PPS data management plan Version 1.1

Instructions for setting up a data management structure for a new project

[Introduction 2](#_Toc43224656)

[Directory structure 2](#_Toc43224657)

[Data formatting requirements 3](#_Toc43224658)

[Data privacy issues 3](#_Toc43224659)

[Meta data 4](#_Toc43224660)

[Set-up script 4](#_Toc43224661)

## Introduction

A good data management structure is an essential part of research. It helps to ensure that data, analyses and results are stored and documented in a systematic way. It contributes to making research more transparent and accessible and helps ensuring that ethical and quality requirements related to data collection are met.

At PPS, we have developed a set of guidelines and structures that aim to ensure data is handled and stored uniformly. It is based around the following set of principles:

1. Integrity of raw data
2. Full documentation of all datasets (meta-data) and variables within a dataset
3. Reproducible pathway from (raw) data to results by commented scripts

In principle, each project such as a thesis chapter or research paper should be associated with a self-contained directory structure containing all data, scripts, results and written output that can be archived and made available for storage or sharing.

Below is a detailed description of what this directory structure should look like , how data should be formatted and documented and how to set up scripts to handle and analyse data transparently. This distribution comes with a helper R script that can be sourced to make an initial project folder structure, and that includes some example data and scripts that demonstrate the entire path from setting up a project directory, to processing raw data to a distributable file with a meta data page to simple analysis that produces and stores results. **A brief description of the implementation of this script can be found at the end of this document**.

## Directory structure

The hierarchical directory structure associated with a project is presented in Figure 1. A central project directory with an informative name contains four main directories: *data*, *results*, *scripts*, and *writing*. The *writing* directory is meant to contain the project output document, i.e. a thesis chapter or research paper. The *data* directory contains the sub-directories *definitions\_protocols*, *processed* and *raw*. The *raw* directory is the most important one, since it is here that files with all raw data should be stored. All raw data needed for the generation of project results should be stored in this directory, even if they are already stored elsewhere. The important thing to note about raw data is that it should be as close to initial data entry format as possible and should not be edited or handled by the user after it has been stored in the *raw* directory. There are however a set of guidelines that raw data is expected to adhere to. These will be outlined further below. The *scripts* directory will contain all computer code needed for data processing, cleaning and analysis. Ideally, they form a numbered set of scripts that, when run successively, reproduces all results referred to in the project output document from the raw data. The scripts should contain sufficient commentary to be human readable. Results generated by scripts can be written to the *results* directory. The analysis scripts should write the tables and figures referred to in the project output document to the corresponding *results/tables* and *results/figures* directories. As a by-product of running the scripts, a set of intermediary or processed data files may be produced. These should be written to the *data/processed* directory. This directory should also contain the distributable copies of the data, including complete meta-data. Finally, the *data/definitions\_protocols* directory may contain documents describing the specifics of data collection such as protocols and detailed descriptions of how variables in the data are defined. Ideally, variable definitions that will be distributed with the data refer to these documents when needed. Statements related to ethical and privacy guidelines followed can also be stored here.

Figure 1 Hierarchical directory structure for storing all data, scripts, results and written output for a research project.

## Data formatting requirements

Raw data files should adhere to a couple of basic formatting rules to ensure that they can be handles efficiently and uniformly. First of all, data in a single file or worksheet should be in the so-called “long” format, with each column representing a variable, factor or identifier. Use underscores instead of spaces in column names. Repeated observations on the same object or experimental unit (e.g. yields measured over different seasons) should be recorded on separate lines by adding a row for each season, repeating all information related to the object and adding a time column to indicate which timepoint the information in a row refers to. Dates should be formatted as YYYYMMDD. Second, numeric variables should have the unit added to the variable name, separated by a “$”, e.g. grain\_yield$kg\_ha. Use standard (SI) abbreviations for units but avoid special symbols that may cause problems when handling and storing data. **Please see the example data downloaded by the helper script to see what a data file should look like.**

## Data privacy issues

It is important to identify data-fields or variables that may be privacy sensitive. This includes all variables that can be traced back to a private individual such as names, phone numbers, location data. These data fields should be removed or anonymised when data is made available publicly.

## Meta data

Each raw data file should be linked to a file containing extensive meta-data. Meta-data is meant to provide all information needed to make data usable for others. It includes a data identifier, information on the project and project funding, author and contact information, data type, species or subjects the data refers to, geographic areas and dates covered.

## Set-up script

A standard *R* script is available that when run from the main project directory (with this directory set as “home”). It can be sourced from GitHub as follows (if you haven’t done so already):

1. Make an empty folder  
     
   2. Set it to working director in R  
     
   3. Paste the following in the terminal:

source("<https://raw.githubusercontent.com/jvheerwaarden/PPS_data_management/master/Master_Script_V1.1.R>")

This will automatically download scripts and files that:

1. creates the data management structures described above. Briefly, it will create the required directory structure, read any raw data files (.csv or .xlsx) stored in the project directory and place them in the *data/raw* directory. It will process the raw data into a corresponding processed .xslx file, containing 1. a meta-data worksheet with all required fields and 2. A list of all variables in the data with appropriate units if supplied and extra columns for variable definitions and privacy sensitivity. Please check the “example\_data.csv“ and “Meststof proef WUR.csv” data files in the *data/raw* directory and the associated “example\_data\_metadata.xlsx“ and “Meststof proef WUR\_metadata.xlsx” files in the *data/processed* directory.

2. It will place a set of R scripts in the *scripts* directory (numbered 0-2, and a functions file) that provide examples of reading, cleaning and analysing data. They are run automatically (by sourcing the *0\_build\_project.R* file) and output will be written to the *results/figures* and the *results/raw* directories. Please have a look at what the scripts do to get an idea of the logic behind them.

3. Will place the current Document and Readme in *writing* directory.