DAPHNE

Data-as-a-service platform for healthy lifestyle and preventive medicine

610440

D5.3 Big Data Services Design

Lead Author: Mario Recio (Treelogic)

With contributions from: Jose Antonio Sánchez (Treelogic), Roni Ram (IBM), Ross Little (ATOS)

Reviewer: Roni Ram (IBM)

Deliverable nature:	Report (R)
Dissemination level: (Confidentiality)	Public (PU)
Contractual delivery date:	30/04/2015
Actual delivery date:	30/04/2015
Version:	1.0
Total number of pages:	49
Keywords:	"big data", "data consumer", "data consumer portal"

Abstract

This document describes the design of the big data services that will use DAPHNE data cloud of collective nutritional, activity, behaviour, psychological wellbeing and health markers information. Three different services have been designed to motivate the use of DAPHNE information by researchers. These services have been defined taken into account the scenario requirements of WP1, the DAPHNE's architecture established in WP2 and considering the Data-as-a-service API offered by the data cloud and all the Privacy and Security guidelines, defined both in WP6.

Executive summary

The goal of this document is to define the big data services that DAPHNE is going to offer. It is the result of all the studies and work carried out in *T5.3 Big Data Services Definition and Design*. These services are going to be hosted in a web application defined in the platform architecture as “Data Consumer Portal”.

End user involvement and collection of requirements is described in this deliverable.

The proposed services are:

- **DAPHNE datasets download**: This service allows Data Consumers to download datasets containing DAPHNE user’s information so it can be analysed by their own systems. These datasets are useful for researchers so they can take it into account in their studies and models.
- **DAPHNE parameters visualization**: This service has been designed for those Data Consumers with no analysis needs over the DAPHNE information. It allows to visualize user’s information offering graphics of multiple parameters.
- **DAPHNE data analysis**: This service allows Data Consumers to apply statistical operations over DAPHNE users’ datasets, combine and export them. This service has been designed exploiting the most important features of R processing language.

For each of these services, the document includes:

- **List of functional requirements**
- **Use cases identification**
- **Final services design**

In addition, general sections related to the technical/security aspects close the document with some notes about the development of these services.

Document Information

IST Project Number	610440	**Acronym**	DAPHNE
Full Title	Data-as-a-service platform for healthy lifestyle and preventive medicine		
Project URL	http://www.DAPHNE-fp7.eu/		
Document URL			
EU Project Officer	Mr. Eduardo González Otero		

Deliverable	**Number**	D5.3	**Title**	Big Data Services Design
Work Package	**Number**	WP5	**Title**	Data Exploitation: services & applications

Date of Delivery	**Contractual**	M18	**Actual**	M18
Status	version 1.0		final ■	
Nature	prototype □ report ■ demonstrator □ other □			
Dissemination level	public ■ restricted □			

Authors (Partner)	Treelogic			
Responsible Author	**Name**	Mario Recio	**E-mail**	mario.recio@treelogic.com
	Partner	Treelogic	**Phone**	mariorecioprado (Skype)

Abstract (for dissemination)	This document describes the design of the big data services that will use DAPHNE data cloud of collective nutritional, activity, behaviour, psychological wellbeing and health markers information. Three different services have been designed to motivate the use of DAPHNE information by researchers.
Keywords	Big data, data consumer, data consumer portal

Version Log			
Issue Date	**Rev. No.**	**Author**	**Change**
01/12/2014	V0.1	Mario Recio (Treelogic)	Table of Contents
15/12/2014	V0.2	Mario Recio (Treelogic)	Update of TOC
12/01/2015	V0.3	Jose Antonio Sánchez (Treelogic)	First requirements added
18/01/2015	V0.4	Roni Ram (IBM)	Requirements added
06/02/2015	V0.5	Mario Recio (Treelogic)	Use cases added
25/02/2015	V0.6	Jose Antonio Sánchez (Treelogic)	Services first design added
26/02/2015	V0.7	Roni Ram (IBM)	Services completed
03/03/2015	V0.8	Jose Antonio Sánchez (Treelogic)	Technical aspects added
10/03/2015	V0.9	Mario Recio (Treelogic)	Services last design added
21/03/2015	V0.91	Ross Little (ATOS)	Security aspects completed
31/03/2015	V1.0	Mario Recio (Treelogic)	First version

Table of Contents

List of figures

List of tables

Abbreviations

DC: Data Consumer

IAM: Identity & Access Manager

ICT: Information and Communication Technologies

JSON: JavaScript Object Notation

PHI: Personal Health Information

PHS: Personal Health Services

REST: REpresentational State Transfer

XML: eXtensible Markup Language

1 Introduction

The DAPHNE platform is based on individuals, from the younger to the elder population, to become co-producers of their health and maintain good health status. A focus on the physical detection and prevention of conditions related with sedentariness and unhealthy habits are set, with the aim to reduce obesity incidence and associated problems.

For that purpose, healthy information from individuals will be stored in the platform, processed, originating a large amount of information. This information will be available, among other purposes, to be consumed as a global big data source, for different stakeholders that could use global market data for clinical research or for orientation in their businesses.

DAPHNE Data-as-a-Service targets customer groups who seek Big Data for analysis and research purposes (universities, insurance companies, pharmaceutical companies, retailers, government institutions, amongst others). In this deliverable, the collection of requirements from different end users is shown.

Big data services may be combined with the *Data-as-a-Service API* [2] to provide long term, scalable, highly available and secure storage capabilities for data made available from various sources as sensors, wearable devices and mobile applications. While complying with privacy laws/directives, it offers the data for personal care and for big data analytics by third parties.

Big data services will be considered in both DAPHNE deployments: “Self care users” and “Patients”. According to D2.7 [1], these deployments will follow the following architectures:

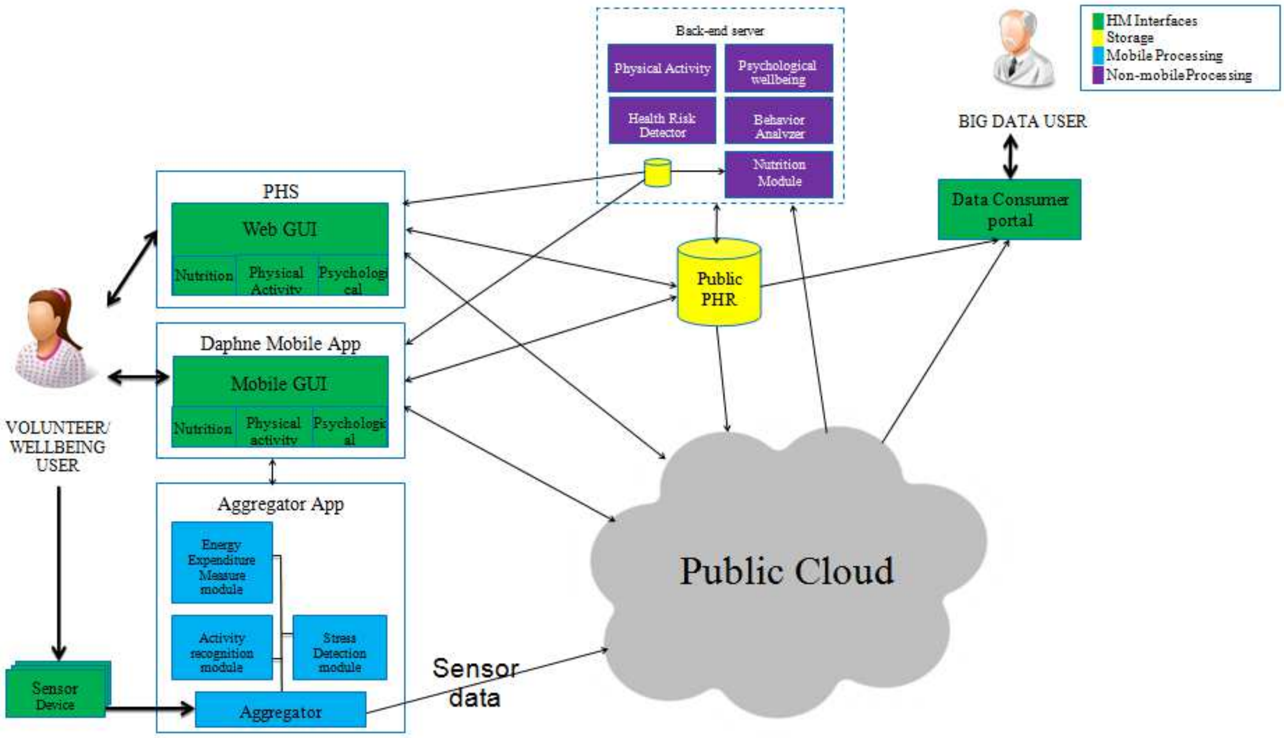


Figure 1 DAPHNE functional blocks diagram for Installation A (self care users)

The Data Consumer Portal will consume data offered by Data-as-a-Service API. Although this element is deeply described in D6.2 [2], the big data services are going to consider information from the **Public Cloud** and the **Public PHR** from both scenarios.

Additionally, data from the private PHR can be exported to the Public PHR upon the approval of an ethical committee.

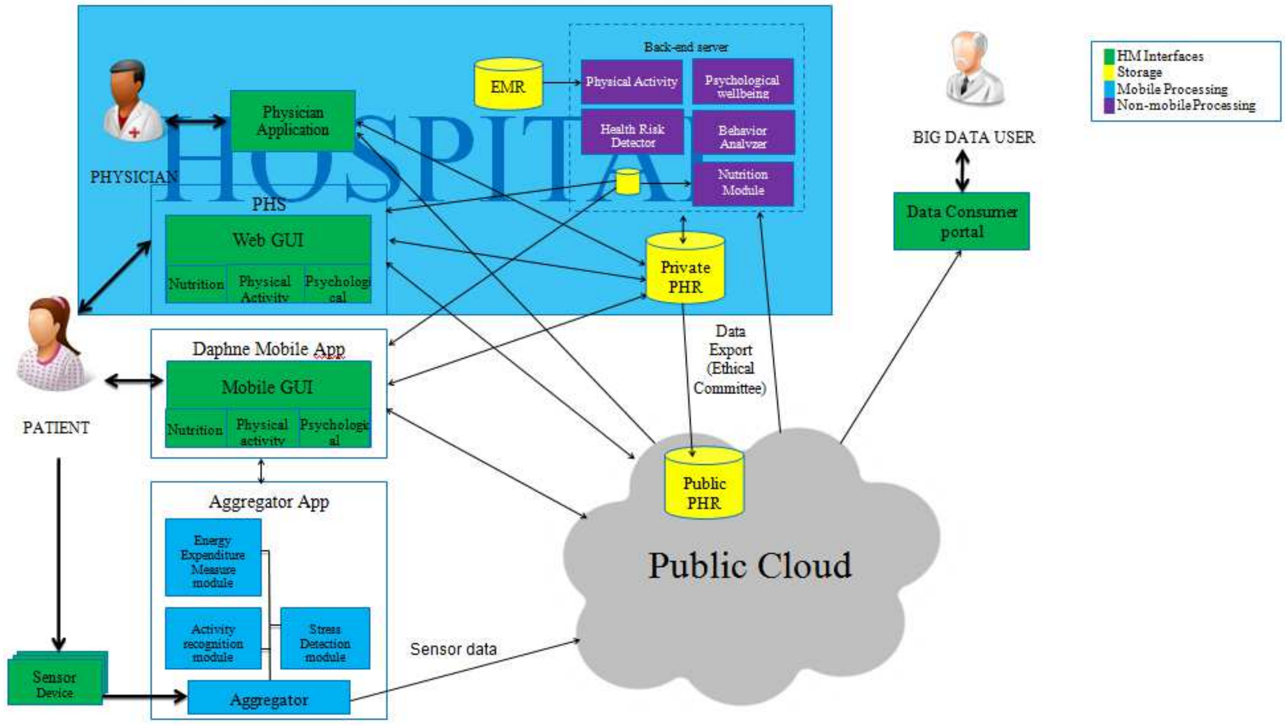


Figure 2 DAPHNE functional blocks diagram for Installation B (patients)

1.1 Relation with other project tasks

The design of the big data services is not an isolated task; it is related to multiple tasks in the DAPHNE project. This fact is shown in the following figure.

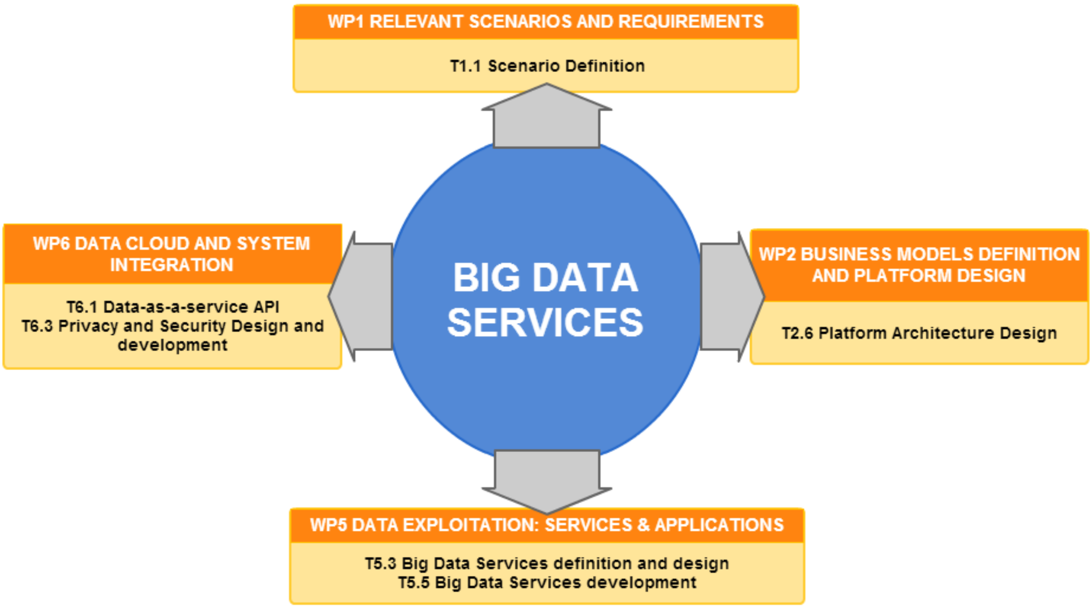

WP1 RELEVANT SCENARIOS AND REQUIREMENTS
T1.1 Scenario Definition
WP6 DATA CLOUD AND SYSTEM INTEGRATION
T6.1 Data-as-a-service API
T6.3 Privacy and Security Design and development
BIG DATA SERVICES
WP2 BUSINESS MODELS DEFINITION AND PLATFORM DESIGN
T2.6 Platform Architecture Design
WP5 DATA EXPLOITATION: SERVICES & APPLICATIONS
T5.3 Big Data Services definition and design
T5.5 Big Data Services development

Figure 3 Big data services relation with other tasks

In particular, this relation can be specified in the following table:

Table 1 Big data services relation with other tasks

WP	Task	Description
WP1	T1.1 Scenario Definition	This task defines all the scenarios considered in DAPHNE.
WP2	T2.6 Platform Architecture Design	In this task, all the components forming the DAPHNE platform are described. This design is defined taking the Big Data services into account.
WP5	T5.3 Big Data Services definition and design	In this task all the big data services must be defined. This deliverable is result of this task.
	T5.5 Big Data Services development	All the services defined in T5.3 are going to be developed in this task.
WP6	T6.1 Data-as-a-service API	In this task, the API that the data consumer portal is going to consume is defined.
	T6.3 Privacy and Security design and development	As the rest of the components forming DAPHNE, the data consumer portal must communicate with the security components in order to guarantee authentication and authorization of the DC users.

1.2 Structure of this deliverable

First in the document a section about the definition of the big data services is introduced. It is "*Section 2: Big Data Services definition and functional requirements*". It will describe the actors who interact with the big data services, it will include a brief general description of each of the presented services and it will conclude with their functional requirements and the use cases for each big data service.

Then, "*Section 3:* ", will specify all technical facts that must be taken into account in the development phase, followed by "*Section 4:* " which defines the security considerations related to the big data services.

After this section, and closing the document, the proposed design of the web portal containing all of these big data services will be presented, in "*Section 5: Data consumer portal design*", describing final screens designs and their functionalities.

2 Big Data Services definition and functional requirements

This document defines a first approach to the DAPHNE big data services, strongly focused on the DC role and its requirements. This deliverable is focused on the definition of the following services:

- Dataset Download
- Data Analysis
- Parameters Visualization

These services are grouped and accessed by DCs through a web portal, the Data Consumer Portal:

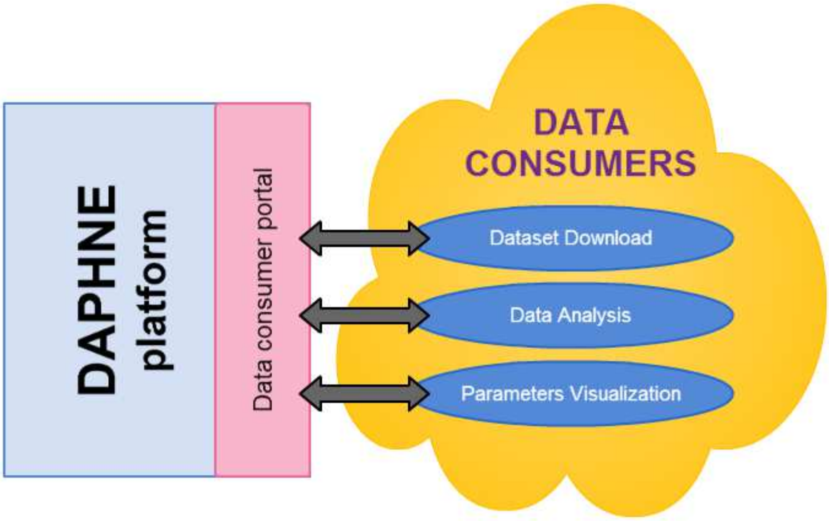


Figure 4 DAPHNE DC services

2.1 Definition of actors

Although actors involved in DAPHNE platform are described in D2.7 [1], this deliverable is focussed on the Data Consumer users (DC). DC is the term that will be used for the big data services users. DC are deeply described in the next section.

2.1.1 Data Consumer

This actor is going to consume the big data services offered by DAPHNE through the data consumer portal. Although there is only one Data Consumer role for the three big data services, these services are oriented to three types of data consumer users, with different skills and needs.

Table 2 Different Data Consumers types

DC needs \ big data service	Dataset Download	Data Analysis	Parameters Visualization
No processing needs and no mathematical knowledge			X
Processing needs and no mathematical knowledge	X		X
Processing needs and mathematical knowledge	X	X	X

2.2 Services Description

This section defines a first approach to the three services introduced above that the data consumer portal is going to offer. It is the first step in the design phase, in order to get an idea of the big data services and get their use cases and functional requirements.

2.2.1 Datasets Download

This service communicates with the DAPHNE Data-as-a-service API (T6.1, [2]) so DC users can download datasets containing anonymized information about DAPHNE's wellbeing users and patients. This information is useful for researchers so they can take it into account in their studies and models.

This service allows DCs to download information from the DAPHNE platform in a comfortable way. DC can configure the download according to:

- Selection criteria of users:
    - Age
    - Gender
    - Weight/Height
    - Users characteristics (Smoker, Pre-diabetic, Diabetic)
- Parameters to download grouped by:
    - Anthropometrics
    - Health Markers
    - Psychological Wellbeing
    - Physical Activity
    - Behaviour Data
    - Nutritional Information
- Dataset file format
    - JSON
    - XML
    - CSV

All this information is going to be hosted in DAPHNE's public Data Cloud and it is going to be consumed by the data consumer portal through the Service-as-a-data API defined in [2].

2.2.2 Parameters Visualization

This service is useful for those DC who want to study the DAPHNE users' health parameters, intake/ activity/mental behaviours and their relation with age, gender, habits and other user conditions in a visual way. There is no dataset downloading.

This service allows DC to analyze data in a visual way, grouped by the PHS categories:

- Anthropometrics
- Health Markers
- Psychological Wellbeing
- Physical Activity
- Behavior Data
- Nutritional Information

Visualization will be based on interactive charts able to show information in multiple time scales combined with other patient's parameters, such as age, habits, gender...

2.2.3 Data Analysis

This service combines DAPHNE data with R programming language [3]. R has been chosen as the data analysis language after a comparison study of some data analysis alternatives languages. The following table contains the results of this study:

Table 3 Data analysis languages comparison

Name	Advantages	Disadvantages	Open Source	Typical Users
R [3]	Library support; visualization	Steep learning curve	Yes	Finance; Statistics
Matlab [13]	Elegant matrix support; visualization	Expensive; incomplete statistic support	No	Engineering
SciPy/ NumPy/ Matpltlib [14]	Python (general-purpose programming language)	Immature	Yes	Engineering
Excel [15]	Easy; visual; flexible	Large datasets	No	Business
SAS [16]	Large datasets	Expensive; outdated programming language	No	Business; Government
Stata [17]	Easy statistical analysis		No	Science

R has been chosen as the data statistic language because of his advantages:

Open source and library support: R is a extended language in statistical studies, and it is open source, so there are multiple packages developed by the R community that can be found in the internet. Actually, most important are listed in the CRAN project (The Comprehensive R Archive Network) [18][19]. These packages add new operations and functions to the R environment in order to improve R experience. This fact allows R users to re-use functions and operations already developed by other users.

Oriented to statistics: Although users have implemented a huge list of R packages that extend its functionalities to multiple study fields, R was initially oriented to data analysis. Mathematical operations and statistical functionalities are its main advantage against other data analysis languages.

An annex containing a brief description of R language and its environment is attached at the end of this document. In addition, some commands have been included in order to facilitate the service validation/testing in the development phase.

This service may provide an online tool to operate, analyse and represent DAPHNE information with a software environment for statistical computing and graphics, based on R. It is designed for those researchers with statistic processing needs over DAPHNE data. This service will be based on R language so these researchers must have a strong experience with this mathematical language.

The most relevant functions available in R language are presented in the next diagram

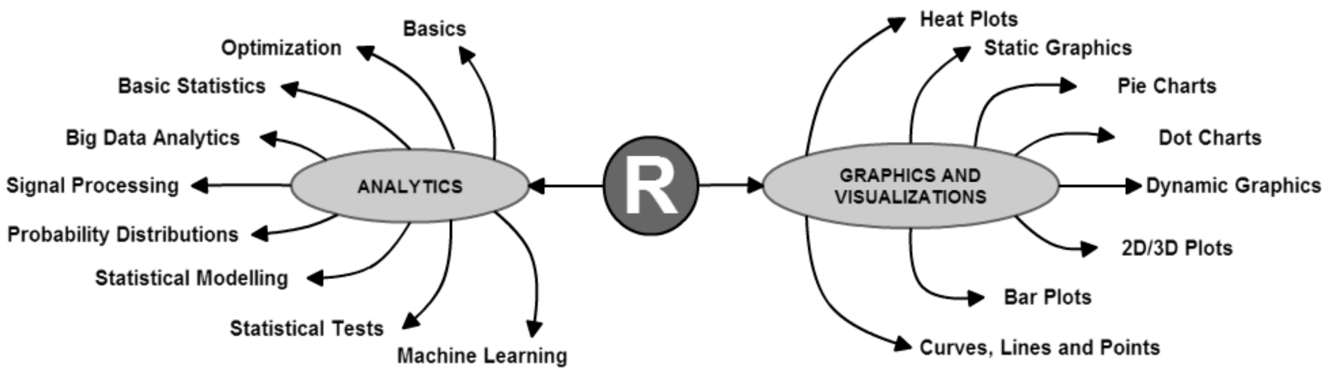


Figure 5 R language features

R will allow DCs to analyze, combine and plot DAPHNE data in the Data Consumer portal, and consider these results in their researches and studies with no data download involved.

2.3 Establishing Requirements

This section describes a complete list of requirements that the services should satisfy. The requirements exposed are described in natural language and are the outcome of several open and close interviews with DAPHNE system stakeholders and the study of the requirements specified along the deliverables of WP1 and D2.4 [4].

The requirements listed in this section are prioritized using the following rules:

- **HIGH**: the requirement will be implemented
- **MEDIUM**: the requirement will be implemented if time and resources allow
- **LOW**: the requirement is discarded at the moment, but in the future could be taken into account

2.3.1 Functional Requirements

This section decribes all the functional requirements from the point of view of final users. Requirements from the technical/security point of view are not considered in this section, they are described in further sections "*3.Technical aspects*" and "*4.Security aspects*". Functional requirements are only focussed on user's experience and user needs, not in technical/security details

The following table contains all the functional requirements identified in the service definition phase. These services are going to be available for data consumers and researchers through a web portal, the "Data Consumer Portal".

Table 4 Big data services requirements table

ID	Description	Priority
Data Consumer Portal		
FR1. Registration and authentication		
FR1.1	A user may be able to register as a new DC in DAPHNE	High
FR1.2	New DC shall sign up in DAPHNE by providing the required information according to DAPHNE security and privacy policies (such as name and position of the person in charge of receiving and processing the requested big	High

	data, purpose of data collection, type of organization, etc).	
FR1.3	DC portal shall require the authentication of the DC credentials before any operation.	High
FR2. Datasets Download		
FR2.1	DC should be able to download from DC portal large anonymized datasets based on DAPHNE users' data for specific research purposes.	High
FR2.2	DC shall choose the type of data to download in the DC portal.	High
FR2.3	DC may be able to choose the type of data to download from the data groups defined in the PHS: - Anthropometrics - Health Markers - Physical Activity - Nutrition Activity - Behavior Analyzer - Psychological Wellbeing	High
FR2.4	Downloaded dataset may be formatted in a well-known data format standard.	High
FR2.5	DC portal shall allow user to download dataset in the following formats: - JSON - XML - CSV	High
FR2.6	DC shall be able to define filtering conditions for dataset download, based on time, users' anthropometrics and habits.	Medium
FR2.7	DC can choose between different basic operations to apply on dataset (AVG, MIN, MAX)	Medium
FR3. Parameters Visualization		
FR3.1	DC portal may support the visualization of DAPHNE data in multiple graphs grouped by PHS data groups: - Anthropometrics - Health Markers - Physical Activity - Nutrition Activity - Behavior Analyzer - Psychological Wellbeing	High
FR3.2	DCs may be able to interact with the displayed graphs by getting individual values when a point of the graph is clicked.	High

FR3.3	The time interval of the graphs can be specified by DC	Medium
FR3.4	Given a graphDC can choose between different basic operations to apply on data (AVG, MIN, MAX)	Medium
FR4. Data Analysis		
FR4.1	DC portal shall provide an online data analysis tool.	High
FR4.2	DC shall support statistics operations in order to process DAPHNE data according to R language.	High
FR4.3	DC shall visualize the outcome of the analysis as numeric results, series and/or graphs.	High
FR4.4	DC shall provide the means to edit, visualize, export and import datasets and variables used during the data analysis.	Medium
FR4.5	DC may be able to upload and execute his own R scripts in the data consumer portal.	Medium

2.3.2 Analysis of requirements

A progressive analysis in the list of requirements was performed in order to categorize them and achieve the functional homogenization which is presented on this document. The relations and dependencies between requirements were analysed, the completeness was validated and it was confirmed that the requirements really meet the needs of the end users.

2.4 End user's feedback

The requirements specified in this document are the result of interviews and meetings of the analysis team with clinical partners, patients and potential end users. The following table summarises the end users involved in the requirements definition:

End user	Type of End User	Interests
OPBG	Clinical partner	Population studies of the effect of lifestyle on health (focused on children). Interested in studying the evolution of their children patients as a group. Interested in carrying out advanced clinical studies with their children patients, with a powerful tool like Daphne
Maccabi Healthcare / Nevet	Clinical partner / SME	Population studies of the effect of lifestyle on health Interested in studying the evolution of their patients as a group Interested in carrying out advanced clinical studies
University of Leeds	University / Research	Interested in studying the effect of lifestyle, physical activity and sedentarism on healthcare Scientific studies
UPM	University / Research	Interested in studying the effect of stress and physical activity

		on healthcare Scientific studies
Dreamgenics [6]	SME	Interested in data analytic techniques. Commercial interests
Lambdoop [7]	SME	Interested in technologies for processing large volumes of data Commercial interests

During the development of Daphne Big Data services it is foreseen the involvement of other external end users, to verify the technical developments and check possible scientific or commercial interests, in line with the Daphne general exploitation plan (D8.1)

2.5 Use case analysis

In this section, the use cases obtained from the study and analysis of the functional requirements are going to be described. They are divided to the following categories according to their functionalities.

Registration and authentication: Data Consumer Portal must allow users to perform registration and authentication actions in order to manage these processes.

Datasets Download: Data Consumer Portal must allow DC to download datasets containing anonymized data about DAPHNE users. This service must allow DC to choose the selection criteria of target users, parameters to be downloaded, as well as the dataset format. These datasets are oriented to be processed by researchers' analysis systems.

Parameters Visualization: Data Consumer Portal may allow DC to check graphs and diagrams related to DAPHNE users and their information managed by the platform.

Data Analysis: Data Consumer Portal may allow DC to define data analysis operations in R language, over DAPHNE datasets.

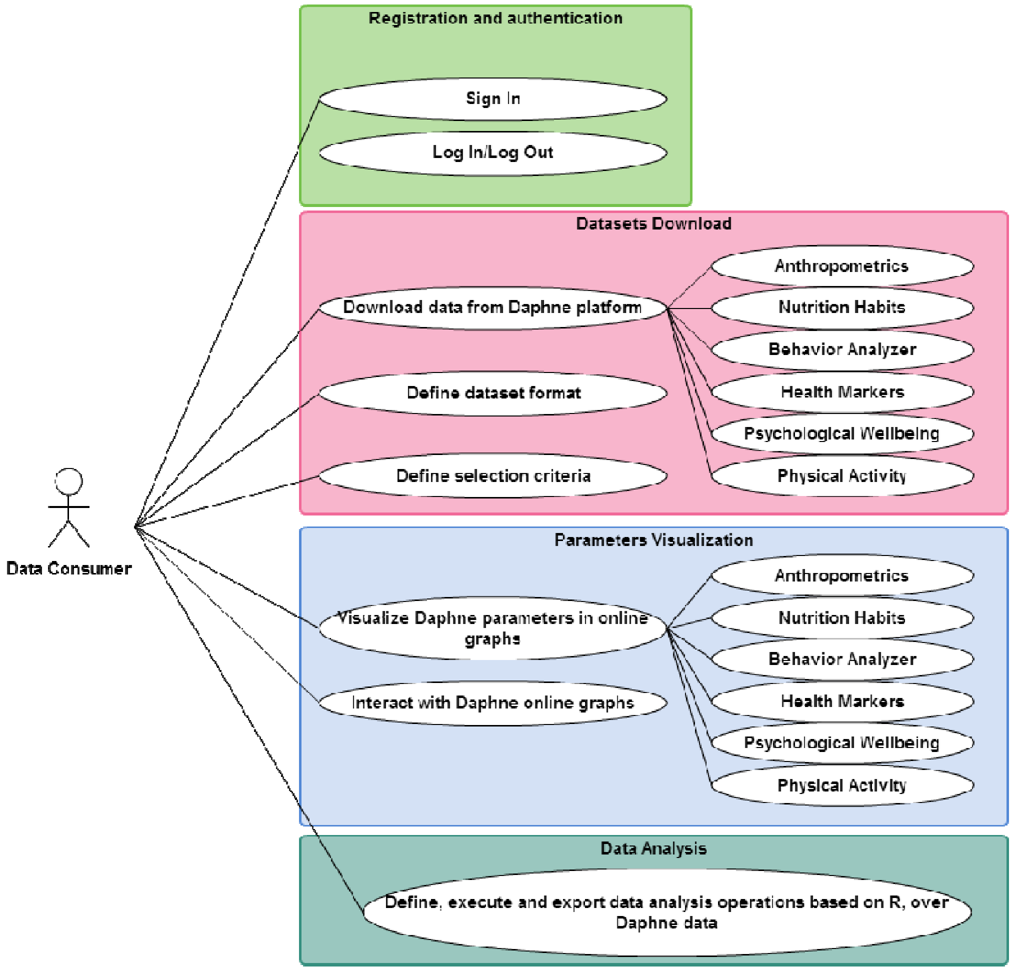


Figure 6 Data Consumer Portal Use Cases

2.5.1 Registration and authentication use cases

ID	UC1.1	Name	Sign In
Description	A user may be able to sign in as a new DC in DAPHNE		
Pre-Condition	DC must have access to the Data Consume portal login screen		
Sequence	1- User visits the login page of the Data Consumer Portal 2- He clicks on the "New Data Consumer account" link 3- He sends an email to the DAPHNE ethical committee with the following information: - The purpose for requesting the access to the big data services. - Name and position of the person or body in charge of receiving and processing the requested information from the big data services. - Data consumer agreement according to terms and conditions to keep the big data information confidential and used for only the stated purpose. - The legal basis for the request. 4- DAPHNE ethical committee evaluates his request and accepts/denies it 5- He receives an email accepting or denying his new DC account request		
Errors	If the Data Consumer portal is not online, then he can not access the "New Data Consumer account" link.		
Post-Condition	DC receives his login credentials and he is able to log in the Data Consumer Portal as a DC		
Notes			

Name	UC1.2	Name	Log in
Description	DC logs in to the DAPHNE platform through the Data Consumer Portal		
Pre-Condition	DC must be registered in DAPHNE as a DC		
Sequence	1- DC accesses to the Data Consumer Portal URL 2- DC introduces his username and password in the log in page 3- The system checks DC credentials 4- DAPHNE allows DC to enter the Data Consumer Portal		
Errors	If the user is not registered in DAPHNE as a DC, he will not be able to enter the Data Consumer Portal		
Post-Condition	DC is logged in the Data Consumer Portal		
Notes			

Name	UC1.3	Name	Log out
Description	DC logs out from the DAPHNE platform through the Data Consumer Portal		
Pre-Condition	DC must be registered in DAPHNE as a DC and logged in the Data Consumer Portal		

Sequence	1- DC clicks on the "log out" button 2- The system logs out the DC in the web portal
Errors	The system is not able to log out DC. The process is finished and the system logs out the user through the security components
Post-Condition	DC is not logged in the Data Consumer Portal
Notes	

2.5.2 Datasets download service use cases

Name	**UC2.1**	**Name**	**Download Data from DAPHNE platform**
Description	DC downloads a dataset containing anonymized data of DAPHNE users, grouped by categories, parameters or results.		
Pre-Condition	DC is registered in DAPHNE as a DC and logged in the Data Consumer Portal		
Sequence	1- DC fills in a web form about the data he is going to download (selection criteria and required parameters) 2- Once the web form has been filled, DC pushes the "Download" button 3- The Data Consumer Portal gets the dataset from the data cloud 4- The DC gets a file containing the downloaded dataset in his web explorer		
Errors	The system is not able to connect with the data cloud		
Post-Condition	DC has downloaded a new dataset containing anonymized data about DAPHNE users		
Notes			

Name	**UC2.2**	**Name**	**Define dataset format**
Description	DC downloads a dataset containing anonymized data of DAPHNE users in a specific file format		
Pre-Condition	DC is registered in DAPHNE as a DC and logged in the Data Consumer Portal		
Sequence	1- DC fills in a web form about the data he is going to download (selection criteria and required parameters) 2- DC chooses the format of the dataset to be downloaded from a list in the "Download" screen 3- The anonymized dataset is retrieved from the data cloud 4- The Data Consumer Portal formats the dataset according to the chosen format 5- Dataset is ready to be downloaded by the DC user through his web explorer		
Errors	The system is not able to connect with the data cloud		
Post-Condition	DC has downloaded a new dataset containing anonymized data about DAPHNE users in the format he has previously chosen in the Download screen		
Notes			

2.5.3 Parameters visualization service use cases

Name	UC3.1	Name	Visualize DAPHNE parameters in online graphs
Description	Data Consumer Portal may be able to display some graphs related to DAPHNE users' parameters. DC can visualize and manipulate them according to his preferences. These graphs will be grouped by - Anthropometrics - Health Markers - Physical Activity - Nutrition Activity - Behavior Analyzer - Psychological Wellbeing		
Pre-Condition	DC must be registered in DAPHNE as a DC and logged in the Data Consumer Portal		
Sequence	1- DC clicks on the "Parameters Visualization" view 2- DC fills in a web form about the data he is going to visualize (selection criteria and required parameters) 3- The anonymized dataset is retrieved from the data cloud 4- Data Consumer Portal shows the anthropometrics graphs 5- DC can navigate between the DAPHNE data groups in order to visualize the graphs: - Anthropometrics - Health Markers - Physical Activity - Nutrition Activity - Behaviour Analyzer - Psychological Wellbeing 6- DC can use these results in his researches		
Errors	The system is not able to connect with the data cloud		
Post-Condition	DC visualizes DAPHNE users parameters in multiple graphs, grouped by DAPHNE data groups		
Notes			

Name	UC3.2	Name	Interact with DAPHNE online graphs
Description	Data Consumer Portal may be able to display some graphs related to DAPHNE users' parameters. DC can interact and manipulate each of them in order to get the graphs values or change data resolution		
Pre-Condition	DC must be registered in DAPHNE as a DC and logged in the Data Consumer Portal		
Sequence	1- DC clicks on the "Parameters Visualization" view 2- DC fills in a web form about the data he is going to visualize (selection criteria and required parameters)		

	3- The anonymized dataset is retrieved from the data cloud 4- Data Consumer Portal shows the anthropometrics graphs 5- DC can navigate between the DAPHNE data groups in order to visualize the graphs: - Anthropometrics - Health Markers - Physical Activity - Nutrition Activity - Behaviour Analyzer - Psychological Wellbeing 6- DC can click on the points forming a graph in order to check parameters values, or change data resolution
Errors	The system is not able to connect with the data cloud
Post-Condition	DC changes data resolution in graphs and checks values over them
Notes	

2.5.4 Data analysis use cases

Name	**UC4.1**	**Name**	**Define and execute data analysis operations based on R language over DAPHNE data**
Description	Data Consumer Portal supports a data analysis view where DC can define data analysis functions over DAPHNE datasets in order to combine and process DAPHNE information in a more "statistical" way		
Pre-Condition	DC must be registered in DAPHNE as a DC and logged in the Data Consumer Portal		
Sequence	1- DC clicks on the "Data Analysis" view. 2- DC fills in a web form about the data he is going to analyze (selection criteria and required parameters) 3- The anonymized dataset is retrieved from the data cloud 4- DC defines R commands to be executed over the DAPHNE parameters 5- The Data Consumer Portal operates R commands and calculates the final result 6- The results are shown to the DC 7- DC is able to export these results if it is necessary		
Errors	The system is not able to connect with the data cloud. DC does not define correct R commands		
Post-Condition	DC gets the results of the data analysis defined in R, applied over DAPHNE information		
Notes			

2.6 Requirements vs use cases matrix

The use cases described in the previous section, meet with all the functional requirements identified for the data consumer portal. This is reflected in the "Functional Requirements VS Use Cases" matrix.

Table 5 Functional requirements vs use cases matrix

Functional Requirement VS Use Case	UC 1.1	UC 1.2	UC 1.3	UC 2.1	UC 2.2	UC 3.1	UC 3.2	UC 4.1
FR 1.1	X							
FR 1.2	X							
FR 1.3		X	X					
FR 2.1				X				
FR 2.2				X	X			
FR 2.3				X	X			
FR 2.4					X			
FR 2.5					X			
FR 2.6				X	X			
FR 2.7				X	X			
FR 3.1						X		
FR 3.2							X	
FR 3.3						X	X	
FR 3.4						X	X	
FR 4.1								X
FR 4.2								X
FR 4.3								X
FR 4.4								X
FR 4.5								X

3 Technical aspects

According to the functionalities described in this document, all the big data services are concentred in the development of the data consumer portal able to interact with the DAPHNE's data cloud in order to offer user's information in a properly format, visualization or analysis. This section specifies the technical aspects related to the development of the data consumer portal.

3.1 Technical requirements

In order to offer the previously defined big data services, the data consumer portal must meet the following requirements:

- Hardware
    - Processor: Dual Core with at least 1 GHz
    - Memory: 4 GB
    - Hard disk: 50Gb
- Software
    - OS: Ubuntu
    - Tomcat 8.0
- Communication
    - SSH (port 22) service open
    - FTP service and Remote Desktop Connection open
    - Connectivity with DAPHNE's data cloud and DAPHNE's security elements

3.2 Technical interaction with other DAPHNE components

In section 1.1 "Relation with other project tasks", the relations between the big data services and the other tasks carried out in DAPHNE project are described. Once the services have been defined, the interactions between these components have to be defined:

Table 6 Technical interactions with other DAPHNE components

Component	Description
Data Cloud	Data consumer portal interacts with DAPHNE Data Cloud by the use of the API defined in *T6.1.2 Data-as-a-service API for services integration.* This means that the big data services described in this deliverable needs to follow the API definition specified in *D6.2 Data-as-a-service API* [2]
Security Components	Data consumer portal must authenticate and authorize DC users before they access the DAPHNE platform, in order to guarantee that only authorized DC can access to the data consumer portal and its services. This will be made through the IAM component. More information about this components can be found in *D6.5 Privacy and Security design guidelines for DAPHNE (ongoing). Ángel Palomares Pérez, DAPHNE* [10]

3.2.1 Integration with DAPHNE DataCloud (Data-as-a-Service API)

The anonymized datasets are retrieved from DAPHNE Data Cloud, the storage of user data in the cloud which includes the Public Personal Health data Repository (PHR) and the Public Cloud.

Component	Description
Public PHR	The Personal Health data Repository (PHR) is a secure repository for all the data that according to the privacy and security regulations cannot leave the

	care giver facilities. To keep the same architecture approach for both installation A and B, PHR is being used in the same way whether the user acts a Patient or a Wellbeing/Volunteer user role; the only difference is that the PHR is deployed in the cloud and not inside the hospital.
Public Cloud	DAPHNE Public Cloud is the component that provides long term, scalable and secure storage capabilities for data coming from various sources as sensors, wearable devices and mobile applications. While complying with privacy laws/directives, it offers the data for personal care and for big data analytics by third parties. DAPHNE Public Cloud will be developed on top of IBM Bluemix[1]™ which is the IBM open cloud platform that hosted on a server in London and provides mobile and web developers access to IBM software for integration, security, transaction, and other key functions, as well as software from business partners.

In terms of the Data-as-a-Service API, the data consumer portal is involved in the following way:

The anonymized data that is stored both in the Public Cloud and in the Public PHR (i.e. it does not identify the person that it belongs to) can be queried by the data consumer portal for specific criteria of users (never for a specific individual). The types of data that are consumed are different health and wellness parameters that define a user profile with the additional data provided by the intelligent algorithms and the physicians.

The data is stored and retrieved from DAPHNE Data Cloud by the Data-as-a-Service API to be consumed by the Data Consumer Portal in the following way:

1. Data Consumer Portal get DC user needs in any of the three services offered
2. Retrieve request is generated including a security token
3. The security token is validated (for authentication and authorization) and an audit message is added to the audit log
4. The profile of the big data user who invokes the action is retrieved
5. The input filter parameters, on the API request, are validated
6. A dataset that matches the input criteria is created
7. The requested dataset is reduced according to the user consents (consent enforcement) and anonymized (anonymization)
8. Au audit message is generated in the Public Cloud internal audit log for each user that his data is included in the returned dataset
9. The anonymized dataset is returned from the Public Cloud to the Data Consumer Portal

More detailed information about the Data-as-a-Service API and the interchange of information with the DAPHNE cloud could be found in *D6.2 Data-as-a-service API (ongoing)*. Roni Ram, DAPHNE [2]

3.3 Development and technologies

The data consumer portal is going to be developed between M19 and M30. Although the details about the development are going to be described in *D5.5 Big Data services first prototype* (M24) and *D5.7 Big Data services final development* (M30), some details can be decided at this point.

[1] https://console.ng.bluemix.net/

- **Installation**: Data Consumer Portal is going to be hosted as part of Treelogic installations following the technical requirements as they are defined in section 4.1 “”.
- **Development**: In order to facilitate the development and integration of the data consumer portal, three separate environments are going to be defined:
    - **Development environment**: This environment is going to be used internally in Treelogic facilities to develop and test the portal functionalities with no external connection.
    - **Test environment**: This environment will be used during the integration of the data consumer portal with the rest of DAPHNE components. It will include the integration with some of the functionalities of the DAPHNE data cloud (Data-as-a-service API) and the security components, as well as the required integration tests.
    - **Production environment**: This environment will host the final and stable version of the data consumer portal. All the functionalities will be available and integration with the rest of DAPHNE components will be completed.
- **Technologies**: Data consumer portal will be developed in Java Platform, Enterprise Edition (J2EE) and will consume Data-as-a-service API, which is based on REST services. Data is going to be formatted in JSON standard.
    - **J2EE** includes several API specifications that could be needed in the data consumer portal development, such as RMI, e-mail, JMS, web-services and XML. This allows creating logic in a portable and scalable way. Java EE applications can handle transactions, security, scalability, concurrency and management of the deployed components.
    - **REST:** According to the API specifications in D6.1 [12] and D6.2 [2], it was decided to use an approach of web services APIs that adhere to the REST architectural constraints. As stated in [9], RESTs client-server separation of concerns simplifies component implementation, reduces complexity of connector semantics, improves the effectiveness of performance tuning, and increases the scalability of pure server components.
    - **JSON:** JavaScript Object Notation is going to be the chosen format for data interchange. It is no tightly tailored to clinical data representation. JSON has all the advantages of XML messages but it is much better suited to data-interchange.

4 Security aspects

Although all the DAPHNE security aspects are described in detail in [10], some aspects related to the enrolment of new DC users in DAPHNE and the requesting information internal process, are described in the next sections.

4.1 Enrolment

The enrolment of new data consumers is an offline process in which the big data consumer sends an email request to the with their registration details to DAPHNE ethical committee. If this request is approved, it is forwarded to the IAM administrator with the new user's details and information.

According to the Council of Europe, Recommendation No. R (97) 18 the following information must be captured during the enrolment process of big data consumers:

- The purpose for requesting the access to the big data services.
- Name and position of the person or body in charge of receiving and processing the requested information from the big data services.
- Data consumer agreement according to terms and conditions to keep the big data information confidential and used for only the stated purpose.
- The legal basis for the request.

Once the data consumer request has been approved and the IAM administrator has registered the user to DAPHNE data consumer portal, this new data consumer will receive his login credentials so he can login and access to DAPHNE big data services.

The data consumer profile may be loaded into the data consumer portal every time a data consumer logins into the portal in order to remind the user to strictly follow the terms that he originally set regarding his purpose for data collection.

4.2 Authentication and Authorization

All the aspects related to authentication and authorization in the Data Consumer Portal, are defined in specified in section 5.1.1.1 and 5.1.2 in D6.5 [10] for the authentication and authorization processes.

4.3 Intellectual Property

The intellectual property right of the Big Data set should be considered and access to the data restricted by contract and technology where the following technologies should be considered:

- Digital Watermarks
- DRM
- Fingerprint Hash

More information about this aspects can be found in D6.5 [10].

5 Data consumer portal design

All the services defined in this deliverable will be available for DC through the Data Consumer Portal. This portal will be divided to three views according to the defined services:

- Datasets Download
- Visualization Tool
- Data Analysis Tool

In addition, in a similar way as the other DAPHNE portals (PHS end-user portals and Physician portal, both described in *D5.2 Personal Health Services (PHS) Design* [8]), the Data Consumer Portal will have a login page in order to identify and authorize portal users (DC).

5.1 Portal Login

DC must login in the Data Consumer Portal in order to get access to the DC services. These services are going to be available only for those DC authorized by the DAPHNE platform.

DC will access the web application by introducing the web portal URL in a web browser (Mozilla Firefox, Google Chrome, Internet Explorer...). The first screen the web application will show is a presentation-login static page, where the DC will have to specify his DAPHNE's credentials:

- User name
- Password

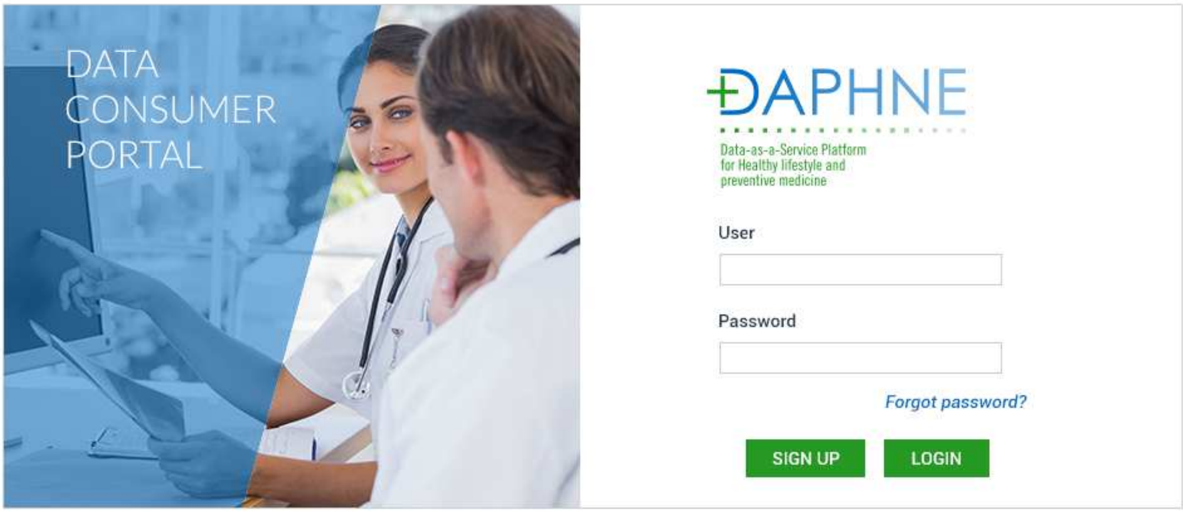

DATA CONSUMER PORTAL
DAPHNE
Data-as-a-Service Platform for Healthy lifestyle and preventive medicine
User
Password
Forgot password?
SIGN UP
LOGIN

Figure 7 DC Portal "Login" page

Once the DC has introduced his DAPHNE credentials, the system may allow him (or deny) to access to the DC web portal. Only DC users with permission may be allowed to enter the DC portal.

5.2 Datasets Download

In this view, DC users may be able to configure and download anonymized datasets containing DAPHNE users' parameters, concerning information as:

- Anthropometrics

- Nutrition
- Behaviour Habits
- Physical Activity
- Health Markers
- Psychological Wellbeing

DC offers web forms to configure the dataset to download based on specified parameters. These parameters define the selection criteria and include the following:

- **Age**: This parameter allows DC to specify the range of users' age that their data will be included in the retrieved dataset. DC can leave it as "All", in order to not filter by age.
- **Gender**: Information can be filtered by gender. DC can choose to download information only related to male/female DAPHNE users, or leave it by default "All", in order to get information from both genders.
- **Weight (Kg) and Height (cm)**: Information can be filtered by the weight and/or height of the DAPHNE users. This may be useful for DC in order to study how it effects on the retrieved parameters.
- **User Characteristics**: The system is also able to filter users' information by their health characteristics, such as:
  - Smoker
  - Pre-Diabetic
  - Diabetic

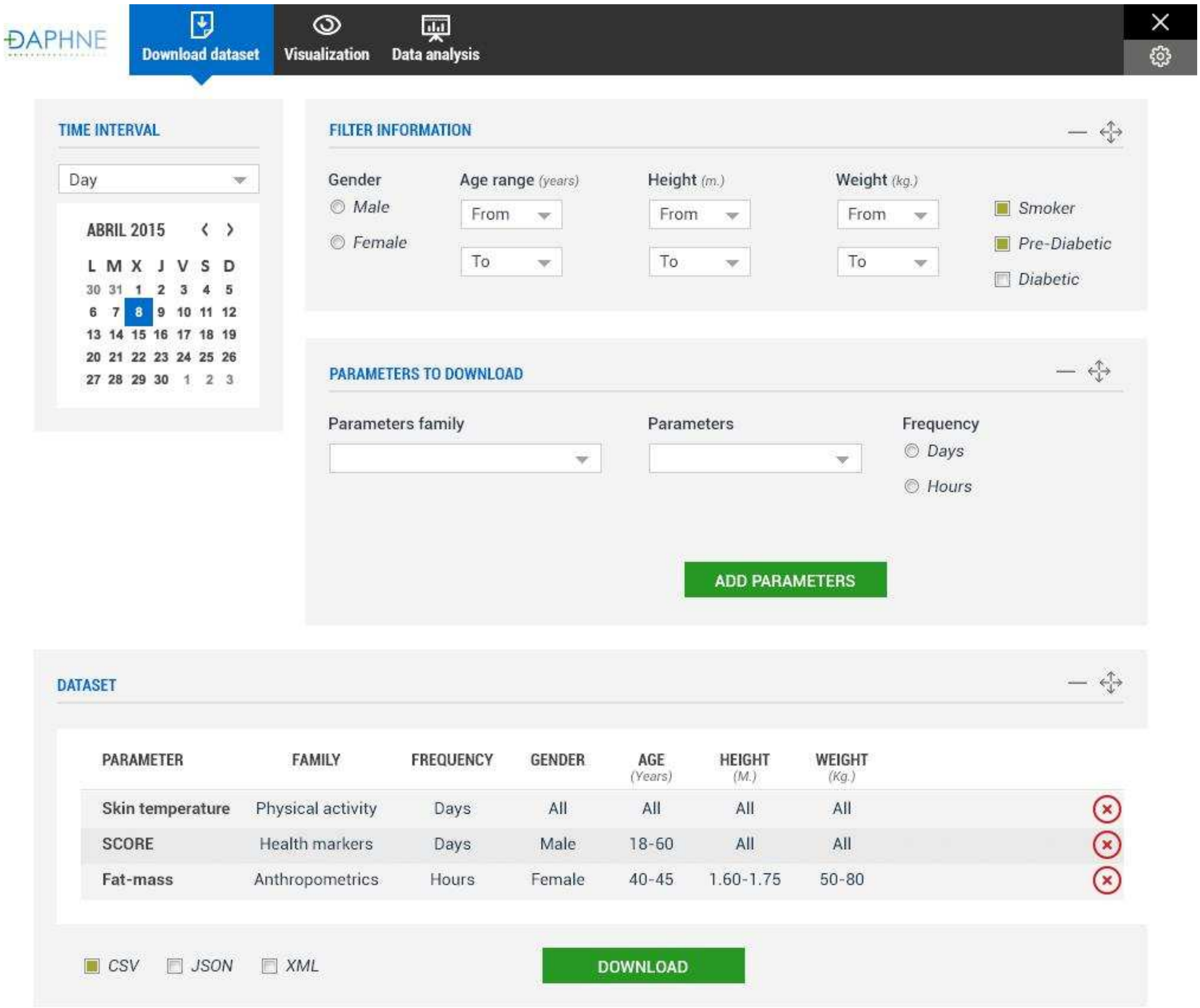
DC will be able to download the following data types:

Table 7 Classification of data types available to be downloaded

Group	Data item	Units	Description
Anthropometrics	Body mass index	Kg/m2	
	Fat-mass	Kg	
	Fat-free mass	Kg	
	Waist circumference	Cm	
	Hip circumference	Cm	
	WHtR	Cm/cm	
Health Markers	Diastolic BP	mmHg	
	Systolic BP	mmHg	
	Cardiovascular fitness	L/min	
	Fasting Glucose	mmol/L	
	HBAlc	mmol/mol	
	Triglycerides	mmol/L	
	HDL	Mmol	
	LDL	Mmol	
	Total Cholesterol	Mmol/L	
	SCORE	0-20	
Psychological Wellbeing	Child Behavior Check List	118 items results/14 scale results	
	Adult Self Report / Adult Behavior Check List	123 items results/24 scale results	
	Beck Depression Inventory	21 items results/1 scale results	
	State-Trait Anxiety Inventory	40 items results/2 scale results	
	Perceived Stress Scale	10 items results/1 scale results	

	Profile of Mood States	65 items results/8 scale results	
	WHO Quality of life	26 items results/4 scale results	
	Psychological Stress response	3 items results/1 scale results	
	Patient Health Questionnaire	9 items results/1 scale results	
	Generalized Anxiety Disorder 7	7 items results/1 scale results	
Physical Activity / Behaviour analyzer	% Classification Activity		Sedentary, Light, Moderate, Vigorous, Very Vigorous
	Heart rate	Bpm	Hours, days
	Heart rate Resting	Bpm	Day
	Skin temperature		
	Sweating	Low, Medium, High	Hours, day
	Pedometer	Steps	Day
	Distance	m	Day
	Energy Expenditure	Kcal	
	Total hours doing activity	h	Hours per sedentary, light, moderate, vigorous, very vigorous activity per day
	Type of activity		(running, walking, cycling)
	Average stress		Per day
	Time of stress peak	Time	
Food Intake	Energy intake	Kcal	Per day
	Macro Nutrients SCORE		Fat, Carbs, Proteins, Water, Fibre, Salt, Sugar

Considering this table with all the data types available for DC users, the following design is propose



Day

■ Smoker
■ Pre-Diabetic
□ Diabetic

PARAMETER	FAMILY	FREQUENCY	GENDER	AGE (Years)	HEIGHT (M.)	WEIGHT (Kg.)
Skin temperature	Physical activity	Days	All	All	All	All
SCORE	Health markers	Days	Male	18-60	All	All
Fat-mass	Anthropometrics	Hours	Female	40-45	1.60-1.75	50-80

■ CSV □ JSON □ XML

Figure 8 DC portal "Download Dataset" view

According to this figure, DC may use this view to download datasets following the following steps:

1) **Choose Time Interval**: DC is able to select the time interval of the parameters under study. In a similar way as the Personal Health Services (*D5.2 Personal Health Services (PHS) design* [8]), DC can choose between the following time intervals:
   i. A day (Selected in a calendar object)
   ii. Last 7 days
   iii. Last 14 days
   iv. Last Month
   v. Last 3 Months
   vi. Custom time interval
2) **Define filtering options**: After a time interval is specified, DC can define filter criteria ranges based on the users' anthropometrics information (age, height, weight or gender) and their health conditions, such as if they smoke, or they are diabetic (or pre-diabetic).
3) **Select Parameters to download**: Once that time interval and filtering options are defined, DC has to choose in the "Parameters to download" view those parameters that are going to form the final dataset. In particular, for every parameter to be downloaded, DC must define the following aspects:
   a. **Parameter Family**: Parameters are grouped by families in order to organize them properly. The existing families are:
      a. Food Intake
      b. Physical Activity/Behaviour Analyzer
      c. Health Markers

d. Psychological Wellbeing
e. Anthropometrics

b. **Parameter**: In this field DC chooses the parameter to include in his dataset. Parameters shown in this textbox depend on the family chosen in the “Parameter Family” field.

c. **Frequency**: In this field, DC can choose the resolution/frequency of the chosen parameter. Resolution may depends on the parameter, and it can be two possible resolutions:
   a. Hours
   b. Days

d. **Operations**: For each parameter, DC can choose between the following list of results/operations:
   a. MAX
   b. MIN
   c. AVG

4) **Download dataset**: Once all the previous steps have been completed, DC must choose one (or more) format for the dataset to be downloaded. DC can choose between the following formats:
   a. JSON
   b. XML
   c. CSV

   Once the format has been chosen, DC must press the “download” button in order to send the request to DAPHNE data cloud.

5) The dataset is retrieved from the Data Cloud as a JSON object which is then parsed. The required operations are calculated and the data is formatted according to the specified format
6) DC can download the anonymized dataset that was generated according to his specific requirements

5.2.1 Datasets format and fields

Downloaded datasets will follow a specific format in order to facilitate data integration with researchers systems. Each dataset will include the following fields:

Table 8 Required fields in dataset consumed by the Data Consumer

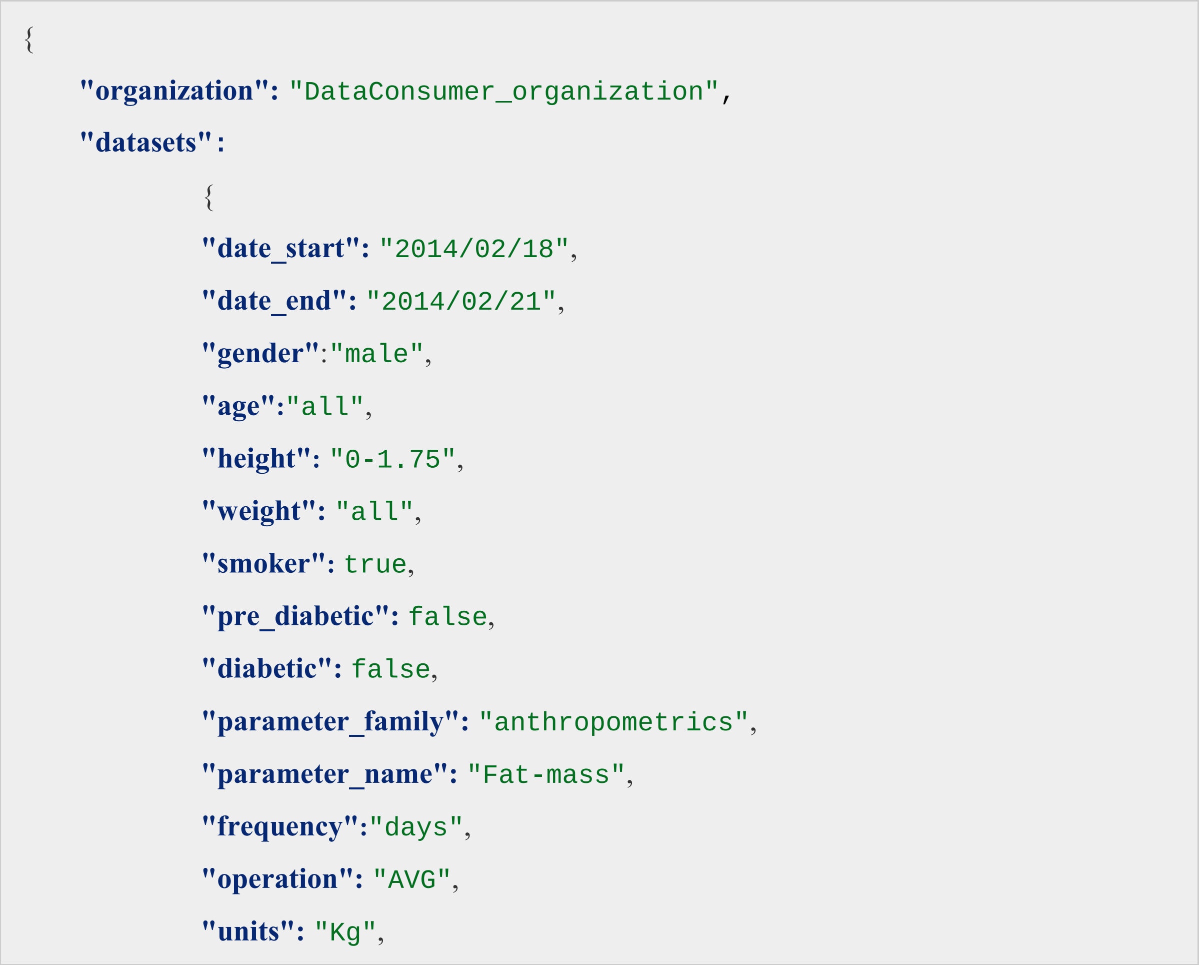
Name	Description	
“dataconsumer”	This field includes the name of the DC responsible of the data downloaded	
“datasets”	This field contains an array of objects. Each object will contain all the data related to a downloaded parameter. This way a dataset file can contain as many “datasets” as objects in this field. Each of these “datasets” objects has the following structure.	
	“date_start”	It represents the starting date of the time interval considered in the dataset.
	“date_end”	It represents the ending date of the time interval considered in the dataset.
	“gender”	It represents the gender of the sample population. Its value can be “male”, “female” or “all”
	“age”	It represents the age interval of the sample population under study.
	“height”	It represents the height interval of the sample population under study.
	“weight”	It represents the weight interval of the sample population under study.
	“smoker”	It represents if the population sample contains smoker users.
	“pre-diabetic”	It represents if the population sample contains users with diabetic forefathers.
	“diabetic”	It represents if the population sample under study is diabetic

		users.	
	"parameter_group"	It represents the name of the group of the parameter under study	
	"parameter_name"	It is the name of the parameter under study.	
	"frequency"	It represents the frequency of the sample values under study. It could be "hours" or "days".	
	"operation"	It represents the operation of the sample values under study. It could be "AVG", "MAX", "MIN"	
	"units"	It represents the unit of the sample values under study.	
	"data"	It is an array containing multiple pairs of timestamp-value for the data requested by the data consumer.	
		"timestamp"	Timestamp of the pair value.
		"value"	Value of the parameter in the given timestamp

If this schema is applied to a dataset in JSON format, containing the average fat-mass per day of DAPHNE's population with the following filters:

- Date: From 2014/02/18 to 2014/02/21
- Gender: Male users
- Age: No limitations
- Height: Less than 1.75m
- Weight: No limitations
- Smokers: Only considering smokers

The resulting dataset would be very similar to the dataset shown in the next figure:



```
{
    "organization": "DataConsumer_organization",
    "datasets":
            {
            "date_start": "2014/02/18",
            "date_end": "2014/02/21",
            "gender":"male",
            "age":"all",
            "height": "0-1.75",
            "weight": "all",
            "smoker": true,
            "pre_diabetic": false,
            "diabetic": false,
            "parameter_family": "anthropometrics",
            "parameter_name": "Fat-mass",
            "frequency":"days",
            "operation": "AVG",
            "units": "Kg",
            "data": [
                    {"timestamp": "20140218", "value": "1.2"},
                    {"timestamp": "20140218", "value": "1.2"},
                    {"timestamp": "20140219", "value": "1.1"},
                    {"timestamp": "20140220", "value": "1.6"},
                    {"timestamp": "20140221", "value": "0.9"}
            ]
    }
}
```

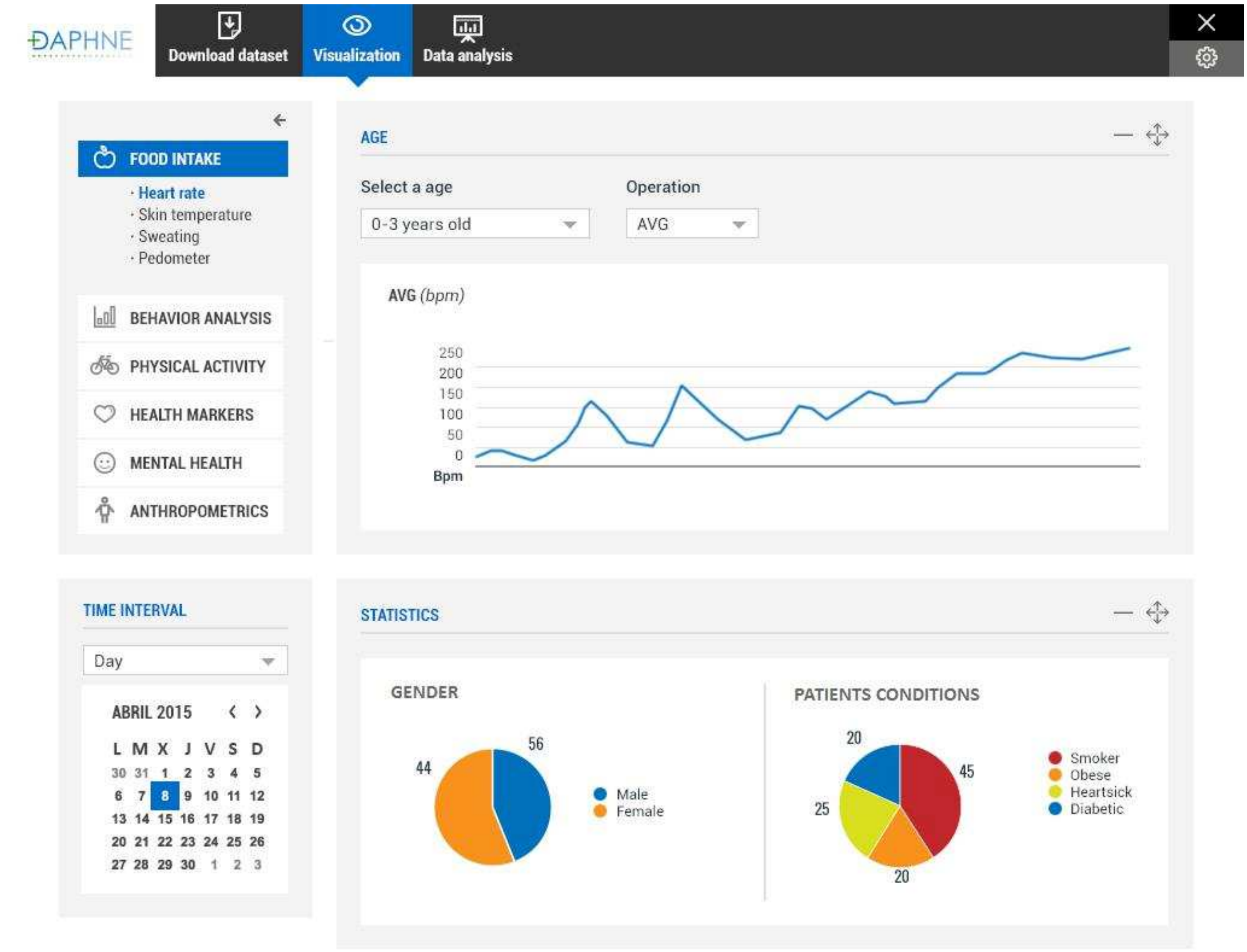
Figure 9 JSON dataset example

5.3 Parameters Visualization

In this view, the anonymized data about the DAPHNE users is displayed in multiple graphs, grouped by:

- Food Intake
- Behaviour Analysis
- Physical Activity
- Health Markers
- Psychological Wellbeing
- Anthropometrics

From the downloaded datasets, DC users can display several graphs related to DAPHNE user’s parameters, as it is shown in the following figure:



Select a age: 0-3 years old

Operation: AVG

TIME INTERVAL: Day

Figure 10 DC portal "Visualization" view

In particular, this view allows DC to check the parameters showed in the **Table 7 Classification of data types available to be downloaded** in a graphic way. For each of these parameters, the data consumer portal is going to show graphs about the parameter evolution in time, and the DC can choose between the following visualization options:

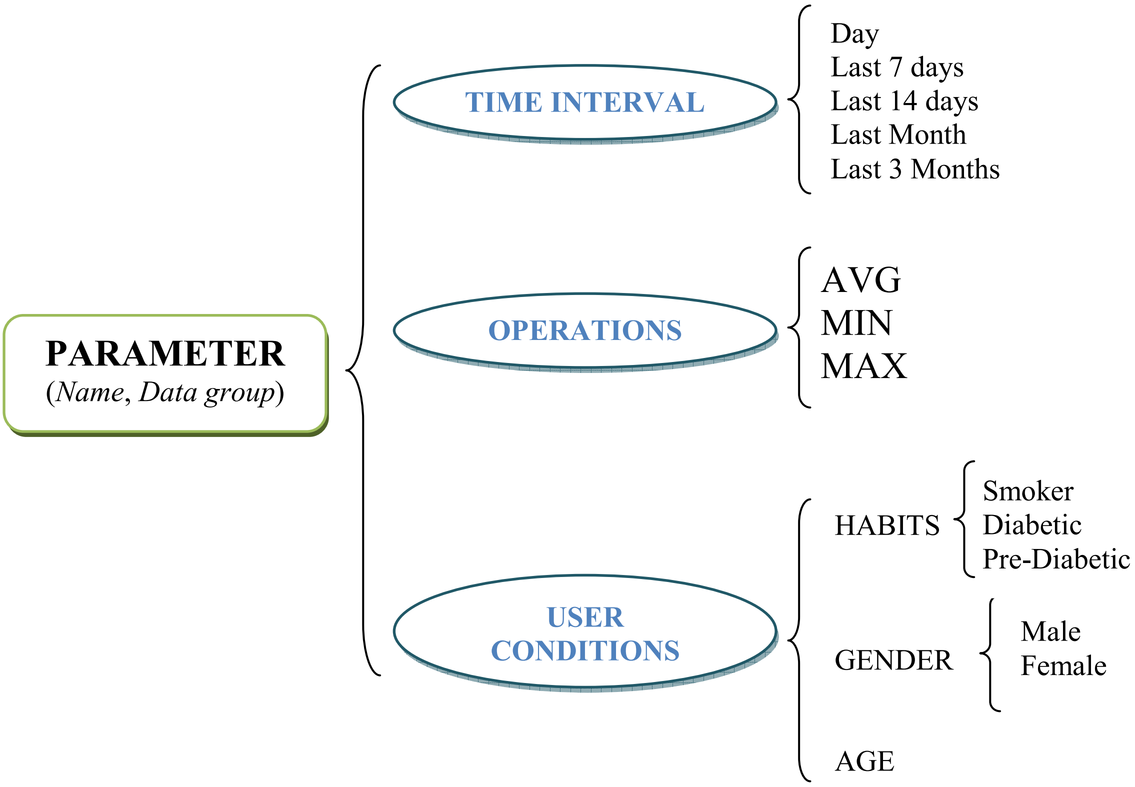


Figure 11 Visualization options for each parameter in "Parameters Visualization"

According to this, given a parameter, DC must choose:

Time Interval: It defines the time period to consider in the graphs. This option is similar to the "Time Interval" used in the Dataset Download view, and DC can choose between:

i. A day (Selected in a calendar object)
ii. Last 7 days
iii. Last 14 days
iv. Last Month
v. Last 3 Months

Operation: The displayed graphs are the results of applying an operation on the parameter under study. This means, that for each parameter, DC can visualize the evolution in time of the parameters:

a. MAX
b. MIN
c. AVG

User Conditions: Displayed graphs can be personalized depending on the user's conditions. According to this, parameters can be visualized related to:

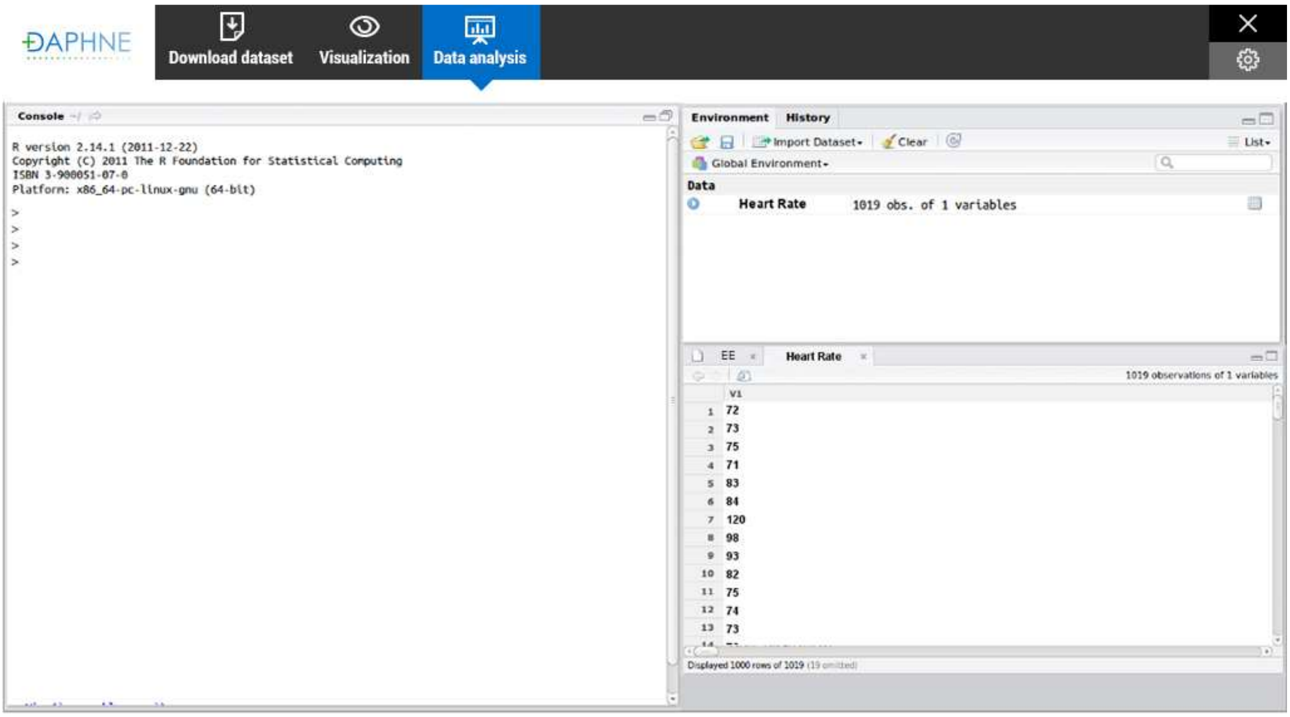
- User's habits:
  - Smoker
  - Diabetic
  - Pre-Diabetic
- Gender:
  - Male
  - Female

- Age

DC can switch between the different visualization options of a parameter by clicking on the multiple icons displayed in the visualization area. This will be made by filtering data according to the different User Conditions from the resulting dataset obtained from the Cloud. Technically, the Data Consumer Portal is going to retrieve the data with no consideration on the User Conditions, and then, once the dataset has been obtained from the Cloud, the portal will select which records must be displayed based on the User Conditions selected by the DC.

5.4 Data Analysis

In this view, DC can use R statistical environment for computing DAPHNE datasets, combing or analyzing them. According to the functional requirements and the use cases described in previous sections of this document, the "Data Analysis" view is similar to the following figure:



```
R version 2.14.1 (2011-12-22)
Copyright (C) 2011 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: x86_64-pc-linux-gnu (64-bit)

>
>
>
>
```

Data

Heart Rate 1019 obs. of 1 variables

1019 observations of 1 variables

	V1
1	72
2	73
3	75
4	71
5	83
6	84
7	120
8	98
9	93
10	82
11	75
12	74
13	73

Displayed 1000 rows of 1019 (19 omitted)

Figure 12 DC portal, "Data Analysis" view

This view is divided to the following "windows" in order to offer to DC a complete environment based on R.

a) **Console**: This window is the most important component of the "Data Analysis" view. This is where DC types R commands and sees the text results. DC are prompted to type commands with the greater than symbol. The console allows command editing. This means that the left and right arrow keys, home, end, backspace, insert and delete work exactly as they do in a common Unix console. It also manages the command history; the up and down keys can be used to scroll through recent commands.

b) **History**: This window shows a list of recent commands executed by the DC in the console section. If the DC clicks on them, the command is re-executed. This facilitates command reutilization.

c) **Environment**: This window allows DC to import datasets that they have already downloaded from DAPHNE (or download new ones), so they can work with them in the R console. In this frame, DC can check a list with all the current variables (and datasets) that are being used in the "console" section. DC can click on any of the items on this list, and its content will appear in the "**edit**" window.

d) **Edit**: The main function of this window is to show the content of the variables and datasets used/defined by the DC. In this area, variables and datasets can be edited row by row.

In this view, a DC usually interacts with the data consumer portal in this way:

1) DAPHNE parameters and datasets that will be considered are defined in the "Environment" window. Datasets can be imported from a local file, or can be created from DAPHNE directly as explained in section 5.2 by using the Data-as-a-service API. For this purpose, there will be an element in the Data Consumer Portal, able to manage DC petitions and convert them in DAPHNE Data-as-a-Service API calls. This element is transparent for the general DAPHNE architecture; it will be integrated in the Data Consumer Portal. Technically, it will "translate" Cloud datasets into R format (Rd format).
2) Once the datasets have been defined, DC is able to operate them by using R commands. These commands can be inserted by DC one by one in the "Command" window, or they can be executed through an R script previously defined by the user and uploaded from his local storage system.
3) The view shows the results of the operations in multiple ways. It depends on the type of the result obtained.
    a. **Numbers or arrays**: They are displayed in the command section after the operation is executed.
    b. **Plots:** R is able to display graphics in emerging windows. This will be necessary when DC defines operations that their results are plots and charts. These plots will be downloadable and exported to image file formats.
    c. **Datasets or defined parameters**: DC can define variables in order to include them in the "Environment" window, in order to edit/visualize and/or reuse them in other operations. By default, R systems save all variables in the "Environment" window.

5.4.1 Data analysis detailed example scenario

This section describes a particular example scenario of use case of the "data analysis" service. In this example, a DC wants to get a graphic representing the evolution of the daily average Heart Beat in the last three months of all the smoker males between 20 and 40 years old using Daphne, and in the same graphic, represent also this evolution but considering also female users in the same age range (and smokers too).

1) **Create source datasets**: First of all, DC has to generate the source datasets containing the required information. In this example, there are two datasets, described below:
    a. **"hrDatasetMALE" dataset**: DC must choose in the "Import Dataset" to download a DAPHNE users' dataset fulfilling the following requirements:
        i. **Time interval**: Last 90 days (Last 3 months)
        ii. Filter information
            1. Gender: Male
            2. Age interval: 20-40 years old
            3. No weigh or height limitations
            4. Only data from smoker users
        iii. Parameters
            1. Daily Heart Rate
    b. **"hrDatasetFEMALE" dataset**: Similar to the "Male" dataset, but considering only woman data
        i. **Time interval**: Last 90 days (Last 3 months)
        ii. Filter information
            1. Gender: Male
            2. Age interval: 20-40 years old
            3. No weigh or height limitations
            4. Only data from smoker users
        iii. Parameters
            1. Daily Heart Rate

Once downloaded, both dataset will appear in the “Data” window, and DC could open, edit or delete them in this section. They will be ready to be operated in the command area.

2) **Apply operations**: Once the datasets have been built, DC must apply the mean on each dataset. This will calculate the mean Heart Rate per day of all the DAPHNE users male, 20-40 years old and smokers of the last 90 days (and the mean Heart Rate per day of all the DAPHNE female users, 20-40 years old, smokers). The commands used in this step are the following ones:

```
> hrAvgMALE = mean(hrDatasetMALE);
> hrAvgFEMALE = mean(hrDatasetFEMALE);
```

3) **Plot results**: DC will plot the resulting means in the same graphic in order to compare both evolutions of Heart Rate in the last 90 days.

```
> plot(hrAvgMALE, type=”l”, col=”blue”);
> lines(hrAvgFEMALE, type=”l”, col=”red”);
```

The resulting graph may be very similar to the following figure, where DC can conclude that the women’s Heart Rate of smoker DAPHNE users between 20-40 years old, is lower than men’s Heart Rate (for example).

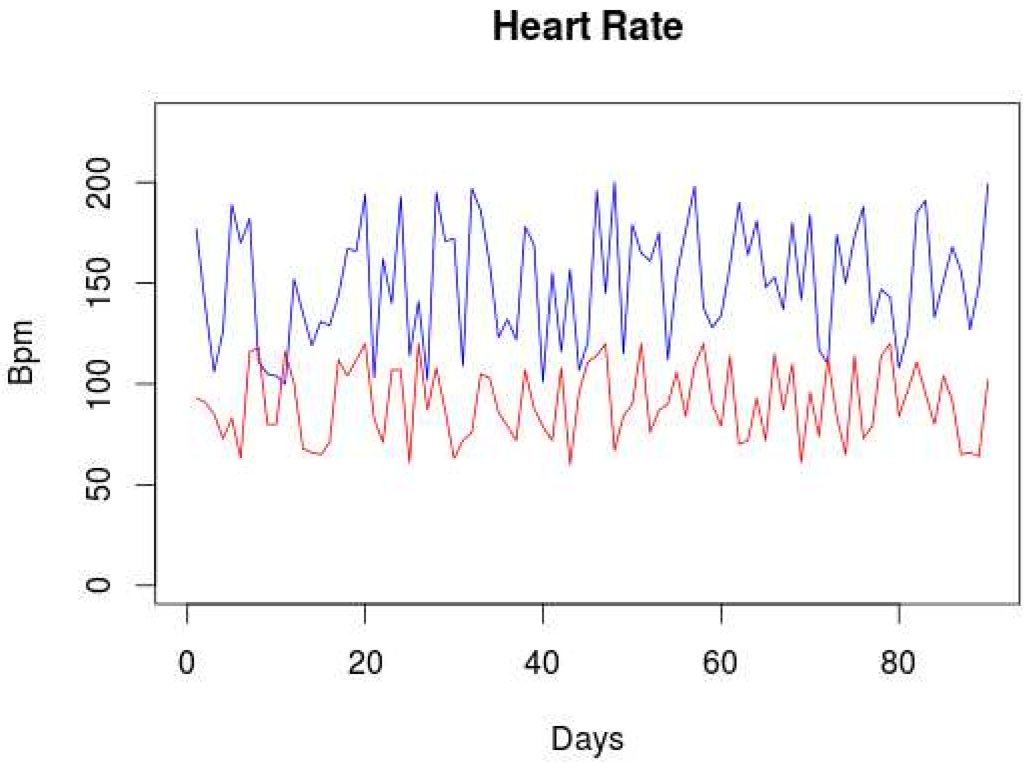


Figure 13 Resulting graph in R example use case

Conclusions

This document defines the three big data services offered by DAPHNE. These services have been specified taking into account Data Consumer's needs, use cases and their functional requirements. They are going to consume DAPHNE data from DAPHNE's data cloud through the Data-as-a-Service API specified in D6.2. In the services definition sections, detailed screens of the Data Consumer portal have been included to facilitate its implementation. For each one of the big data services, functional requirements and use cases have been identified, and concluded with the Data Consumer final design.

Dataset Download: This service allows DC to download datasets containing anonymized information about DAPHNE's wellbeing users and patients. This information is useful for researchers so they can take it into account in their studies and models. DC can define the format and content of the dataset.

Parameters Visualization: This service is useful for those DC who want to study the DAPHNE users' health parameters, intake/ activity/mental behaviours and their relation with age, gender, habits and other user conditions in a visual way.

Data Analysis: This service may provide an online tool to operate, analyse and represent DAPHNE information with a software environment for statistical computing and graphics, based on R. It is designed for those researchers with statistic processing needs over DAPHNE data. This service will be based on R language so these researchers must have a strong experience with this mathematical language.

The technical requirements section specifies the technical aspects related to the development of the data consumer portal. These aspects focus on defining the technical requirements of the portal (related to hardware, software and communications), the technical interactions with other DAPHNE components and the technologies and the development methodology and technologies that are going to be considered in the development phase.

In addition, the security aspects that must be considered in the Data Consumer portal are also defined in this document. The DC enrolment process, authentication and authorization aspects and the intellectual property of the downloaded datasets are mentioned in this deliverable, although are more detailed described in D6.5.

The development of the Data Consumer portal will be done in the scope of *T5.5 Big Data Services Development* (M18-M30). The results of this task will be included and detailed in the following deliverables:

D5.5 Big Data services first prototype (M24) – It will consist on a first prototype version of the Data Consumer Portal. Data-as-a-Service API will be tested in this prototype and a first version of the big data services will be implemented.

D5.7 Big Data services final development (M30) – This deliverable will consist on the Data Consumer portal final version with all the big data services implemented and fully operational. Data-as-a-service API will be completely integrated

References

[1] *D2.7 DAPHNE System Architecture final design and development (ongoing)*. Carlos Marcos. DAPHNE

[2] *D6.2 Data-as-a-service API (ongoing)*. Roni Ram, DAPHNE

[3] *R programming language* http://www.r-project.org/

[4] *D.2.4 Set Requirements of the future DAPHNE Platform.* Carlos Marcos. DAPHNE

[5] *RStudioServer http://www.rstudio.com/*

[6] *Dreamgenics http://www.dreamgenics.com/en*

[7] Lambdoop *http://lambdoop.com/*

[8] *D5.2 Personal Health Services (PHS) Design.* Jose Antonio Sánchez. DAPHNE

[9] *Chapter 5: Representational State Transfer (REST). Architectural Styles and the Design of Network-based Software Architectures (Ph.D.) (2000). Fielding, Roy T. University of California, Irvine*

[10] *D6.5 Privacy and Security design guidelines for DAPHNE (ongoing). Ángel Palomares Pérez, DAPHNE*

[11] *About OpenAM Web Policy Agents [Online], http://docs.forgerock.org/en/openam-pa/3.1.0-Xpress/agent-install-guide/index/chap-about-web-agents.html*

[12] *D6.1Data provider API.* Roni Ram. DAPHNE

[13] *Matlab: The Language of technical computing http://uk.mathworks.com/products/matlab/index.html*

[14] *Matpltlib http://matplotlib.org/*

[15] *Microsoft Excel http://products.office.com/en-GB/excel?omkt=en-GB*

[16] *SAS* http://www.sas.com/en_us/software/analytics/stat.html

[17] *Stata: Data analysis and Statistical software* http://www.stata.com/

[18] *CRAN project (The Comprehensive R Archive Network) http://cran.r-project.org/*

[19] *CRAN r-project packages http://cran.r-project.org/web/packages/available_packages_by_date.html*

Annex I : The R environment

Introduction

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. Among other things it has

- an effective data handling and storage facility,
- a suite of operators for calculations on arrays, in particular matrices,
- a large, coherent, integrated collection of intermediate tools for data analysis,
- graphical facilities for data analysis and display either directly at the computer or on hardcopy, and
- a well developed, simple and effective programming language (called 'S') which includes conditionals, loops, user defined recursive functions and input and output facilities. (Indeed most of the system supplied functions are themselves written in the S language.)

The term "environment" is intended to characterize it as a fully planned and coherent system, rather than an incremental accretion of very specific and inflexible tools, as is frequently the case with other data analysis software.

R is very much a vehicle for newly developing methods of interactive data analysis. It has developed rapidly, and has been extended by a large collection of *packages*. However, most programs written in R are essentially ephemeral, written for a single piece of data analysis.

Environment

R can be regarded as an implementation of the S language which was developed at Bell Laboratories by Rick Becker, John Chambers and Allan Wilks, and also forms the basis of the S-PLUS systems.

The evolution of the S language is characterized by four books by John Chambers and coauthors. For R, the basic reference is *The New S Language: A Programming Environment for Data Analysis and Graphics* by Richard A. Becker, John M. Chambers and Allan R. Wilks. The new features of the 1991 release of S are covered in *Statistical Models in S* edited by John M. Chambers and Trevor J. Hastie. The formal methods and classes of the **methods** package are based on those described in *Programming with Data* by John M. Chambers.

There are now a number of books which describe how to use R for data analysis and statistics, and documentation for S/S-PLUS can typically be used with R, keeping the differences between the S implementations in mind.

R and statistics

Our introduction to the R environment did not mention *statistics*, yet many people use R as a statistics system. We prefer to think of it of an environment within which many classical and modern statistical techniques have been implemented. A few of these are built into the base R environment, but many are supplied as *packages*. There are about 25 packages supplied with R (called "standard" and "recommended" packages) and many more are available through the CRAN family of Internet sites[i] and elsewhere. More details on packages are given later.

Most classical statistics and much of the latest methodology are available for use with R, but users may need to be prepared to do a little work to find it.

There is an important difference in philosophy between S (and hence R) and the other main statistical systems. In S a statistical analysis is normally done as a series of steps, with intermediate results being stored in objects. Thus whereas SAS and SPSS will give copious output from a regression or discriminate analysis, R will give minimal output and store the results in a fit object for subsequent interrogation by further R functions.

Commands

Technically R is an *expression language* with a very simple syntax. It is *case sensitive* as are most UNIX based packages, so A and a are different symbols and would refer to different variables. The set of symbols which can be used in R names depends on the operating system and country within which R is being run (technically on the *locale* in use). Normally all alphanumeric symbols are allowed (and in some countries this includes accented letters) plus ‘.’ and ‘_’, with the restriction that a name must start with ‘.’ or a letter, and if it starts with ‘.’ the second character must not be a digit. Names are effectively unlimited in length.

Elementary commands consist of either *expressions* or *assignments*. If an expression is given as a command, it is evaluated, printed (unless specifically made invisible), and the value is lost. An assignment also evaluates an expression and passes the value to a variable but the result is not automatically printed.

Commands are separated either by a semi-colon (‘;’), or by a newline. Elementary commands can be grouped together into one compound expression by braces (‘{’ and ‘}’). *Comments*can be put almost anywhere, starting with a hashmark (‘#’), everything to the end of the line is a comment.

If a command is not complete at the end of a line, R will give a different prompt, by default

+

on second and subsequent lines and continue to read input until the command is syntactically complete. This prompt may be changed by the user. We will generally omit the continuation prompt and indicate continuation by simple indenting.

Command lines entered at the console are limited to about 4095 bytes (not characters).

R is an interpreted language; users typically access it through a command-line interpreter. If a user types 2+2 at the R command prompt and presses enter, the computer replies with 4, as shown below:

```
> 2+2

> 4
```

Like other similar languages such as APL and MATLAB, R supports matrix arithmetic. R's data structures include vectors, matrices, arrays, data frames (similar totables in a relational database) and lists. R's extensible object system includes objects for (among others): regression models, time-series and geo-spatial coordinates. The scalar data type was never a data structure of R. A scalar is represented as a vector with length one in R.

R supports procedural programming with functions and, for some functions, object-oriented programming with generic functions. A generic function acts differently depending on the type of arguments passed to it. In other words, the generic function dispatches the function (method) specific to that type of object. For example, R has a generic print function that can print almost every type of object in R with a simple print(objectname) syntax.

Although used mainly by statisticians and other practitioners requiring an environment for statistical computation and software development, R can also operate as a general matrix calculation toolbox – with performance benchmarks comparable to GNU Octave or MATLAB. Arrays are stored in column-major order.

The following R commands are going to be considered to test the “Data Analysis” service described in this document.

Input and display

```
#read files with labels in first row
read.table(filename,header=TRUE)          #read a tab or space delimited file
read.table(filename,header=TRUE,sep=',')  #read csv files
```

```
x <- c(1,2,4,8,16 )                     #create a data vector with specified elements
y <- c(1:10)                            #create a data vector with elements 1-10
```

```
n <- 10
x1 <- c(rnorm(n))                       #create a n item vector of random normal deviates
y1 <- c(runif(n))+n                     #create another n item vector that has n added to each random uniform distribution
z <- rbinom(n,size,prob)                   #create n samples of size "size" with probability prob from the binomial
vect <- c(x,y)                          #combine them into one vector of length 2n
mat <- cbind(x,y)                       #combine them into a n x 2 matrix
mat[4,2]                             #display the 4th row and the 2nd column
mat[3,]                              #display the 3rd row
mat[,2]                              #display the 2nd column
subset(dataset,logical)                 #those objects meeting a logical criterion
subset(data.df,select=variables,logical)  #get those objects from a data frame that meet a criterion
data.df[data.df=logical]                #yet another way to get a subset
x[order(x$B),]                          #sort a dataframe by the order of the elements in B
x[rev(order(x$B)),]                     #sort the dataframe in reverse order

browse.workspace                        #a Mac menu command that creates a window with information about all variables in the workspace
```

Moving around

```
ls()                                 #list the variables in the workspace
rm(x)                                #remove x from the workspace
rm(list=ls())                          #remove all the variables from the workspace
attach(mat)                            #make the names of the variables in the matrix or data frame available in the workspace
detach(mat)                            #releases the names (remember to do this each time you attach something)
with(mat, .... )                       #a preferred alternative to attach ... detach
new <- old[,-n]                           #drop the nth column
new <- old[-n,]                           #drop the nth row
new <- old[,-c(i,j)]                    #drop the ith and jth column
new <- subset(old,logical)                  #select those cases that meet the logical condition
complete  <-  subset(data.df,complete.cases(data.df)) #find those cases with no missing values
new <- old[n1:n2,n3:n4]
```

Distributions

```
beta(a, b)
gamma(x)
choose(n, k)
factorial(x)

dnorm(x, mean=0, sd=1, log = FALSE)      #normal distribution
pnorm(q, mean=0, sd=1, lower.tail = TRUE, log.p = FALSE)
qnorm(p, mean=0, sd=1, lower.tail = TRUE, log.p = FALSE)
rnorm(n, mean=0, sd=1)


dunif(x, min=0, max=1, log = FALSE)      #uniform distribution
punif(q, min=0, max=1, lower.tail = TRUE, log.p = FALSE)
qunif(p, min=0, max=1, lower.tail = TRUE, log.p = FALSE)
runif(n, min=0, max=1)
```

Data manipulation

```
replace(x, list, values)                #remember to assign this to some object i.e., x <- replace(x,x==-9,NA)
                                #similar to the operation x[x==-9] <- NA
scrub(x, where, min, max, isvalue,newvalue)  #a convenient way to change particular values (in psych package)

cut(x, breaks, labels = NULL,
   include.lowest = FALSE, right = TRUE, dig.lab = 3, ...)

x.df <- data.frame(x1,x2,x3 ...)            #combine different kinds of data into a data frame
                   as.data.frame()
                    is.data.frame()
x <- as.matrix()
scale()                              #converts a data frame to standardized scores
```

```
round(x,n)                      #rounds the values of x to n decimal places

ceiling(x)                      #vector x of smallest integers > x
floor(x)                        #vector x of largest interger < x
as.integer(x)                   #truncates real x to integers (compare to round(x,0)
as.integer(x < cutpoint)             #vector x of 0 if less than cutpoint, 1 if greater than cutpoint)
factor(ifelse(a < cutpoint, "Neg", "Pos"))  #is another way to dichotomize and to make a factor for analysis
transform(data.df,variable names = some operation) #can be part of a set up for a data set

x%in%y              #tests each element of x for membership in y
y%in%x              #tests each element of y for membership in x
all(x%in%y)          #true if x is a proper subset of y
all(x)             # for a vector of logical values, are they all true?
any(x)              #for a vector of logical values, is at least one true?
```

Statistics and transformations

```
max(x, na.rm=TRUE)                  #Find the maximum value in the vector x, exclude missing values
min(x, na.rm=TRUE)
mean(x, na.rm=TRUE)
median(x, na.rm=TRUE)
sum(x, na.rm=TRUE)
var(x, na.rm=TRUE)                  #produces the variance covariance matrix
sd(x, na.rm=TRUE)                   #standard deviation
mad(x, na.rm=TRUE)                  #(median absolute deviation)
fivenum(x, na.rm=TRUE)              #Tukey fivenumbers min, lowerhinge, median, upper hinge, max
table(x)                            #frequency counts of entries, ideally the entries are factors(although it works with integers or even reals)
scale(data,scale=FALSE)                         #centers around the mean but does not scale by the sd)
cumsum(x,na=rm=TRUE)                            #cumulative sum, etc.
cumprod(x)
cummax(x)
cummin(x)
rev(x)                                          #reverse the order of values in x

cor(x,y,use="pair")                             #correlation matrix for pairwise complete data, use="complete" for complete cases

aov(x~y,data=datafile)                          #where x and y can be matrices
aov.ex1 = aov(DV~IV,data=data.ex1)              #do the analysis of variance or
aov.ex2 = aov(DV~IV1*IV21,data=data.ex2)        #do a two way analysis of variance
summary(aov.ex1)                                 #show the summary table
print(model.tables(aov.ex1,"means"),digits=3)    #report the means and the number of subjects/cell
boxplot(DV~IV,data=data.ex1)                    #graphical summary appears in graphics window

lm(x~y,data=dataset)                            #basic linear model where x and y can be matrices  (see plot.lm for plotting options)
t.test(x,g)
pairwise.t.test(x,g)
power.anova.test(groups = NULL, n = NULL, between.var = NULL,
          within.var = NULL, sig.level = 0.05, power = NULL)
power.t.test(n = NULL, delta = NULL, sd = 1, sig.level = 0.05,
          power = NULL, type = c("two.sample", "one.sample", "paired"),
          alternative = c("two.sided", "one.sided"),strict = FALSE)
```

Regression, the linear model, factor analysis and principal components analysis

```
matrices
t(X)                                            #transpose of X
X %*% Y                                         #matrix multiply X by Y
solve(A)                                        #inverse of A
solve(A,B)                                      #inverse of A * B    (may be used for linear regression)

data frames are needed for regression
lm(Y~X1+X2)
lm(Y~X|W)

factanal()    (see also fa in the psych package)
princomp()     (see principal in the psych package)
```

Additional commands

```
colSums (x, na.rm = FALSE, dims = 1)
rowSums (x, na.rm = FALSE, dims = 1)
colMeans(x, na.rm = FALSE, dims = 1)
rowMeans(x, na.rm = FALSE, dims = 1)
rowsum(x, group, reorder = TRUE, ...)       #finds row sums for each level of a grouping variable
apply(X, MARGIN, FUN, ...)                  #applies the function (FUN) to either rows (1) or columns (2) on object X
apply(x,1,min)                              #finds the minimum for each row
apply(x,2,max)                              #finds the maximum for each column
col.max(x)                                  #another way to find which column has the maximum value for each row
which.min(x)
which.max(x)
z=apply(x,1,which.min)                      #tells the row with the minimum value for every column
```

Graphics

```
par(mfrow=c(nrow,mcol))                                #number of rows and columns to graph
par(ask=TRUE)                                          #ask for user input before drawing a new graph
par(omi=c(0,0,1,0) )                                   #set the size of the outer margins
mtext("some global title",3,outer=TRUE,line=1,cex=1.5) #note that we seem to need to add the global title last cex = character expansion factor

boxplot(x,main="title")                                        #boxplot (box and whiskers)

title( "some title")                                           #add a title to the first graph

hist()                                                          #histogram
plot()
    plot(x,y,xlim=range(-1,1),ylim=range(-1,1),main=title)
    par(mfrow=c(1,1))                                           #change the graph window back to one figure
    symb=c(19,25,3,23)
    colors=c("black","red","green","blue")
    charact=c("S","T","N","H")
    plot(PA,NAF,pch=symb[group],col=colors[group],bg=colors[condit],cex=1.5,main="Postive vs. Negative Affect by Film condition")
    points(mPA,mNA,pch=symb[condit],cex=4.5,col=colors[condit],bg=colors[condit])

curve()
abline(a,b)
    abline(a, b, untf = FALSE, ...)
    abline(h=, untf = FALSE, ...)
    abline(v=, untf = FALSE, ...)
    abline(coef=, untf = FALSE, ...)
    abline(reg=, untf = FALSE, ...)

identify()
    plot(eatar,eanta,xlim=range(-1,1),ylim=range(-1,1),main=title)
    identify(eatar,eanta,labels=labels(energysR[,1])  )                  #dynamically puts names on the plots
locate()

legend()
pairs()                   #SPLOM (scatter plot Matrix)
pairs.panels ()          #SPLOM on lower off diagonal, histograms on diagonal, correlations on diagonal not standard R, but in the psych
package
matplot ()
biplot ())
plot(table(x))                       #plot the frequencies of levels in x

x= recordPlot()                      #save the current plot device output in the object x
replayPlot(x)                         #replot object x
dev.control                           #various control functions for printing/saving graphic files
pdf(height=6, width=6)               #create a pdf file for output
dev.of()                             #close the pdf file created with pdf
layout(mat)                          #specify where multiple graphs go on the page experiment with the magic code from Paul Murrell to do
fancy graphic location
layout(rbind(c(1, 1, 2, 2, 3, 3),
          c(0, 4, 4, 5, 5, 0)))
for (i in 1:5) {
  plot(i, type="n")
  text(1, i, paste("Plot", i), cex=4)
}
```

Distributions and random samples

```
rnorm(n,mean,sd)
rbinom(n,size,p)
sample(x, size, replace = FALSE, prob = NULL)          #samples with or without replacement
```

Dates

```
date <-strptime(as.character(date), "%m/%d/%y")        #change the date field to a internal form for time  see ?formats and ?POSIXlt
 as.Date
 month= months(date)                                    #see also weekdays, Julian
```

[i] http://CRAN.R-project.org