OF startdate and date of vaccination
  vaccination=merge(vaccination,patients,by="patientid",all.X=T ,all.y=F)
  #
write.table(vaccination,paste(output_folder,"vaccinationFULL.txt",sep="/"),sep=",",row.name
s=F)
  #QUALITY CHECKS
  cat("Some data checks....\n")
  cat(".............\n")
  cat(sprintf("Start date occurring before birthdate:
%d\n",length(which(patients$startdate<patients$birthdate))))
  cat(sprintf("End date occurring before birthdate:
%d\n",length(which(patients$enddate<patients$birthdate))))
  cat(sprintf("End date occurring before startdate:
%d\n",length(which(patients$enddate<patients$startdate))))
  cat(sprintf("Vaccination date occurring before start of fu date:
%d\n",length(which(vaccination$date<vaccination$birthdate))))
  cat(sprintf("Vaccination date occurring after end of fu date:
%d\n",length(which(vaccination$date>vaccination$enddate))))
  cat(".............\n")

  ####creating the COHORT AND POPULATION DEFINITIONS
  ####population definition

  cat("Creating the Source population definition\n")
  cat('populationstart is set at 01/01/2014\n')
  cat('populationend is set at min(31/12/2020, patient end of follow-up, 6 years of age)\n')
  cat('the follow-up of a patient should be >0\n')


  population=data.frame(patients) ##population file is just an extension of patients file and 2
more columns(population start and population end)
```

```
colnames(population)=tolower(colnames(population))

population$populationstart = pmax(population$startdate,(as.Date("01/01/2014", format =
"%d/%m/%Y")))
sixyearenddate <- as.POSIXlt(population$birthdate)
sixyearenddate$year <- sixyearenddate$year+6
population$populationend= pmin((as.Date("31/12/2020", format = "%d/%m/%Y")),
population$enddate, as.Date(sixyearenddate))   #Added max age of 6 years old
#at least one day of follow-up
population<-population[population$populationend-population$populationstart+1>0,]

# write.table(population,paste(output_folder,"population.txt",sep="/"),sep=",",row.names=F)

cat("..............\n")


######COHORT DEFINITION##########
cat("Creating the  Cohort definition\n")
cohort=data.frame(population)
colnames(cohort)[colnames(cohort)=="populationstart"]="cohortstart"
colnames(cohort)[colnames(cohort)=="populationend"]="cohortend"


#write.table(cohort,file="cohort.txt",sep=",")
# write.table(cohort,paste(output_folder,"cohort.txt",sep="/"),sep=",",row.names=F)
cat("..............\n")

#Frequency tables
#
write.table(table(vaccination$atc,useNA="always"),paste(output_folder,"atc_frequencies.txt",
sep="/"),sep=",",row.names=F,col.names=c("ATC","Freq"))
#
write.table(table(vaccination$vactype,useNA="always"),paste(output_folder,"vactype.txt",sep
="/"),sep=",",row.names=F,col.names=c("Vactype","Freq"))
#
write.table(table(vaccination$dosederived,useNA="always"),paste(output_folder,"dosederive
d.txt",sep="/"),sep=",",row.names=F,col.names=c("dosederived","Freq"))
#
write.table(table(vaccination$doserecorded,useNA="always"),paste(output_folder,"doserecor
ded.txt",sep="/"),sep=",",row.names=F,col.names=c("doserecorded","Freq"))
```

```
####Creating dosesequence, dosecombined and vactypecombined
cat('Creating dosesequence')
cat('.....\n')

#initial variables in the Vaccination file
keeps=c("patientid", "date","brand","atc","vactype","doserecorded","dosederived")
###clean vaccination = an extention of vaccination file
cleanvaccinations=vaccination[,(names(vaccination)%in% keeps)]

##2. DoseCombined
cat('Creating dosecombined')
cat('.....\n')
#DoseRecorded currently has the priority, this should however be programmed flexible
#Remove the remaining empty ones (not both doserecorded and dosederived)
cat('Number of times doserecorded is empty:', sum(is.na(cleanvaccinations$doserecorded)),
'\n')
cat('Number of times dosederived is empty:', sum(is.na(cleanvaccinations$dosederived)),
'\n')
cat('Number of times dosederived and doserecorded are empty:',
sum(is.na(cleanvaccinations$dosederived) & is.na(cleanvaccinations$doserecorded)), '\n')
cat(sprintf("Number of times discordance between 'dosederived' and 'doserecorded' is
present (not taking empty into account): %d\n",
          sum(cleanvaccinations[(!is.na(cleanvaccinations$doserecorded) &
!is.na(cleanvaccinations$dosederived)),]$doserecorded !=
                cleanvaccinations[(!is.na(cleanvaccinations$doserecorded) &
!is.na(cleanvaccinations$dosederived)),]$dosederived)), "Currently doseRecorded is used for
further analysis\n")
cleanvaccinations$dosecombined<-ifelse(!is.na(cleanvaccinations$doserecorded),
as.character(cleanvaccinations$doserecorded), as.character(cleanvaccinations$dosederived))

#Prevent the ordering problem (P3 and B1 - problem)
cleanvaccinations$dosecombined<-as.character(cleanvaccinations$dosecombined)
cleanvaccinations$dosederived<-as.character(cleanvaccinations$dosederived)
cleanvaccinations$doserecorded<-as.character(cleanvaccinations$doserecorded)

cleanvaccinations$dosecombined[cleanvaccinations$dosecombined=="P1"]<-"aP1"
cleanvaccinations$dosecombined[cleanvaccinations$dosecombined=="P2"]<-"aP2"
cleanvaccinations$dosecombined[cleanvaccinations$dosecombined=="P3"]<-"aP3"

cleanvaccinations$doserecorded[cleanvaccinations$doserecorded=="P1"]<-"aP1"
cleanvaccinations$doserecorded[cleanvaccinations$doserecorded=="P2"]<-"aP2"
cleanvaccinations$doserecorded[cleanvaccinations$doserecorded=="P3"]<-"aP3"
```

```
  cleanvaccinations$dosederived[cleanvaccinations$dosederived=="P1"]<-"aP1"
  cleanvaccinations$dosederived[cleanvaccinations$dosederived=="P2"]<-"aP2"
  cleanvaccinations$dosederived[cleanvaccinations$dosederived=="P3"]<-"aP3"

  cleanvaccinations=cleanvaccinations[order(cleanvaccinations$patientid,
cleanvaccinations$date, cleanvaccinations$doserecorded, cleanvaccinations$dosederived),]

  ##1. DoseSequence (using the package data.table would be much faster)
  print(str(cleanvaccinations))
  cleanvaccinations<-transform(cleanvaccinations,
dosesequence.number=as.numeric(ave(as.numeric(date), patientid, FUN=function(x) rank(x,
ties.method="first"))))
  print(str(cleanvaccinations))
  cleanvaccinations$dosesequence<-NA
  cleanvaccinations[cleanvaccinations$dosesequence.number==1,]$dosesequence<-"P1"
  cleanvaccinations[cleanvaccinations$dosesequence.number==2,]$dosesequence<-'P2'
  cleanvaccinations[cleanvaccinations$dosesequence.number==3,]$dosesequence<-'P3'
  cleanvaccinations[cleanvaccinations$dosesequence.number==4,]$dosesequence<-'B1'
  cleanvaccinations[cleanvaccinations$dosesequence.number==5,]$dosesequence<-'B2'

  ##3. VactypeCombined (vactype derived currenlty not set)
  cat('Creating vactypecombined')
  cat('.....\n')
  cleanvaccinations$vactypecombined<-ifelse(!is.na(cleanvaccinations$vactype),
as.character(cleanvaccinations$vactype), as.character(cleanvaccinations$vactypederived))
  cat('\n')

  ####Vaccination Definition module

  ###adding  some variables to the vaccination file TO CREATE THE CLEAN
VACCINATIONS FILE
  cat('Vaccination Cleaning (SAP 7.5.3.2)\n')
  cat('.....\n')

  ##1. Missing date
  cat('Checking for missing dates of vaccination\n')
  cat('Number vaccination records that do not have a vaccination date:',
sum(is.na(cleanvaccinations$date)), "\n")
  cat('Removed from the database\n')
  missing.date<-cleanvaccinations[is.na(cleanvaccinations$date),]
  cleanvaccinations<-cleanvaccinations[!is.na(cleanvaccinations$date),]
```

```
  cat('\n')

  ##2. Missing dose
  cat('Checking for missing dose in dosecombined\n')
  cat('Number vaccination records that do not have a dosecombined:',
sum(is.na(cleanvaccinations$dosecombined)), "\n")
  cat('Removed from the database\n')
  missing.dose<-cleanvaccinations[is.na(cleanvaccinations$dosecombined),]
  cleanvaccinations<-cleanvaccinations[!is.na(cleanvaccinations$dosecombined),]
  cat('\n')

  ##2.1. Minimum necessary distance rule
  ## Minimum necessary distance, currently set to -1 days
  minimumnecessarydistance<- 1
  cat('Checking the distances between the vaccination dates\n')
  cat('Minimum necessary distance currently set to -1\n')

  cleanvaccinations$diffdates<-as.numeric(cleanvaccinations$date-(c(NA, cleanvaccinations[-
nrow(cleanvaccinations),]$date)))
  mnd.vaccination<-cleanvaccinations[(duplicated(cleanvaccinations$patientid) &
cleanvaccinations$diffdates<minimumnecessarydistance),]
  cleanvaccinations<-cleanvaccinations[!(duplicated(cleanvaccinations$patientid) &
cleanvaccinations$diffdates<minimumnecessarydistance),]

  #cleanvaccinations<-cleanvaccinations[!(as.factor(cleanvaccinations$patientid) %in%
as.factor(mnd.vaccination$patientid)),]
  cat(sprintf("Number of records in which the mimumun necessary distance is not respected:
%d\n",nrow(mnd.vaccination)), "Removed from database\n")
  cat(sprintf("Persons for who the mimumun necessary distance is not respected:
%d\n",length(unique(mnd.vaccination$patientid[!is.na(mnd.vaccination$patientid)]))))
  cat('\n')


  ##3. Patient who received uPE  - Optional (don't take booster into account):
  cat('Checking for patients who received one or more unknown vactypes (uPE)\n')
  cat('Booster vaccinations are not taken into account \n')
  cat('Number of vaccination records with uPE:', sum(cleanvaccinations$vactype=="uPE"),
"\n")
  cat('Number of patient who received uPE:',
length(unique(cleanvaccinations[cleanvaccinations$vactype=="uPE",]$patientid)), "\n")
```

```
  uPE.vaccinations<-
as.data.frame(cbind(as.character(cleanvaccinations[(cleanvaccinations$vactype=="uPE"),]$pa
tientid),
as.numeric(cleanvaccinations[(cleanvaccinations$vactype=="uPE"),]$dosesequence.number))
)
  cleanvaccinations<-merge(cleanvaccinations, uPE.vaccinations, by.x='patientid', by.y='V1',
all.x=T)
  patients.with.uPE<-
unique(as.character(cleanvaccinations[(cleanvaccinations$vactype=="uPE"),]$patientid))
  lost.due.to.uPE<-
cleanvaccinations[cleanvaccinations$dosesequence.number>=as.numeric(cleanvaccinations$
V2) & !is.na(cleanvaccinations$dosesequence.number>=as.numeric(cleanvaccinations$V2)),]
  cleanvaccinations<-
cleanvaccinations[!(cleanvaccinations$dosesequence.number>=as.numeric(cleanvaccinations
$V2)&
!is.na(cleanvaccinations$dosesequence.number>=as.numeric(cleanvaccinations$V2))),]
  cleanvaccinations<-subset(cleanvaccinations, select=-c(V2))
  cat('The sum of uPE records and records following uPE vaccination (for the same person):',
nrow(lost.due.to.uPE), "\n")
  cat('Removed uPE vaccination record and later records from patients with at least one uPE in
(P1/P2/P3) from the database\n')
  cat('\n')

  ##4. Patient who received a mixture of vactypes (e.g. both aPE and wPE) - Optional (don't
take booster into account):
  cat('Checking for patients who received a mixture of vactypes (eg aPE and wPE)\n')
  cat('Booster vaccinations are not taken into account \n')
  cleanvaccinations<-cleanvaccinations[order(cleanvaccinations$patientid,
cleanvaccinations$dosesequence.number),]
  patientid.vactype<-cleanvaccinations[!duplicated(cleanvaccinations[c('patientid',
'vactype')]),]
  patientid.vactype<-patientid.vactype[order(patientid.vactype$patientid,
patientid.vactype$date),]
  mixture.of.vactypes<-patientid.vactype[duplicated(patientid.vactype$patientid), c('patientid',
'dosesequence.number')]
  colnames(mixture.of.vactypes)[2]<-'dosesequence.mixture'
  mixture.of.vactypes<-mixture.of.vactypes[!duplicated(mixture.of.vactypes$patientid),]
  cleanvaccinations<-merge(cleanvaccinations,
as.data.frame(cbind(as.character(mixture.of.vactypes$patientid),
as.numeric(mixture.of.vactypes$dosesequence.mixture))), by.x='patientid', by.y='V1',
all.x=T)
```

```
  records.lost.due.to.mixture.of.vactypes<-
cleanvaccinations[(cleanvaccinations$dosesequence.number>=as.numeric(as.character(cleanv
accinations$V2)) & !is.na(cleanvaccinations$V2)),]
  cleanvaccinations<-
cleanvaccinations[!(cleanvaccinations$dosesequence.number>=as.numeric(as.character(clean
vaccinations$V2)) & !is.na(cleanvaccinations$V2)),]
  cleanvaccinations<-subset(cleanvaccinations, select=-c(V2))
  cat('Number of patient who received a mixture of vactypes (e.g. both aPE and wPE):',
length(unique(mixture.of.vactypes$patientid)), "\n")
  cat('Number of records lost due to a mixture of vactypes (e.g. both aPE and wPE):',
nrow(records.lost.due.to.mixture.of.vactypes), "\n")
  cat('The record of the -switch- and later records of that patient are removed from the
database\n')
  cat('\n')

  ##5. Remove duplicates
  cat('Checking for duplicates (dates and dosecombined)\n')
  cleanvaccinations<-cleanvaccinations[order(cleanvaccinations$patientid,
cleanvaccinations$date, cleanvaccinations$dosesequence.number),]
  duplicate.vaccination<-cleanvaccinations[duplicated(cleanvaccinations[c('patientid', 'date',
'dosecombined')]),]
  cleanvaccinations<-cleanvaccinations[!duplicated(cleanvaccinations[c('patientid', 'date',
'dosecombined')]),]
  cat(sprintf("Same vaccine, same person, same date:
%d\n",length(unique(duplicate.vaccination$patientid))), "Removed from database, earliest
record kept\n")

  duplicate.dosevaccination<-cleanvaccinations[duplicated(cleanvaccinations[c('patientid',
'dosecombined')]),]
  cleanvaccinations<-cleanvaccinations[!duplicated(cleanvaccinations[c('patientid',
'dosecombined')]),]
  cat(sprintf("Same vaccine, same person:
%d\n",length(unique(duplicate.dosevaccination$patientid))), "Removed from database,
earliest record kept\n")
  cat('\n')


  ##7. Order of the vaccination dates
  cat('Checking the order of the vaccination dates\n')
  cat('Changing the dosecombined if dates not consecutive \n')
```

```
  cleanvaccinations<-cleanvaccinations[order(cleanvaccinations$patientid,
cleanvaccinations$date),]
  cleanvaccinations.patient.dose<-cleanvaccinations[c('patientid', 'dosecombined')]
  cleanvaccinations.patient.dose<-
cleanvaccinations.patient.dose[order(cleanvaccinations.patient.dose$patientid,
cleanvaccinations.patient.dose$dosecombined),]

  order.vaccination<-cbind(cleanvaccinations,
cleanvaccinations.patient.dose[c('dosecombined')])
  colnames(order.vaccination)[13]<-"dosecombined2"
  order.vaccination<-
order.vaccination[order.vaccination$dosecombined!=order.vaccination$dosecombined2,]

  cleanvaccinations<-subset(cleanvaccinations, select=-dosecombined)
  cleanvaccinations<-cbind(cleanvaccinations,
cleanvaccinations.patient.dose[c('dosecombined')])

  cleanvaccinations$dosecombined[cleanvaccinations$dosecombined=="aP1"]<-"P1"
  cleanvaccinations$dosecombined[cleanvaccinations$dosecombined=="aP2"]<-"P2"
  cleanvaccinations$dosecombined[cleanvaccinations$dosecombined=="aP3"]<-"P3"

  cleanvaccinations$doserecorded[cleanvaccinations$doserecorded=="aP1"]<-"P1"
  cleanvaccinations$doserecorded[cleanvaccinations$doserecorded=="aP2"]<-"P2"
  cleanvaccinations$doserecorded[cleanvaccinations$doserecorded=="aP3"]<-"P3"

  cleanvaccinations$dosederived[cleanvaccinations$dosederived=="aP1"]<-"P1"
  cleanvaccinations$dosederived[cleanvaccinations$dosederived=="aP2"]<-"P2"
  cleanvaccinations$dosederived[cleanvaccinations$dosederived=="aP3"]<-"P3"

  cat(sprintf("Number of vaccination records with consecutive doses but not consecutive
dates: %d\n",nrow(order.vaccination)), "Dosecombined changed\n")
  cat(sprintf("Number of persons affected by non-consecutive dates:
%d\n",length(unique(order.vaccination$patientid[!is.na(order.vaccination$patientid)]))), "\n")


  ################################################
  #####Create wide format of vaccination file#####
  ################################################

  cleanvaccinations<-reshape(cleanvaccinations, idvar=c("patientid"),
timevar="dosecombined", direction="wide")
  print(str(cleanvaccinations))
```

```
 rm(list = ls()[!ls() %in% c("cleanvaccinations", "missing.date", "missing.dose",
"patients.with.uPE", "duplicate.vaccination", "duplicate.dosevaccination", "mnd.vaccination",
                            "records.lost.due.to.mixture.of.vactypes", "order.vaccination",
"lost.due.to.uPE","cohort")])
 gc()
 vactypes.df<-subset(cleanvaccinations, select=c(vactype.P1, vactype.P2, vactype.P3,
vactype.B1))
 # cleanvaccinations[,3] <- sapply(seq_len(nrow(vactypes.df)), function(x)
as.character(vactypes.df[x,min(which(!is.na(vactypes.df[x,])))]))
 for(i in 1:dim(cleanvaccinations)[1]) {
  tempVT <- as.character(vactypes.df[i,min(which(!is.na(vactypes.df[i,])))])
  if(length(tempVT)>0) {
   cleanvaccinations[i,3] <- tempVT
  } else {
   cleanvaccinations[i,3] <- NA
  }
 }
 print(str(cleanvaccinations))
 cleanvaccinations<-subset(cleanvaccinations, select=-c(vactype.P2, vactype.P3,
vactype.B1))
 colnames(cleanvaccinations)[3]<-"vactype"

#  ##8. P1 missing but P2 (and/or) P3 (and/or) B1 present
#  cat('Checking for P1 missing, making division bbetween other vaccination data present (P2
(and/or) P3 (and/or) B1 present) or not other vaccination data present\n')
#  vacdata.present.vaccination<-cleanvaccinations[!(is.na(cleanvaccinations$date.P1) &
is.na(cleanvaccinations$date.P2) & is.na(cleanvaccinations$date.P3) &
is.na(cleanvaccinations$date.B1)),]
#  P1missing.vaccination<-vacdata.present.vaccination[is.na(cleanvaccinations$date.P1),]
#
#  cat(sprintf("P1 missing but P2 (and/or) P3 (and/or) B1 present:
%d\n",length(unique(P1missing.vaccination$patientid[!is.na(P1missing.vaccination$patientid
)]))), "Removed from database\n")
#  cat(sprintf("P1 missing (without looking at other data):
%d\n",length(unique(cleanvaccinations[is.na(cleanvaccinations$date.P1),]$patientid)),
"Removed from database\n"))
#      P1missing.novaccination<-cleanvaccinations[is.na(cleanvaccinations$date.P1),]
#      cleanvaccinations<-cleanvaccinations[!is.na(cleanvaccinations$date.P1),]
#
#
#  ##9. P2 missing but P1 and P3 present
```

```
#   vacdata.present.vaccination<-cleanvaccinations[!(is.na(cleanvaccinations$date.P1) &
is.na(cleanvaccinations$date.P2) & is.na(cleanvaccinations$date.P3) &
is.na(cleanvaccinations$date.B1)),]
#   P2missing.vaccination<-vacdata.present.vaccination[is.na(cleanvaccinations$date.P2),]
#   P3missing.vaccination<-vacdata.present.vaccination[is.na(cleanvaccinations$date.P3),]
#   cat('Checking for P2 missing but P1 and P3 present\n')
#   P2missingP1andP3present.vaccination<-
P2missing.vaccination[!(P2missing.vaccination$patientid %in%
c(as.character(P3missing.vaccination$patientid))),]
#   cleanvaccinations<-cleanvaccinations[!(cleanvaccinations$patientid %in%
P2missingP1andP3present.vaccination$patientid),]
#   cat(sprintf("P2 missing but P1 and P3 present:
%d\n",length(P2missingP1andP3present.vaccination[!is.na(P2missingP1andP3present.vaccin
ation$patientid),]$patientid)), "Removed from database\n")
#   cat('\n')
#

#   ##10. List of persons who lost their vaccination data
#   cat('Results of the vaccination cleaning\n')
#   cat('\n')
#
#   persons.vaccdata.lost.vaccination<-
c(as.character(P2missingP1andP3present.vaccination$patientid),
#                                 as.character(unique(P1missing.novaccination$patientid)))
#   cat(sprintf("Summarizing: total number of persons who lost all vaccination data during
cleaning: %d\n",sum(!is.na(persons.vaccdata.lost.vaccination))))
#   cat('.....\n')
#
  persons.vaccdata.somelost.vaccination<-c(as.character(missing.date$patientid),
as.character(missing.dose$patientid), as.character(patients.with.uPE),
                                 as.character(duplicate.vaccination$patientid),
as.character(duplicate.dosevaccination$patientid),
                                 as.character(mnd.vaccination$patientid),
as.character(unique(records.lost.due.to.mixture.of.vactypes$patientid)))
  cat(sprintf("Summarizing: total number of persons who lost some vaccination data during
cleaning: %d\n",sum(!is.na(persons.vaccdata.somelost.vaccination))))
  cat('.....\n')
  cat('\n')

  persons.vaccdata.changed.vaccination<-order.vaccination
```

```
  cat(sprintf("Summarizing: total number of records that were changed (currently only due to order change): %d\n",nrow(persons.vaccdata.changed.vaccination)))
  cat('.....\n')
  cat('\n')

  #write.table(persons.vaccdata.somelost.vaccination,paste(output_folder,"persons.vaccdata.somelost.vaccination.txt",sep="/"),sep=",",row.names=F)
  #write.table(persons.vaccdata.changed.vaccination,paste(output_folder,"persons.vaccdata.changed.vaccination.txt",sep="/"),sep=",",row.names=F)

  ######Cleaning after the vaccination cleaning
  cat('Merging the vaccination-file with the patient-file and additional cleaning steps\n')

  ##1. Keep only those patients in the vaccination file that have a patientid
  cat('Keep only the vaccination records of those patients that have a patientid\n')
  cat('Records of patients without patientid will be removed\n')
  cat(sprintf("patientid is not specified: %d\n",sum(is.na(cleanvaccinations$patientid))), "Removed from database\n")
  cleanvaccinations<-cleanvaccinations[!is.na(cleanvaccinations$patientid),]
  cat('\n')

  ##2. Merging & Limiting the vaccination-file to persons also in patient-file
  cat('Merging the vaccination-file with the patient-file\n')
  cat('Limiting the vaccination-file to persons also in patient-file\n')
  cleanvaccinations<-merge(cohort[c('patientid', 'birthdate', 'gender', 'cohortstart', 'cohortend')], cleanvaccinations, by='patientid', all.y=T)
  cat('Number of records from vaccinationfile lost:', nrow(cleanvaccinations[!is.na(cleanvaccinations$cohortend),]), '\n')
  cleanvaccinations<-cleanvaccinations[!is.na(cleanvaccinations$cohortend),]
  cat('\n')

  ##4. Limiting the patient-time because of uPE or mixture of vactypes
  cat('Adding variables to cleanvaccinations-dataset that indicate right censoring due to uPE or mixture of vactypes\n')
  cat('These new variables dateoffirstuPEvaccination and dateoffirstmixedvaccination possibly indicate the start of right censoring\n')
  lost.due.to.uPE<-transform(lost.due.to.uPE, dateoffirstuPEvaccination=as.numeric(ave(as.numeric(date), patientid, FUN=function(x) min(x, na.rm=T))))
```

```
  records.lost.due.to.mixture.of.vactypes<-transform(records.lost.due.to.mixture.of.vactypes,
dateoffirstmixedvaccination=as.numeric(ave(as.numeric(date), patientid, FUN=function(x)
min(x, na.rm=T))))

  changes.patient.time<-merge(unique(lost.due.to.uPE[c('patientid',
'dateoffirstuPEvaccination')]), unique(records.lost.due.to.mixture.of.vactypes[c('patientid',
'dateoffirstmixedvaccination')]), by='patientid', all.x=T, all.y=T)

  cleanvaccinations<-merge(cleanvaccinations, changes.patient.time[c('patientid',
'dateoffirstuPEvaccination', 'dateoffirstmixedvaccination')], by='patientid', all.x=T)
  cat('\n')

  cleanvaccinations$dateoffirstuPEvaccination<-
as.Date(cleanvaccinations$dateoffirstuPEvaccination, origin="1970-01-01")
  cleanvaccinations$dateoffirstmixedvaccination<-
as.Date(cleanvaccinations$dateoffirstmixedvaccination, origin="1970-01-01")
  cleanvaccinations<-subset(cleanvaccinations, select=-c(diffdates.P1, diffdates.P2,
diffdates.P3, diffdates.B1))


  # cat('writing "cleanvaccinations" to output folder \n')
  #
write.table(cleanvaccinations,paste(output_folder,"cleanvaccinations.txt",sep="/"),sep=",",ro
w.names=F)
  # cat('..... \n')
  # cat('\n')
#
#   ##3. Bin the vaccination dates inside the cohort dates (< not <=)
#   cat('Limiting the vaccination-data to those dates within the patient-time\n')
#   vacdates.outside.cohortdates<-
cleanvaccinations[(cleanvaccinations$date.P1<cleanvaccinations$cohortstart),]
#   vacdates.outside.cohortdates<-
vacdates.outside.cohortdates[!(is.na(vacdates.outside.cohortdates$patientid)),]
#   cleanvaccinations<-cleanvaccinations[!(as.factor(cleanvaccinations$patientid) %in%
as.factor(vacdates.outside.cohortdates$patientid)),]
#   cat(sprintf("Persons with vaccination dates outside the cohort dates (date P1 < cohortstart):
%d\n",length(unique(vacdates.outside.cohortdates$patientid[!is.na(vacdates.outside.cohortdat
es$patientid)]))), "Removed from database\n")
#

  ###Exporting the cleanvaccinations file
```

```
 return(cleanvaccinations)

}
```

IMPORT EVENTS AND PATIENTS FILE and CLEAN

```
######ADVANCE: pertussis POC f_import events_population

####################
##Import the data##
####################

cat('\n')
cat('Importing the data\n')
cat('\n')

cat('\n')
cat('The import module will create 3 functions; \n')
cat('events.import.function, population.import.function and vaccination.import.function,\n')
cat('for importing and cleaning data.\n')
cat('This module should be ran first and the workspace should not be cleaned afterwards\n')
cat('\n')

##############
#-POPULATION-#
##############

population.import.function=function(folder, population_file,date_format){
 ####Importing and Cleaning the population_file
 #checking parameters
 if (length(folder)==0){
  throw("No work space specified")
 }else{
  setwd(folder)
 }

 if (length(date_format)==0){
  warning("No date format specified try YYYYmmdd")
  date_format="%Y%m%d"
 }
```

```
if (date_format=="YYYYmmdd"){
  date_format="%Y%m%d"
}


if (date_format=="YYYY-mm-dd"){
  date_format="%Y-%m-%d"
}

if (date_format=="YYYY/mm/dd"){
  date_format="%Y/%m/%d"
}



  #Create output folder
  output_folder=paste(folder,"results_population",sep="/")
  dir.create(output_folder)#,showwarnings=T,recursive=T)

  ####Read in the data
  population<-read.table(patients_file, header=T, sep=",", na.strings = c("","NA"))
  colnames(population)=tolower(colnames(population))
  population$birthdate<-as.Date(as.character(population$birthdate), "%Y%m%d")
  population$startdate<-as.Date(as.character(population$startdate), "%Y%m%d")
  population$enddate<-as.Date(as.character(population$enddate), "%Y%m%d")

  ####Additional data cleaning for the population file
  cat('Cleaning of the populationfile\n')
  cat('\n')
  cat('Number of records in the population_file:', nrow(population), '\n')

  cat('Removing duplicated patientid\n')
  duplicated.patientid.population<-population[duplicated(population$patientid),]
  cat('Number of duplicated patientid:', nrow(duplicated.patientid.population), 'Keep first
record\n')
  population<-population[!duplicated(population$patientid),]

  cat('Removing records with missingess in birthdate, patientid, startdate, enddate, gender \n')
  missing.population<-population[!complete.cases(population),]
  cat('Number of records with missing variables:', nrow(missing.population), 'Removed
record\n')
  population<-population[complete.cases(population),]
```

```
  cat('Select those birthcohort from 2008 \n')
  cat('Number of records outside of birthcohort 2008:2020:',
nrow(population[!(format(population$birthdate, '%Y') %in% c(2008:2020)),]), 'Removed\n')
  population<-population[format(population$birthdate, '%Y') %in% c(2008:2020),]

  cat('Number of records remaining in the population_file:', nrow(population), '\n')

  cleanpopulation<-population

  ###Exporting the cleanpopulation file
  cat('Number of records left in the population_file:', nrow(population), '\n')
  cat('writing "cleanpopulation" to output folder \n')

write.table(cleanpopulation,paste(output_folder,"cleanpopulation.txt",sep="/"),sep=",",row.na
mes=F)
  cat('..... \n')
  cat('\n')
  return(cleanpopulation)
}

##########
#-EVENTS-#
##########

events.import.function=function(folder, events_file,date_format){
  ####Importing and Cleaning the events_file
  #checking parameters
  if (length(folder)==0){
    throw("No work space specified")
  }else{
    setwd(folder)
  }

  if (length(date_format)==0){                    #This is not used! A fix format is used
below - Maria
    warning("No date format specified try YYYYmmdd")
    date_format="%Y%m%d"
  }


  if (date_format=="YYYYmmdd"){
```

```
  date_format="%Y%m%d"
 }


 if (date_format=="YYYY-mm-dd"){
  date_format="%Y-%m-%d"
 }

 if (date_format=="YYYY/mm/dd"){
  date_format="%Y/%m/%d"
 }


 #Create output folder
 output_folder=paste(folder,"results_events",sep="/")
 dir.create(output_folder)#,showwarnings=T,recursive=T)


 ####Read in the data
 events<-read.table(events_file, header=T, sep=",", fill=T, na.strings = c("","NA"))
 colnames(events)=tolower(colnames(events))
 events$date<-as.Date(as.character(events$date), "%Y%m%d")
 print(str(events))
 ####Additional data cleaning for the events file
 cat('Cleaning of the events_file\n')
 cat('\n')
 cat('Number of records in the events_file:', nrow(events), '\n')
 cat('Removing duplicated patientid+eventtype+date\n')
 duplicated.events<-events[duplicated(events[c('patientid', 'date', 'eventtype')]),]
 cat('Number of duplicated patientid+eventtype+date:', nrow(duplicated.events), 'Keep first
record\n')
 events<-events[!duplicated(events[,c('patientid', 'date', 'eventtype')]),]

 cat('Removing records with missingess in patientid, date, eventtyper \n')
 missing.events<-events[!complete.cases(events),]
 cat('Number of records with missing variables:', nrow(missing.events), 'Removed record\n')
 events2<-events[complete.cases(events),]

 cat('Removing records with date <01/01/2014 \n')
 lateevents<-events[events$date>as.Date('20201231', format='%Y%m%d'),]
 cat('Number of records with dates outside of 2014-2020:', nrow(lateevents), 'Removed
record\n')
```

```
 events2<-events[events$date>=as.Date('20140101', format='%Y%m%d') &
events$date<=as.Date('20201231', format='%Y%m%d'),]

 cat('Number of remaining events in the events_file:', nrow(events2), '\n')

 cleanevents<-events2

 ###Exporting the cleanevents file
 cat('Number of records left in the events_file:', nrow(events), '\n')
 cat('writing "cleanevents" to output folder \n')
 write.table(cleanevents,paste(output_folder,"cleanevents.txt",sep="/"),sep=",",row.names=F)
 cat('..... \n')
 cat('\n')
 return(cleanevents)
}
```

SEE STATISTICAL ANALYSIS PLAN
Phase 3: Transformation of CDM data files into the analytical dataset

---

PERSON TIME for RISK EVENTS and OVERALL

```
################################################################################
# POC1.2 function f_dtrans_pyrisk
# Data transformation step3: from individual level data (wide format) to aggregated data that
will serve as input to the dashboard
# Kaatje Bollaerts

# INPUT parameters:
#   dat: data in wide format
#   FOLDER_OUT: folder name to save aggregated data.
#   OUTFILE: name of the Rdat out data file - omit .RData. One file will be generated for each
event in eventlist
#   rwdlist2: list with unique risk window lenghts


# OUTPUT FILES GENERATED:
#   X: person time in risk windows after each dose of varying lengths (1,2 and 3), person time
in vaccination eligible age excl risk windows, person time overall by calendar weeks

# final check and changes; 07 May 2018 (Kaat Bollaerts)
```

```
################################################################################

#----------------------------------------------------------

"f_dtrans_pyrisk" <- function(dat, folder_out, outfile, rwdlist2, startstudy, endstudy) {


  # weeks from startstudy till endstudy
  seqwk <- as.Date(seq(startstudy, endstudy, by = 'week'))
  seqwk <- seqwk[1: (length(seqwk)-1)]                    # date start of every week, from start study till last week before end study
  n_wk <- length(seqwk)                                   # total number of study weeks
  wk_id <- seq(1, n_wk, 1)
  wk_id <- data.table(wk_id)


  # preallocate output matrix
  X <- matrix(data=NA, nrow=n_wk, ncol=23)

  for(i in 1:n_wk) {

    # start and end of week i
    wki_start <- seqwk[i]
    wki_end <- seqwk[i] + 7

    pt_a <- pmax(dat$startfu, wki_start)
    pt_b <- pmin(dat$endfu, wki_end)
    PT <- round(as.numeric(difftime(pt_b, pt_a, units = 'days')))
    PT[is.na(PT)| PT < 0] <- 0
    PTall <- sum(PT)

    # person time within the risk windows (for each of the 3 doses and the different risk window lengths) and person time for the baseline events
    for (ii in 1:3) {

      # (start) fu within risk window
      vaccdate <- dat[, c(paste0("date.P", ii))]
      rw_a <- pmax(dat$startfu, wki_start,vaccdate)

      # (start) fu for person time events outside risk window, but within 'vaccination' age range
```

```
    medvacc <- median((as.numeric(difftime(vaccdate, dat$DoB, units = 'days')) +1), na.rm =
TRUE)

    bs_lower_age <- dat$DoB + (medvacc - 14)          # date at child's lower age limit for
vaccination eligible age
    bs_upper_age <- dat$DoB + (medvacc + 14)          # date at child's upper age limit for
vaccination eligible age
    bs_a <- pmax(dat$startfu, wki_start, bs_lower_age) # start fu for person time events outside
risk window

    # (start) fu 10 days after vaccination
    bs2_a <- pmax(dat$startfu, wki_start, vaccdate + 10)


    for (iii in 1:length(rwdlist2)) {

     # 1.person time within risk window
      rwdays <- rwdlist2[iii]
      vaccdate_rw_end <- vaccdate + rwdays
      rw_b <- pmin(dat$endfu, wki_end, vaccdate_rw_end)
      PTrw <- round(as.numeric(difftime(rw_b, rw_a, units = 'days')))
      PTrw[is.na(PTrw)| PTrw < 0] <- 0
      PTsum <- sum(PTrw)
      outvarname <- paste0('PT_d', ii, '_rw', rwdays)
      assign(outvarname, PTsum)

     # 2.person time within age 'vaccination' age range
      bs_b <- pmin(dat$endfu, wki_end, bs_upper_age)
      PTbl <- round(as.numeric(difftime(bs_b, bs_a, units = 'days')))
      PTbl[is.na(PTbl)| PTbl < 0] <- 0

      # excluding "risk window after vaccination" time within baseline person time
      D_a <- pmax(rw_a, bs_a)
      D_b <- pmin(rw_b, bs_b)

      PTrw2 <- D_b - D_a
      PTrw2[is.na(PTrw2)| PTrw2 < 0] <- 0

      PTbl2 <- PTbl - PTrw
      PTbl2[is.na(PTbl2)| PTbl2 < 0] <- 0
      PTsum <- sum(PTbl2)
      outvarname <- paste0('PTbl_d', ii, '_rw', rwdays)
```

```
      assign(outvarname, PTsum)

    }

     # 3.person time within [10-15] days post-vaccination
      bs2_end <- vaccdate + 15
      bs2_b <- pmin(dat$endfu, wki_end, bs2_end)
      PTbl2 <- round(as.numeric(difftime(bs2_b, bs2_a, units = 'days')))
      PTbl2[is.na(PTbl2)| PTbl2 < 0] <- 0
      PTsum <- sum(PTbl2)
      outvarname <- paste0('PTbl2_d', ii)
      assign(outvarname, PTsum)

  }

  X[i,] <- c(i, PT_d1_rw1,PT_d1_rw2,PT_d1_rw3,
                        PT_d2_rw1,PT_d2_rw2,PT_d2_rw3,
                        PT_d3_rw1,PT_d3_rw2,PT_d3_rw3,
                        PTbl_d1_rw1,PTbl_d1_rw2,PTbl_d1_rw3,
                        PTbl_d2_rw1,PTbl_d2_rw2,PTbl_d2_rw3,
                        PTbl_d3_rw1,PTbl_d3_rw2,PTbl_d3_rw3,
                        PTbl2_d1,PTbl2_d2,PTbl2_d3,PTall)
}

# save
colnames(X) <- c("Wk_id", "PT_d1_rw1","PT_d1_rw2","PT_d1_rw3",
                 "PT_d2_rw1","PT_d2_rw2","PT_d2_rw3",
                 "PT_d3_rw1","PT_d3_rw2","PT_d3_rw3",
                 "PTbl_d1_rw1","PTbl_d1_rw2","PTbl_d1_rw3",
                 "PTbl_d2_rw1","PTbl_d2_rw2","PTbl_d2_rw3",
                 "PTbl_d3_rw1","PTbl_d3_rw2","PTbl_d3_rw3",
                 "PTbl2_d1","PTbl2_d2","PTbl2_d3",
                 "PTall")

outfile <- paste0(folder_out, outfile, ".RData")
save(X, file = outfile)

}
```

NUMBER OF RISK EVENTS within RISK WINDOWS and baseline

```
################################################################################
# Data transformation step2: from individual level data (wide format) to aggregated data that will serve as input to the dashboard.
# the individual level data is generated in "1_dataimport.R"

# INPUT parameters:
#   FOLDER_IN: folder name to read data containing the individual level data
#   INFILE: name of the Rdat input data file - omit .RData
#   FOLDER_OUT: folder name to save aggregated data.
#   OUTFILE: name of the Rdat out data file - omit .RData. One file will be generated for each event in eventlist
#   eventlist: list with risk events of interest
#   rwdlist: list with risk windows for each event (some order as in eventlist)
#   startstudy: macro variable data at start study
#   wk_id: vector containing week id variable
#   mode_agevacc: vector (3x1) with modes for age at vaccination for dose 1, 2 and 3


# OUTPUT FILES GENERATED:
#   OUTFILE_ISR: cumulative number of ISR events over calendar time (in weeks), within the risk window and outside risk window but at vaccination eligble age, for eacch of the 3 doses
#   OUTFILE_FEVER: .....of FEVER events ....
#   OUTFILE_SOMNOL: ...
#   OUTFILE_FCONVULS
#

# # final check and changes; 07 May 2018 (Kaat Bollaerts)
################################################################################

"f_dtrans_nrisk" <- function(dRaw, dWide, folder_out, outfile, eventlist, rwdlist, startstudy, wk_id) {


  counter <- 0
  for (j in eventlist) {

    selVar <- dRaw[dRaw$EventId == j, c("AnonId", "EventId", "EventDate")]

    d <- merge(x = dWide, y = selVar, by = "AnonId")
    n <- dim(d)[1]
    counter <- counter + 1
```

```
  rw_end <- rwdlist[counter]

  eventdate <- d$EventDate


  if(j == "PERT"){

    # calculate time at event
    TimeEvent_wk <- ceiling(as.numeric(difftime(eventdate, startstudy, units = 'weeks')))
    dt <- data.table(TimeEvent_wk)
    dt2 <- dt[,list(N_pert=.N), by=list(TimeEvent_wk)]   # cross-tabulate and count
    dt2 <- na.omit(dt2[with(dt2, order(TimeEvent_wk)),])

    # count weeks without events
    dt3 <- merge(wk_id, dt2, by.x = c('wk_id'), by.y = c('TimeEvent_wk'), all=TRUE)
    f_dowle3(dt3)                    # replace NA with 0


  } else {

    # calculate age and time at events
    AgeEvent_wk <- floor(as.numeric(difftime(eventdate, d$DoB, units = 'weeks')))
    TimeEvent_wk <- ceiling(as.numeric(difftime(eventdate, startstudy, units = 'weeks')))


    # total number of events
    Ntot <- rep(0,n)
    Ntot[!is.na(eventdate)] <-  1


    # create indicator variables indicating events a) within risk window for each of the 3 doses and b) around vaccination eligble age
    for (k in 1:3 ) {
      # varname <- paste0('d$date.P', k)
      vaccdate <- d[,paste0("date.P",k)]

      # (start) fu for person time events outside risk window, but within 'vaccination' age range
      medvacc <- median((as.numeric(difftime(vaccdate, d$DoB, units = 'days')) +1), na.rm = TRUE)

      # a) number of events within the risk period
      vara <- rep(0,n)
```

```
    vara[!is.na(eventdate) & !is.na(vaccdate)  & (eventdate > vaccdate) & (eventdate <=
vaccdate + rw_end)] <-  1
    varname <- paste0("rw_d", k)
    assign(varname, vara)

    # b1) number of events around vaccination eligible age, but not in risk period
    aa <- medvacc - 14
    bb <- medvacc + 14
    varb <- rep(0,n)
    varb[!is.na(eventdate) & vara == 0  & (AgeEvent_wk > aa) & (AgeEvent_wk <= bb)] <-
1

    varname <- paste0("bl_d", k)
    assign(varname, varb)

    # b2) number of events [10-15] days after vaccination
    vara <- rep(0,n)
    vara[!is.na(eventdate) & !is.na(vaccdate)  & (eventdate > vaccdate + 9) & (eventdate <=
vaccdate + 15)] <-  1
    varname <- paste0("bl2_d", k)
    assign(varname, vara)

  }
  # cross-tabulate by time and sort
  dt <- data.table(TimeEvent_wk, Ntot, rw_d1, rw_d2, rw_d3, bl_d1, bl_d2, bl_d3, bl2_d1,
bl2_d2, bl2_d3)
  dt2 <- dt[,list(Ntot = sum(Ntot), N_rw_d1=sum(rw_d1), N_rw_d2=sum(rw_d2),
N_rw_d3=sum(rw_d3), N_bl_d1=sum(bl_d1), N_bl_d2=sum(bl_d2), N_bl_d3=sum(bl_d3),
N_bl2_d1=sum(bl2_d1), N_bl2_d2=sum(bl2_d2), N_bl2_d3=sum(bl2_d3)),
by=list(TimeEvent_wk)]  # cross-tabulate and count
  dt2 <- na.omit(dt2[with(dt2, order(TimeEvent_wk)),])  # order and omit missing values

  # count weeks without events
  dt3 <- merge(wk_id, dt2, by.x = c('wk_id'), by.y = c('TimeEvent_wk'), all=TRUE)
  f_dowle3(dt3)                                                    # replace NA with 0

  # cumulative sums
  dt3$cumN_tot <- cumsum(dt3$Ntot)

  dt3$cumN_rw_d1  <- cumsum(dt3$N_rw_d1)
  dt3$cumN_rw_d2  <- cumsum(dt3$N_rw_d2)
  dt3$cumN_rw_d3  <- cumsum(dt3$N_rw_d3)
```

```
    dt3$cumN_bl_d1  <- cumsum(dt3$N_bl_d1)
    dt3$cumN_bl_d2  <- cumsum(dt3$N_bl_d2)
    dt3$cumN_bl_d3  <- cumsum(dt3$N_bl_d3)

    dt3$cumN_bl2_d1  <- cumsum(dt3$N_bl2_d1)
    dt3$cumN_bl2_d2  <- cumsum(dt3$N_bl2_d2)
    dt3$cumN_bl2_d3  <- cumsum(dt3$N_bl2_d3)
   }

    outfile1 <- paste0(folder_out, outfile,"_", j, Sys.Date(), ".RData")
    save(dt3, file = outfile1)
  }
}
```

NUMBER OF DOSES by CALENDAR TIME

```
################################################################################
# POC1.2 function f_dtrans_ndose
# Data transformation step2: from individual level data (wide format) to aggregated data that will serve as input to the dashboard.
# the individual level data is generated in "1_dataimport.R"

# INPUT parameters:
#   FOLDER_IN: folder name to read data containing the individual level data
#   INFILE: name of the Rdat input data file - omit .RData
#   FOLDER_OUT: folder name to save aggregated data.
#    OUTFILE: name of the Rdat out data file - omit .RData. Tree files will be generated: outfile_d1, outfile_d2 and outfile_d3 (one for each dose)

# OUTPUT FILES GENERATED:
#   OUTFILE_d1: number of doses by calendar time (in weeks) x age (in weeks), dose 1
#   OUTFILE_d2: number of doses by calendar time (in weeks) x age (in weeks), dose 2
#   OUTFILE_d3: number of doses by calendar time (in weeks) x age (in weeks), dose 3
################################################################################

"f_dtrans_ndose" <- function(wide_res, folder_out, outfile, startstudy) {

  # read data
  # datname <- paste0(folder_in, infile, ".RData")
  d <- wide_res
```

```
 # calculate nr of doses by calendar time x age, for each dose 1,2,3
 for (j in 1:3) {

   # varname <- paste0('d$date.P', j)
   vaccdate <- d[,paste0("date.P", j)]

   AgeVac_wk <- floor(as.numeric(difftime(vaccdate, d$DoB, units = 'weeks')))     # calculate
age and time at vaccination
   TimeVac_wk <- ceiling(as.numeric(difftime(vaccdate, startstudy, units = 'weeks')))

   dt <- data.table(TimeVac_wk, AgeVac_wk)
   dt2 <- dt[,.N,by=list(TimeVac_wk, AgeVac_wk)]                    # cross-tabulate and
count
   dt2 <- na.omit(dt2[with(dt2, order(TimeVac_wk, AgeVac_wk)),])          # order and
omit missing values (~non-vaccinated subjects)

   # save output
   outname <- paste0(folder_out, outfile, j, Sys.Date(), ".RData")
   save(dt2, file = outname)
 }
}
```

DENOMINATOR INFORMATION for COVERAGE

```
# POC1.2  function f_dtrans_denom

# Data transformation step3: generates aggregated data table for plotting coverage by birth
cohort
# Kaatje Bollaerts - 14/03/2018

# INPUT parameters:
#   dat: data in wide format!!
#   folder_out: folder name to save aggregated data.
#   outifle:
#   startstudy: start of study
#   endstudy: end of study


# OUTFILE FILE: one output file with the following variables: Time since start study, DoB
(year-month), N, Nvacc1, Nvacc2, Nvacc3, Cov_vacc1, Cov_vacc2, Cov_vacc3
```

```
#-----------------------------------------------------------

"f_dtrans_denom" <- function(dat, folder_out, outfile, startstudy, endstudy) {

  # weeks from startstudy till end study
  seqwk <- as.Date(seq(startstudy, endstudy, by = 'week'))
  seqwk <- seqwk[1: (length(seqwk)-1)]                    # date start of every week, from start study till last week before end study
  n_wk <- length(seqwk)                                   # total number of study weeks
  wk_id <- seq(1, n_wk, 1)
  wk_id <- data.table(wk_id)


  # preallocate output matrices
  denom_n_dob <- data.table(Time_wki=numeric(), DoB_ymi=numeric(), N_all=numeric(), N_vacc1=numeric(), N_vacc2=numeric(), N_vacc3=numeric(), Cov_vacc1=numeric(), Cov_vacc2=numeric(), Cov_vacc3=numeric())  # total number of children by week x month-year of birth

  for(i in 1:n_wk) {

    # week i
    wki <- seqwk[i]
    Time_wki <- unlist(wk_id[i])

    # Only keep children born before week i, with a max. age of 1.5 years and with a complete fu (start follow up at 1 month or earlier and end follow up at 2 years or later)
    d2 <- dat
    d2$Age_wki2 <- floor(as.numeric(difftime(wki, d2$DoB, units = 'weeks')))
    d2$Age_startfu <- floor(as.numeric(difftime(d2$startfu, d2$DoB, units = 'weeks')))
    d2$Age_endfu <- floor(as.numeric(difftime(d2$endfu, d2$DoB, units = 'weeks')))
    d2 <- d2[(d2$Age_wki2 > 0 & d2$Age_wki2 <= (1.5*52.25) & d2$Age_startfu <= 6),]


    # total number of children
    ni <- dim(d2)[1]

    if (ni > 0){
      Den_all <- rep(1,ni)
      DoB_ymi <- format(d2$DoB, "%m/%Y")
      # total number of children vaccinated with dose1, dose2 and dose3
```

```
    for (j in 1:3) {
      # print(str(d2))
      vaccdate <- d2[, c(paste0("date.P", j))]

      Den_vac <- rep(0,ni)
      Den_vac[!is.na(vaccdate) & vaccdate <= wki] <- 1

      varname2 <- paste0("Den_vac", j)
      assign(varname2, Den_vac)
     }
    } else {
      DoB_ymi <- NA
      Den_all <- NA
      Den_vac1 <- NA
      Den_vac2 <- NA
      Den_vac3 <- NA
    }


   #count children (total population, children vaccinated with dose 1, dose 2 and dose 3, by year-month of birth)
   di <- data.table(DoB_ymi, Den_all, Den_vac1, Den_vac2, Den_vac3)
   ddi <- di[ ,list(N_all =sum(Den_all), N_vacc1 =sum(Den_vac1), N_vacc2 =sum(Den_vac2), N_vacc3 =sum(Den_vac3)), by=list(DoB_ymi)]
   ddi$Cov_vacc1 <- ddi$N_vacc1/ddi$N_all
   ddi$Cov_vacc2 <- ddi$N_vacc2/ddi$N_all
   ddi$Cov_vacc3 <- ddi$N_vacc3/ddi$N_all

   # append data
   Time_wki <- unlist(wk_id[i])
   Time_wki <- rep(Time_wki, dim(ddi)[1])
   denomi <- data.frame(Time_wki,ddi)
   denom_n_dob <- rbind(denom_n_dob, denomi)
   dat.denom_n_dob <- denom_n_dob[with(denom_n_dob, order(DoB_ymi, Time_wki)),]
  }
  # save output
  outname <- paste0(folder_out, outfile, Sys.Date(),".RData")
  save(dat.denom_n_dob, file = outname)
}
```

Appendix 3: R Shiny scripts

SEE STATISTICAL ANALYSIS PLAN
Phase 4: Web-application server

Server.R

```
#####################################################################
#   Data visualization dashboard for ADVANCE POC 1.2        #
#   Developed in R using the 'Shiny' package (and the       #
#   associated shinydashboard package) by RStudio.          #
#=================================================================#
#   Author:   Tom De Smedt                          #
#   Email:    tom.desmedt at p-95 dot com                   #

#=================================================================#

# Load libraries
library(highcharter)
# library(sas7bdat)
print("Packages loaded")
print(Sys.time())

# Set-up password protection system
Logged = FALSE;

# Set color palette
colorPalQuant <- c("#7f3b08",
  "#b35806",
  "#e08214",
  "#fdb863",
  "#fee0b6",
  "#b2abd2",
  "#8073ac",
  "#542788",
```

```
  "#2d004b")

# Prepare data and functions for number of doses and coverage



"f_ndoses" <- function(brks, db, dose) {
  #---------------------------------------------------------
  # Load data
  load(paste0("data/resultsPOC12_ndose_", db,"_", dose,".RData"))
  load(paste0("data/resultsPOC12_CovbyYMBirth_", db,"_.RData"))
  dat.denom_n <- dat.denom_n_dob
  dat.denom_n$Age_wki <- dat.denom_n$Age_wki + min(na.omit(dat.denom_n$Age_wki))

  library('data.table')
  n_brks <- length(brks)
  brks0 <- c(0, brks, 318)
  popsize <- diff(brks0)* 777165
  datn <- dat.denom_n

  #-------------------------------------------------------------------------------------------------------
  ## DOSE 1
  datv <- dt2

  # number of doses per week per age group
  age <- unlist(subset(datv,select=c("AgeVac_wk")))
  Age_gr <- cut(age, breaks = brks0, labels = FALSE)
  datv <- data.table(datv,Age_gr)

  datvv <- datv[,list(Nvac=sum(N)),by=list(TimeVac_wk, Age_gr)]

  # number of children per week per age group
  age <- unlist(subset(datn,select=c("Age_wki")))
  Age_gr <- cut(age, breaks = brks0, labels = FALSE)
  datn <- data.table(datn,Age_gr)

  datnn <- datn[,list(N=sum(N_all)),by=list(Time_wki, Age_gr)]

  # merge data.tables
  dtot <- merge(datnn, datvv, by.x = c('Time_wki', 'Age_gr'), by.y = c('TimeVac_wk', 'Age_gr'),
all=TRUE)
  dtot$pop <- 0
```

```
 for(i in 1:n_brks) {
  dtot$pop[dtot$Age_gr == i] <- popsize[i]
 }

 # number of vaccinations per week, extrapolated to the whole UK population
 dtot$weight <- dtot$pop/dtot$N
 dtot$Nweigth <- round(dtot$weight * dtot$Nvac,0)

 # reshape
 dtot <- dtot[ , c('Time_wki','Age_gr','Nweigth'), with = FALSE]
 plot.ndose1 <- reshape(dtot, idvar = c("Time_wki"), timevar=c("Age_gr"), direction =
"wide")

 plot.ndose1 <- plot.ndose1[plot.ndose1$Time_wki > 52 & plot.ndose1$Time_wki < 253,]

 return(datvv)
}



"f_coverage" <- function(brks, db, dose) {
 #------------------------------------------------------
 # load data
 load(paste0("data/resultsPOC12_CovbyYMBirth_", db,"_.RData"))
 dat.denom_n <- dat.denom_n_dob
 dat.denom_n <- dat.denom_n[dat.denom_n$Age_wki > 0,]
 library('data.table')

 #------------------------------------------------------
 # only keep data with ages as in 'brks'
 d <- dat.denom_n[dat.denom_n$Age_wki  %in% brks]
 if(dose == "d1") {
  d$cov <- round((d$N_vacc1/d$N_all) * 100,1)
 }
 if(dose == "d2") {
  d$cov <- round((d$N_vacc2/d$N_all) * 100,1)
 }
 if(dose == "d3") {
  d$cov <- round((d$N_vacc3/d$N_all) * 100,1)
 }

 d <- d[ , c('Time_wki', 'Age_wki', 'cov'), with = FALSE]
```

```
  plot.cov <- reshape(d, idvar = c("Time_wki"), timevar=c("Age_wki"), direction = "wide")

  return(plot.cov)
}

print(Sys.time())

print("Initial data & functions loaded - starting server")
print(Sys.time())

shinyServer(function(input, output, session) {

  print("Server started")
  print(Sys.time())


  output$textuser <- reactive({
    USER$Logged == TRUE
  })


  # Encapsulate code in observe and if to make dependent on passw status
  observe({
    if (USER$Logged == TRUE) {

      # Title for navigation bar, dependent on tab
      output$header_title <- renderUI({
        if(input$tabs == 'view1'){
          temp_text <- "Doses"
        }
        if(input$tabs == 'view2'){
          temp_text <- "Coverage"
        }
        if(input$tabs == 'view3'){
          temp_text <- "Risks"
        }
        if(input$tabs == 'view4'){
          temp_text <- "Benefits"
        }
        if(input$tabs == 'view5'){
          temp_text <- "About"
        }
```

```
    HTML(paste0(temp_text))
    return(HTML(""))
  })

  # Selectors for age, dose, etc.
  output$selAgeCov1 <- renderUI({
    textInput(inputId = "agegDosesCov1", label = "Select age groups (in weeks)", value = "8,
12, 16, 20, 24, 44, 52, 104", width = 400)
  })

  output$selAgeCov2 <- renderUI({
    textInput(inputId = "agegDosesCov2", label = "Select age groups (in weeks)", value = "8,
12, 16, 20, 24, 44, 52, 104", width = 400)
  })

  output$selCountry1 <- renderUI({
    div(style = "color: #222222; font-size: 125%;", selectInput(inputId = "selCountry", label
= "Select database", choices = list("RCGP RCS (UK)" = "UK_RCGP", "SSI (Denmark)" =
"DK_SSI", "SIDIAP (Spain)" = "SIDIAP", "ATSVP (Italy)" = "ASLCR", "ARS (Italy)" =
"ARS"), selected = "UK_RCGP", multiple = FALSE, width = 200))
  })

  output$selDoseCov1 <- renderUI({
    selectInput(inputId = "selDoseCov1", label = "Select dose", choices = list("Dose 1" =
"d1", "Dose 2" = "d2", "Dose 3" = "d3"), selected = "d1", multiple = FALSE, width = 200)
  })

  output$selDoseCov2 <- renderUI({
    selectInput(inputId = "selDoseCov2", label = "Select dose", choices = list("Dose 1" =
"d1", "Dose 2" = "d2", "Dose 3" = "d3"), selected = "d1", multiple = FALSE, width = 200)
  })

  output$ChartDose <- renderHighchart({
    country <- input$selCountry
    brks <- as.numeric(unlist(strsplit(input$agegDosesCov1, ",")))
    dose <- input$selDoseCov1

    tempDataD1 <- f_ndoses(brks, country, "d1")
    tempDataD2 <- f_ndoses(brks, country, "d2")
    tempDataD3 <- f_ndoses(brks, country, "d3")

    if(dose == "d1") {
```

```
  tempData <- tempDataD1
 }
 if(dose == "d2") {
  tempData <- tempDataD2
 }
 if(dose == "d3") {
  tempData <- tempDataD3
 }

 tempSeq <- seq.Date(from = as.Date("2014-01-01"), to = as.Date("2020-01-01"), by = "week")
 tempData <- tempData[tempData$TimeVac_wk > 0,]
 tempData$Date <- tempSeq[tempData$TimeVac_wk]

 breaks <- c("0", as.character(brks), "208")
 nameBreaks <- paste0("Age ", breaks[1:(length(breaks)-1)], "-", breaks[2:(length(breaks))], " wks")
 tempData$AgeChar <- factor(x = nameBreaks[tempData$Age_gr], levels = nameBreaks)
 tempData <- tempData[!is.na(tempData$AgeChar)]

 hc <- highchart()
 hc <- hc_colors(hc = hc, colors = colorPalQuant)
 hc <- hc_yAxis(hc = hc, title = list(text = "# Administered Doses"), reversedStacks = FALSE, opposite = FALSE, turnopposite = TRUE)
 tempData$tooltip1 <- as.character(tempData$Nvac)
 tempData$tooltip1[tempData$Nvac < 5] <- "<5"
 print(tempData$tooltip1)
 hc <- hc_tooltip(hc = hc, shared = TRUE)
 uniAge <- unique(tempData$AgeChar)
 hc <- hc_add_series(hc = hc, data = tempData, type = "column", hcaes(x = Date, y = Nvac, group = AgeChar), stacking ="normal", tooltip = list(pointFormat = '<span style="color:{point.color}">\u25CF</span> {series.name}: <b>{point.tooltip1}</b><br/>', dateTimeLabelFormats = list(day = "%b %e, %Y")))
 hc <- hc_chart(hc = hc, zoomType = "xy")
 hc <- hc_legend(hc = hc, enabled = TRUE)
 hc <- hc_navigator(hc = hc, maskFill = "rgba(255,140,0,0.15)", series = list(lineColor = 'rgba(255,140,0,0.5)'))
 hc <- hc_xAxis(hc = hc, type = "datetime", max = datetime_to_timestamp(as.Date("2018-06-07")))#, min = datetime_to_timestamp(minDate), max = datetime_to_timestamp(maxDate))
 hc <- hc_exporting(hc = hc,enabled=TRUE, buttons=list( contextButton = list( align = 'left', verticalAlign = 'top' ) ) )
```

```
  return(hc)
 })

 dataDoBRaw <- reactive({
  load(paste0("data/resultsPOC12_CovbyYMBirth_", input$selCountry,"_.RData"))
  dataDoB <- dat.denom_n_dob
  dataDoB$Year <- as.numeric(format(as.Date(x = paste0("01/", dataDoB$DoB_ymi),
format = "%d/%m/%Y"), "%Y"))
  return(dataDoB)
 })

 output$selYM <- renderUI({
  dataDoB <- dataDoBRaw()
  choicesTemp <- sort(unique(dataDoB$Year), na.last = NA)
  choicesTemp <- choicesTemp[choicesTemp != 2013]
  return(checkboxGroupInput(inputId = "selYM", label = "Select Year-Month Birth
Cohorts", choices = choicesTemp, selected = choicesTemp, inline = TRUE))
 })

 output$ChartCovDoB <- renderHighchart({
  dataDoB <- dataDoBRaw()
  tempData <- dataDoB[dataDoB$Year %in% input$selYM,]
  tempData <- tempData[order(tempData$Year),]
  tempSeq <- seq.Date(from = as.Date("2014-01-01"), to = as.Date("2018-07-01"), by =
"week")
  tempData$Date <- tempSeq[tempData$Time_wki]
  tempData <- tempData[!is.na(tempData$Date)]
  uniDoB <- unique(tempData$DoB_ymi)
  uniDoB <- uniDoB[!is.na(uniDoB)]
  dose <- input$selDoseCov2
  if(dose == "d1") {
   tempData$Cov <- tempData$Cov_vacc1
  }
  if(dose == "d2") {
   tempData$Cov <- tempData$Cov_vacc2
  }
  if(dose == "d3") {
   tempData$Cov <- tempData$Cov_vacc3
  }
  print(str(tempData))
  tempData$Cov <- round(tempData$Cov * 100, 2)
  hc <- highchart(type = "stock")
```

```
    hc <- hc_colors(hc = hc, colors = colorPalQuant)
    hc <- hc_legend(hc = hc, enabled = TRUE)
    hc <- hc_tooltip(hc = hc, split = FALSE, shared = TRUE)
    for(i in 1:length(uniDoB)) {
      tempData1 <- tempData[tempData$DoB_ymi == uniDoB[i],]
      hc <- hc_add_series(hc = hc, data = tempData1, type = "line", hcaes(x = Date, y = Cov),
name = uniDoB[i], tooltip = list(shared = FALSE, split = FALSE))
    }
    hc <- hc_chart(hc = hc, zoomType = "xy")
    hc <- hc_navigator(hc = hc, maskFill = "rgba(255,140,0,0.15)", series = list(lineColor =
'rgba(255,140,0,0.5)'))
    hc <- hc_yAxis(hc = hc, title = list(text = "Coverage [%]"), max = 100, opposite = FALSE,
turnopposite = TRUE)
    hc <- hc_xAxis(hc = hc, type = "datetime", max = datetime_to_timestamp(as.Date("2018-
06-07")))#, min = datetime_to_timestamp(minDate), max = datetime_to_timestamp(maxDate))
    hc <- hc_exporting(hc = hc,enabled=TRUE, buttons=list(
contextButton = list( align = 'left', verticalAlign = 'top' )
)  )
    return(hc)
  })

  output$selDose2 <- renderUI({
    selectInput(inputId = "selDose2", label = "Select dose", choices = list("Dose 1" = "d1",
"Dose 2" = "d2", "Dose 3" = "d3"), selected = "d1", multiple = FALSE, width = 200)
  })
  listEvent <- list("Febrile Convulsions" = "FCONVULS", "Fever" = "FEVER", "Hypotonic
hypo-responsive episode" = "HHE", "Persistent Crying" = "PCRYING", "Somnolence" =
"SOMNOL")
  output$selEvent <- renderUI({
    selectInput(inputId = "selEvent", label = "Select event", choices = listEvent, selected =
"FEVER", multiple = FALSE, width = 300)
  })

  output$ChartRisk <- renderHighchart({
    country <- input$selCountry
    dose <- input$selDose2
    event <- input$selEvent
    load(paste0("data/resultsPOC12_nevent_", country,"__", event,".RData"))
    tempEvent <- as.data.frame(dt3)
    load(paste0("data/resultsPOC12_py_", country, "_.RData"))
    tempPY <- as.data.frame(X)
    names(tempPY) <- c("Wk_id", "PT_d1_rw1","PT_d1_rw2","PT_d1_rw3",
```

```
                    "PT_d2_rw1","PT_d2_rw2","PT_d2_rw3",
                    "PT_d3_rw1","PT_d3_rw2","PT_d3_rw3",
                    "PTbl_d1_rw1","PTbl_d1_rw2","PTbl_d1_rw3",
                    "PTbl_d2_rw1","PTbl_d2_rw2","PTbl_d2_rw3",
                    "PTbl_d3_rw1","PTbl_d3_rw2","PTbl_d3_rw3",
                    "PTbl2_d1","PTbl2_d2","PTbl2_d3",
                    "PTall")
rwEvent <- tempEvent[, paste0("cumN_rw_", dose)]
baseEvent <- tempEvent[, paste0("cumN_bl2_", dose)]
if(event %in% c("PCRYING")) {
  tempRW <- "rw1"
  tempRWChar <- "1 day"
}
if(event %in% c("HHE", "SOMNOL")) {
  tempRW <- "rw2"
  tempRWChar <- "2 days"
}
if(event %in% c("FEVER", "FCONVULS")) {
  tempRW <- "rw3"
  tempRWChar <- "3 days"
}
if(event %in% c("ISR")) {
  tempRW <- "rw7"
  tempRWChar <- "7 days"
}
rwPY <- tempPY[,paste0("PT_", dose, "_",tempRW)] / 365
basePY <- tempPY[,paste0("PTbl2_", dose)] / 365

rwCum <- ave(rwPY, FUN=cumsum)
baseCum <- ave(basePY, FUN=cumsum)

rwInc <- rwEvent / rwCum * 1000
baseInc <- baseEvent / baseCum * 1000
rwIncll <- (( qchisq(.025, df=rwEvent*2)/2)/rwCum)*1000
rwIncul <- (( qchisq(.975, df=rwEvent*2)/2)/rwCum)*1000
baseIncll <- (( qchisq(.025, df=baseEvent*2)/2)/baseCum)*1000
baseIncul <- (( qchisq(.975, df=baseEvent*2)/2)/baseCum)*1000
rwInc[is.nan(rwInc)] <- NA
rwIncll[is.nan(rwIncll)] <- NA
rwIncul[is.nan(rwIncul)] <- NA
baseInc[is.nan(baseInc)] <- NA
```

```
    tempData <- data.frame(wk = tempEvent$wk_id, inc = rwInc, base = baseInc, incll = rwIncll, incul = rwIncul, basell = baseIncll, baseul = baseIncul)
    tempSeq <- seq.Date(from = as.Date("2014-01-01"), to = as.Date("2025-01-01"), by = "week")
    print(str(tempData))
    print(unique(tempData$wk))
    tempData <- tempData[tempData$wk > 0,]
    tempData$Date <- tempSeq[tempData$wk]
    tempData <- tempData[is.finite(tempData$inc),]
    tempData <- tempData[!is.nan(tempData$base),]
    tempData <- tempData[!is.na(tempData$Date),]

    hc <- highchart(type = "stock")
    hc <- hc_legend(hc = hc, enabled = TRUE)
    hc <- hc_tooltip(hc = hc, valueDecimals =  3)
    hc <- hc_navigator(hc = hc, maskFill = "rgba(255,140,0,0.15)", series = list(lineColor = 'rgba(255,140,0,0.5)'))
    hc <- hc_yAxis(hc = hc, title = list(text = "Incidence [cases per 1000 PY]"), opposite = FALSE, turnopposite = TRUE)
    hc <- hc_xAxis(hc = hc, type = "datetime", min = datetime_to_timestamp(as.Date("2014-01-01")), max = datetime_to_timestamp(as.Date("2018-06-07")))
    hc <- hc_add_series(hc = hc, data = tempData, type = "arearange", hcaes(x = Date, low = basell, high = baseul), name = "Baseline Incidence", color = "#e66101", fillOpacity = 0.5)
    hc <- hc_add_series(hc = hc, data = tempData, type = "line", hcaes(x = Date, y = base), name = "Baseline Incidence", linkedTo = ":previous", color = "#e66101")
    hc <- hc_add_series(hc = hc, data = tempData, type = "arearange", hcaes(x = Date, low = incll, high = incul), name = "Risk Window Incidence", color = "#5e3c99", fillOpacity = 0.5)
    hc <- hc_add_series(hc = hc, data = tempData, type = "line", hcaes(x = Date, y = inc), name = "Risk Window Incidence", linkedTo = ":previous", color = "#5e3c99")  # TEMP HACK
    hc <- hc_chart(hc = hc, zoomType = "xy")
    hc <- hc_exporting(hc = hc,enabled=TRUE, buttons=list( contextButton = list( align = 'left', verticalAlign = 'top' ) ) )
    return(hc)
  })

  output$ChartPert <- renderHighchart({
    country <- input$selCountry
    load(paste0("data/resultsPOC12_nevent_", country,"__PERT.RData"))
    tempEvent <- as.data.frame(dt3)
    load(paste0("data/resultsPOC12_py_", country, "_.RData"))
    tempPY <- as.data.frame(X)
```

```
    names(tempPY) <- c("Wk_id", "PT_d1_rw1","PT_d1_rw2","PT_d1_rw3",
                       "PT_d2_rw1","PT_d2_rw2","PT_d2_rw3",
                       "PT_d3_rw1","PT_d3_rw2","PT_d3_rw3",
                       "PTbl_d1_rw1","PTbl_d1_rw2","PTbl_d1_rw3",
                       "PTbl_d2_rw1","PTbl_d2_rw2","PTbl_d2_rw3",
                       "PTbl_d3_rw1","PTbl_d3_rw2","PTbl_d3_rw3",
                       "PTbl2_d1","PTbl2_d2","PTbl2_d3",
                       "PTall")
    rwEvent <- tempEvent[, "N_pert"]
    rwPY <- ave(tempPY[,"PTall"],FUN=cumsum)
    rwInc <- rwEvent / rwPY * 100000
    rwIncll<-(( qchisq(.025, df=rwEvent*2)/2)/rwPY)*100000
    rwIncul<-(( qchisq(.975, df=rwEvent*2)/2)/rwPY)*100000
    rwInc[is.nan(rwInc)] <- NA
    tempData <- data.frame(wk = tempEvent$wk_id, inc = rwInc, incll = rwIncll, incul =
rwIncul)
    tempSeq <- seq.Date(from = as.Date("2014-01-01"), to = as.Date("2025-01-01"), by =
"week")
    tempData$Date <- tempSeq[tempData$wk]
    tempData <- tempData[is.finite(tempData$inc),]

    hc <- highchart(type = "stock")
    hc <- hc_legend(hc = hc, enabled = TRUE)
    hc <- hc_tooltip(hc = hc, valueDecimals =  7)
    hc <- hc_navigator(hc = hc, maskFill = "rgba(255,140,0,0.15)", series = list(lineColor =
'rgba(255,140,0,0.5)'))
    hc <- hc_yAxis(hc = hc, title = list(text = "Incidence [cases per 100.000 PY]"), opposite =
FALSE, turnopposite = TRUE)
    hc <- hc_xAxis(hc = hc, type = "datetime", min = datetime_to_timestamp(as.Date("2014-
01-01")), max = datetime_to_timestamp(as.Date("2018-06-07")))
    hc <- hc_add_series(hc = hc, data = tempData, type = "arearange", hcaes(x = Date, low =
incll, high = incul), name = "Pertussis Incidence", color = "#5e3c99", fillOpacity = 0.5)  #
TEMP HACK
    hc <- hc_add_series(hc = hc, data = tempData, type = "line", hcaes(x = Date, y = inc),
name = "Pertussis Incidence", linkedTo = ":previous", color = "#5e3c99")  # TEMP HACK
    hc <- hc_chart(hc = hc, zoomType = "xy")
    hc  <-  hc_exporting(hc  =  hc,enabled=TRUE,                    buttons=list(
contextButton = list(                  align = 'left',                  verticalAlign = 'top'                  )
)      )
    return(hc)
  })
```

```
output$infoDose <- renderUI({
  div(style = "font-size: 100%; padding-bottom: 50px;", HTML("Number of doses for the selected dose by age groups (as specified above), by calendar time (weekly).<br>"))
})

output$infoCov <- renderUI({
  div(style = "font-size: 100%; padding-bottom: 50px;", HTML("Coverage (%) over time by year-month birth cohort for the selected dose.<br>"))
})

output$infoRisk <- renderUI({
  event <- input$selEvent
  dose <- input$selDose2
  if(dose == "d1") {
    base <- 8
  }
  if(dose == "d2") {
    base <- 12
  }
  if(dose == "d3") {
    base <- 16
  }
  # base <- c(8, 12, 16)[as.numeric(dose)]
  if(event %in% c("PCRYING")) {
    tempRW <- "rw1"
    tempRWChar <- "0-1 day"
  }
  if(event %in% c("HHE", "SOMNOL")) {
    tempRW <- "rw2"
    tempRWChar <- "0-2 days"
  }
  if(event %in% c("FEVER", "FCONVULS")) {
    tempRW <- "rw3"
    tempRWChar <- "0-3 days"
  }
  if(event %in% c("ISR")) {
    tempRW <- "rw7"
    tempRWChar <- "0-7 days"
  }
  event1 <- names(listEvent)[which(listEvent == event)]
  div(style = "font-size: 100%; padding-bottom: 50px;", HTML(paste0("Incidence rate (/1000 person years) and 95% confidence intervals of ", event1, " estimated cumulatively over
```

```
time in a risk window of ", tempRWChar, " after vaccination. The baseline incidence is estimated 10-15 days post-vaccination."))) # . Baseline incidence estimated in children aged ", base-2, " to ", base + 2, " weeks not in the respective risk window.<br>")))
  })


  output$infoBene <- renderUI({
   div(style = "font-size: 100%; padding-bottom: 50px;", HTML("Weekly pertussis incidence (/100.000 person years) and 95% confidence intervals"))
  })


  output$infoTab1 <- renderUI({
   tempHTML <- "<div style='color: #222222;'>This data was constructed from local country reports. Incidence was calculated using the official country population downloaded from PLACEHOLDER at PLACEHOLDER.<br>"
   ageGroups <- paste0(input$selAge1, collapse = ",")
   seroGroups <- paste0(input$selSero1, collapse = ",")
   if(input$selPlot1 == 1) {
    plotInfo <- paste0("Line chart by year depicting the ", input$typeData1, " data for the selected country (", input$selCountry1,"), age groups (", ageGroups, ") and serogroups (", seroGroups,").<br>")
   }
   if(input$selPlot1 == 2) {
    plotInfo <- paste0("Stacked bar chart by year of all serogroups depicting the ", input$typeData1, " data for the selected country (", input$selCountry1,"), and age groups (", ageGroups, ").<br>")
   }
   if(input$selPlot1 == 3) {
    plotInfo <- paste0("Stacked bar chart by age group of all serogroups depicting the ", input$typeData1, " data for the selected country (", input$selCountry1,"), and year.<br>")
   }
   if(input$selPlot1 == 4) {
    plotInfo <- paste0("Bar chart by year and by age group of all serogroups depicting the ", input$typeData1, " data for the selected country (", input$selCountry1,"), and year.<br>")
   }
   tempHTML <- paste0(tempHTML, plotInfo, "This chart were created using R and the following additional packages: shiny, highcharter. Zooming through clicking and dragging.", "</div>")
   HTML(tempHTML)
  })
```

```
    output$disclaimer1 <- renderUI({
      HTML(paste0("<footer><br><br><br><br><br>   Copyright
","&copy;", "2018 ADVANCE</footer>"))
    })

    output$disclaimer2 <- renderUI({
      HTML(paste0("<footer><br><br><br><br><br>   Copyright
","&copy;", "2018 ADVANCE</footer>"))
    })

    output$disclaimer3 <- renderUI({
      HTML(paste0("<footer><br><br><br><br><br>   Copyright
","&copy;", "2018 ADVANCE</footer>"))
    })

    output$disclaimer4 <- renderUI({
      HTML(paste0("<footer><br><br><br><br><br>   Copyright
","&copy;", "2017 ADVANCE</footer>"))
    })

    output$disclaimer5 <- renderUI({
      HTML(paste0("<footer><br><br><br><br><br>   Copyright
","&copy;", "2017 ADVANCE</footer>"))
    })

  })
})
```

UI.R

```
 ###################################################################
#  Data visualization dashboard for ADVANCE POC 1.2        #
#  Developed in R using the 'Shiny' package (and the       #
#  associated shinydashboard package) by RStudio.          #
#==================================================================#
#  Author:   Tom De Smedt                          #
#  Email:    tom.desmedt at p-95 dot com                 #
#==================================================================#

 # Use shinydashboard to create the UI (easy creation of header, navbar, sidebar, menu, tabs, body, etc.)
library(shinydashboard)
library(shinyBS)
library(highcharter)

dashboardHeader <- function(..., title = NULL, disable = FALSE,title.navbar=NULL, .list = NULL) {
 items <- c(list(...), .list)
 #lapply(items, tagAssert, type = "li", class = "dropdown")
 tags$header(
  class = "main-header",
  style = if (disable) "display: none;",
  span(class = "logo", title),
  tags$nav(class = "navbar navbar-static-top", role = "navigation",
           # Embed hidden icon so that we get the font-awesome dependency
           span(shiny::icon("bars"), style = "display:none;"),
           # Sidebar toggle button
           a(href="#", class="sidebar-toggle", `data-toggle`="offcanvas",
             role="button",
             span(class="sr-only", "Toggle navigation")
           ),title.navbar,
           div(class = "navbar-custom-menu",
              tags$ul(class = "nav navbar-nav",
                      items
              )
           )
  )
 )
}
```

```
title.navbar.html <- tags$div(style="margin-left: 30%;
                        color:#29405B;
                        text-align:center;
                        vertical-align:middle;
                        display:inline-block;
                        font-size: 200%;
                        font-weight: 100;
                        padding-left: 5px;
                        padding-top: 13px;",
                        span(style="color: #222222;", "Benefit-Risk Monitoring - Pertussis"),
                        span(style="color: #222222;", uiOutput("header_title"))
)

header <- dashboardHeader(
  title = img(src="logo_ADVANCE.png", style="padding-top:13px;"),
  title.navbar = title.navbar.html
)


### Create dashboard page with several tabs, title bar detailed above (header) and sidemenu for
tab nav and user inputs
shinyUI(
  dashboardPage(
    title="Benefit-Risk Dashboard",skin="blue",

    header,

    dashboardSidebar(width = 235,
                sidebarMenu(id = "tabs",
                        div(style="height:30px;"),
                        menuItem(HTML("   Doses"), tabName = "view1",
icon = icon("bar-chart")),
                        menuItem(HTML("   Coverage"), tabName =
"view2", icon = icon("line-chart")),
                        menuItem(HTML("   Risks"), tabName = "view3",
icon = icon("area-chart")),
                        menuItem(HTML("   Benefits"), tabName = "view4",
icon = icon("map")),
                        menuItem(HTML("   About"), tabName = "view5",
icon = icon("info")),   # Click on tab element to change body to appropriate tab (icon from
glyphicon)
                        HTML("<hr style='margin: 0.1em auto;'><br>"),
```

```
                    uiOutput("selCountry1")
                )  # CLOSE SIDEBARMENU
  ),  #CLOSE DASHBOARDSIDEBAR

  # Dashboard body subdivided in tab displaying the output calculated and rendered in server.R
  dashboardBody(
   # Use style described in www/dashboard.css
   tags$head(tags$link(rel = "stylesheet", type = "text/css", href = "dashboard.css"),
          tags$script(src = "https://vac4euorgdev.wpengine.com/wp-content/plugins/advanced-iframe/js/ai_external.js")),

    tabItems(
     # DOSES
     tabItem(value = 1, tabName = "view1",
          box(width = 3, uiOutput("selDoseCov1")),
          box(width = 9, uiOutput("selAgeCov1")),
          box(width = 12,
            highchartOutput("ChartDose"),
            uiOutput("infoDose")),
          # Disclaimer text added at the bottom of the page
          box(width = 12, title = "", solidHeader = TRUE, collapsible = FALSE, color = "white", htmlOutput("disclaimer1"))
     ),  # CLOSE DOSES TAB

     # COVERAGE
     tabItem(value = 2, tabName = "view2",
          box(width = 3, uiOutput("selDoseCov2")),
          box(width = 9, uiOutput("selYM")),
          box(width = 12,
            highchartOutput("ChartCovDoB"),
            uiOutput("infoCov")),
          # Disclaimer text added at the bottom of the page
          box(width = 12, title = "", solidHeader = TRUE, collapsible = FALSE, color = "white", htmlOutput("disclaimer2"))
     ),  # CLOSE COVERAGE TAB

     # RISKS
     tabItem(value = 3, tabName = "view3",
          box(width = 3, uiOutput("selDose2")),
          box(width = 9, uiOutput("selEvent")),
          box(width = 12, highchartOutput("ChartRisk"),
            uiOutput("infoRisk")),
```

```
            # Disclaimer text added at the bottom of the page
            box(width = 12, title =  "", solidHeader = TRUE, collapsible = FALSE, color =
"white", htmlOutput("disclaimer3"))
        ),  # CLOSE RISKS TAB

        # BENEFITS
        tabItem(value = 4, tabName = "view4",
            box(width = 12, highchartOutput("ChartPert")),
            box(width = 12, uiOutput("infoBene")
                #,
               # tags$style(type="text/css", "#selCountry1 {display:center-align;")
            ),
            # Disclaimer text added at the bottom of the page
            box(width = 12, title =  "", solidHeader = TRUE, collapsible = FALSE, color =
"white", htmlOutput("disclaimer4"))
        ),  # CLOSE CHARTS TAB

        # Tab with About
        tabItem(value = 5, tabName = "view5",
            box(width = 12,
               h2("COMING SOON")),
            # Disclaimer text added at the bottom of the page
            box(width = 12, title =  "", solidHeader = TRUE, collapsible = FALSE, color =
"white", htmlOutput("disclaimer5"))
        )
      )  # CLOSE TABITEMS
    )  # CLOSE DASHBOARDBODY
  )  # CLOSE DASHBOARDPAGE
)  # CLOSE SHINYUI
```