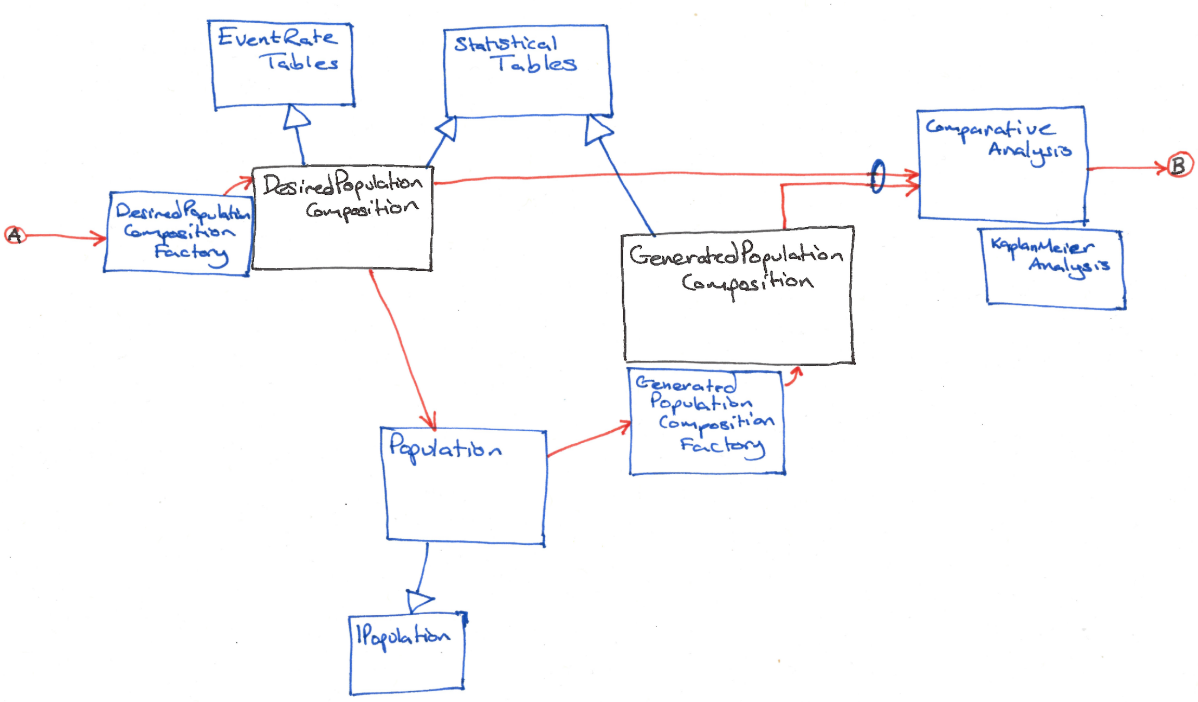
Population Model Version 3 – Documentation

The aim of this document is to inform of the overall structuring of the model, the finer details and reasoning of the implementation where it may be otherwise illusive. It is expected that this document will be read alongside the project's Javadoc and in the case of the finer details the inline comments and source code.

1 Model Overview

The key focus of this model was to focus on a structure that would be able to take input in the form of summative information about the desired composition of the generated population and then to be able to make a statistical assertion regarding the similarity of the desired and the generated population. The need for a robust statistical comparative approach was deemed necessary due to the complexity of modelling a population across many variables and summative statistics, as was evidenced in the earlier versions of our population models.



Detailed below is what is involved and expected in each stage of the model.

1.1 Data Input

Associated classes: ***DesiredPopulationConpositionFactory, DesiredPopulationComposition***

Inputted data represents the rate at which modelled events occur at in the desired population. The data can be provided for every year in the population. If not, then the approach by which data will be imputed is detailed in section [TO IMPLEMENT].

The events modelled are:

- Birth – specifically the proportion of females of a given age in a given year that give birth.
- Marriage – specifically the proportion of males of a given age marrying females of a given age in a given year.
- Death – specifically the proportion of people (divided by gender) of a given age that will die within one year from the given year.

In the case of birth and death the data form used is often described as a Lifetable or as the Kaplan-Meier method in the domain of demography and actuarial sciences.

The input and construction of this data into a collection is handled by the DesiredPopulationCompositionFactory and results in a DesiredPopulationComposition object.

1.2 Desired Population Composition

Associated Classes: ***DesiredPopulationComposition, EventRateTables, StatisticalTables, PopulationInformationCollection***

Once data has been placed into a DesiredPopulationComposition, the model now has an understanding of what characteristics the user wishes for the end population to exhibit. Access to the information identifying the rates and proportions at which events are desired to occur in the population is provided through the interface EventRateTables. The rate data that is returned is expressed in the form of Tables, details of which are explained in section 1.6 and for the association of Table formats with data types the JavaDoc is the best and persistently updated source.

The information regarding the desired information is also needed by our statistical approaches and often in specific formats, the StatisticalTables interface makes provision for this and is implemented by the GeneratedPopulationComposition class as well.

1.3 Generating the Simulated Population

Associated Classes: ***Population, DesiredPopulationComposition, EventRateTables, IPopulation, IPerson, IPartnership***

In the process of generating the simulated population calls are made to the DesiredPopulationComposition to access data about the rates and proportions that modelled events should occur to the generated population.

The simulation approach is to be able to define cohorts within each year of the simulation and then to apply to each cohort the number of events as specified by the DesiredPopulationComposition.

1.4 Deriving Generated Population Composition

Associated classes: ***Population, GeneratedPopulationCompositionFactory, StatisticalTables, GeneratedPopulationComposition***

Once a population has been generated we need to place it into a form that allows for comparative analysis. This is done by passing the Population to the GeneratedPopulationCompositionFactory that produces a GeneratedPopulationComposition object. The GeneratedPopulationComposition also implements the StatisticalTables interface like the DesiredPopulationComposition class and thus allows for statistical comparisons of the two to be made, this is outlined in section 1.5.

The process of creating the GeneratedPopulationComposition involves processing the generated population so as to create the summative data of the population as required by the StatisticalTables interface.

1.5 Statistical Comparison of Desired and Generated Populations

Associated classes: ***ComparativeAnalysis, KaplanMeierAnalysis, StatisticalTables, DesiredPopulationComposition, GeneratedPopulationComposition***

As detailed in the overview, the aim for the model is that the generated population can be statistically verified as being significantly similar to the desired population.

The ComparativeAnalysis class coordinates the retrieval of the required data from the different population composition classes and then makes use of the assisting statistical classes to compare the collected data.

For strict event based data that can be expressed in survivor table is compared using the Kaplan Meier method (Tierney et al., 2007; Kleinbaum & Klein, 2006). The method compares the two survival curves and can be used to calculate a Hazard Ratio which compares the level of risk in the desired and generated populations which can be used to indicate the similarity of the two populations.

1.6 Tables – Population Data Representation

Associated classes: ***Table, OneWayTable, TwoWayTable***

The handling of data in the model is a regular occurrence, therefore a table approach is provided. The tables can have either one or two look up variables. For example, in terms of death rates for men there is one look up variable, this being age. Whereas for marriage there are two look up variables, these been the age of the male and the age of the female. Example of the structuring of these tables and example tables for each event can be seen in the JavaDoc in the StatisticalTables interface.

2 Simulation Design

2.1 Terminology

To save confusion a set of definitions are set out here for any words to which ***emphasis*** is placed upon in this document.

population – the set of individuals who are alive in the simulation at an instant in time, this also implies knowledge of their inter-relatedness.
full population – the set of individuals who have been alive at any point within the simulation, this also implies knowledge of their inter-relatedness.

T_0 - The time in the simulation from which the model is required to produce a population that has ***integrity***.
T_S - Where we set the very beginning of the simulation to be to ensure the population at T_0 has ***integrity***.
T_E - The end time in out population model.

seed population - The population of people who will have been created and be reachable (by traversal of the population graph) at T_0.
genesis population - The population introduced synthetically at T_S (and shortly afterwards) which is then used as the population to begin the simulation with; which in turn becomes the ***seed population*** at T_0.

integrity - a full population is considered to have integrity where all people alive at T_0 have all the information (both within themselves and by traversal of the population graph) to fill in the required population outputs – in the initial case these are the ***event certificates***. A consideration of what integrity requires is given in section 2.2.1.1.

event certificates - The birth, marriage and death certificates that are used in Scotland/the UK to record events. And although the reality of these should not limit the extent of the population model, it is reasonably to use such to consider where we need to set T_S to be for our needs. *To be clear, it is possible to set the simulation date to begin earlier to suit the intended usage of the system and the population simulation is foremost concerned with producing a **full population** with a set of attributes which can then be output into the form as desired by the user – in the case of the original intention for the system this was birth, marriage and death certificates as used in Scotland.*

2.2 Seed Population

The problem being addressed: For the population generation project there is the need to be able to create a ***seed*** population to exist at T_0. This ***seed*** population needs to consist of a set of people who are capable of representing all information that will need to be expressed in the outputs of our system to give a ***full population*** with ***integrity*** between the times T_S and T_E.

The traditional way we have gone about addressing this problem has been to step back in time from T_0, creating a ***genesis seed*** population (and create relationships between the whole population at a given instant in time for all previous times) and then beginning the simulation assuming that ***integrity*** will be reached by T_0. This then leads to the question of how far back we need to step from T_S to be sure that the population will have ***integrity*** by T_0 and also how the initial ***genesis population*** should be constructed to allow for the simulation to be ran in its standard approach as early as possible – i.e. we don't want to write a really complex separate algorithm to generate a ***genesis population*** if we can achieve the same with a couple of intelligent tweaks to the standard approach we use for the rest of the simulation .

2.2.1 Conceptual Approach

2.2.1.1 What defines integrity for a given output format?

How we define and construct the ***genesis population***, which we begin our simulation with, effects what the ***seed population*** will look like at T_0. T_0, by definition, is the point in time from which all data that you wish to be present in your output format must be present in the simulation – i.e. the people alive at T_0 need to exist and have however much ancestry that your output format is able to express.

2.2.1.2 For our output format – how far back to start?

For our uses where the output format is the statutory records of Scotland for birth, marriages and deaths we need for all the data that is present on those records for events occurring at T_0 to be represented in the dataset at T_0. The information found on these records can be seen in the appendix in section 5.1.

Inspection of these records identify that we need to have information that requires information about people to whom events occurred before T_0. These informational dependencies are represented below in diagrammatic life segment format, a introduction to which can be found in the appendix in section 5.2. The labels on the diagrams indicate what information is required from individuals existing before T_0.

For a Birth Record occurring at T_0:

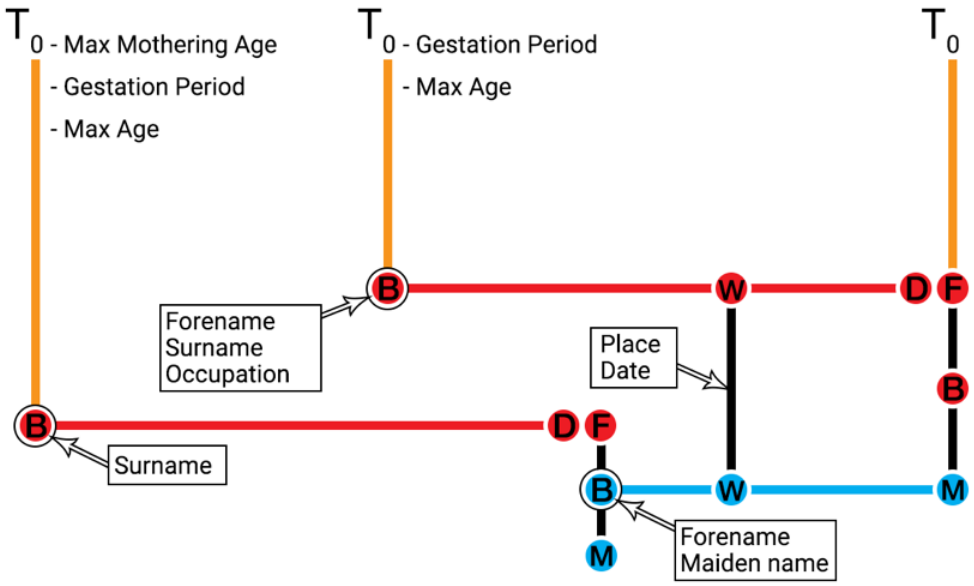


Figure 1 - A life segment diagram showing the individuals whose information appears on a birth record at T_0

For a Marriage Record occurring at T_0:

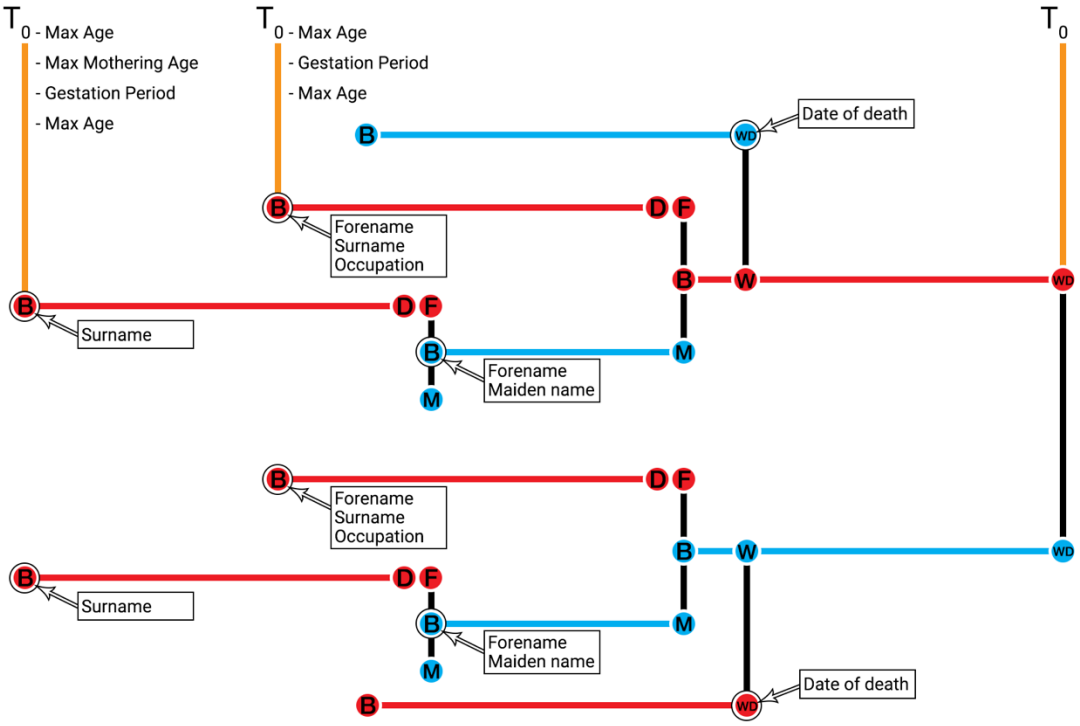


Figure 2 - A life segment diagram showing the individuals whose information appears on a marriage record at T_0

For a Death Record occurring at T_0:

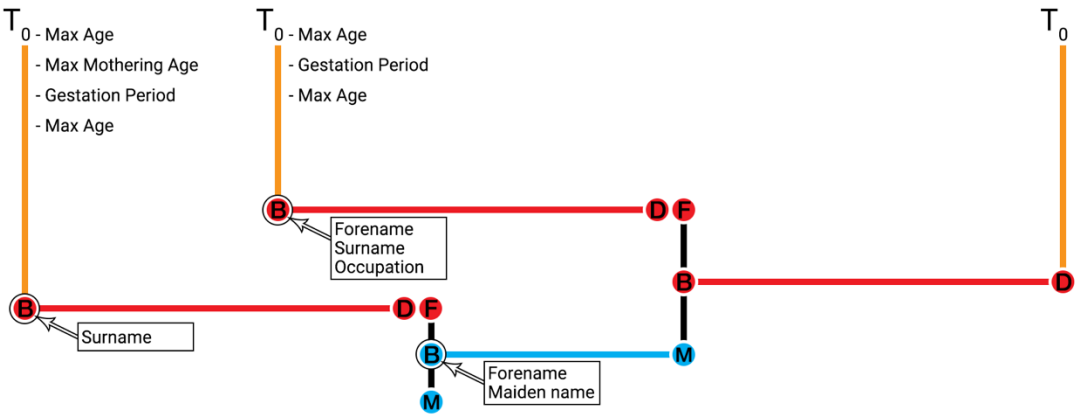


Figure 3 - A life segment diagram showing the individuals whose information appears on a death record at T_0

Given the details specified for our output format we can consider how far back it is necessary to begin, assuming we will allow for the population to run organically from T_S. As the furthest information reach of our records is to the birth of a grandparent for any record in T_0. Therefore, by considering the maximum time steps back to such an event we can identify the latest point in time that it is possible to exist and to have a direct implication for an ***event certificate*** at T_0.

However, there is an obvious recursion that causes for earlier individuals to have an indirect implication on ***event certificates*** at T_0, this being that a surname implies knowledge of the father. The nearer to T_0 we start the less levels of cousins of individual at T_0 are we exerting direct control over. To consider from a data linkage viewpoint; when linking ***event certificates*** existing at T_0 the

nature of the ***event certificate*** allows only for cousins to be potentially identified through the comparison of grandparents surnames (via mothers' maiden names) on the birth or death certificates. If we also consider that surnames are not unique in a population this further asserts that the links made at this level are weak although still loosely indicative of the underlying population structure. Therefore, there is value in controlling the population structure at this point, but the case to control it further back in time becomes progressively weaker.

2.2.1.3 How to start the population going?

We now are able to work out the date the first birth needs to happen in our population and that the first generation of people created don't need to have information regarding their parentage. Also to note, we want to create all people in the simulation at age zero and then to allow the simulation algorithm to handle aging and the progression of life events.

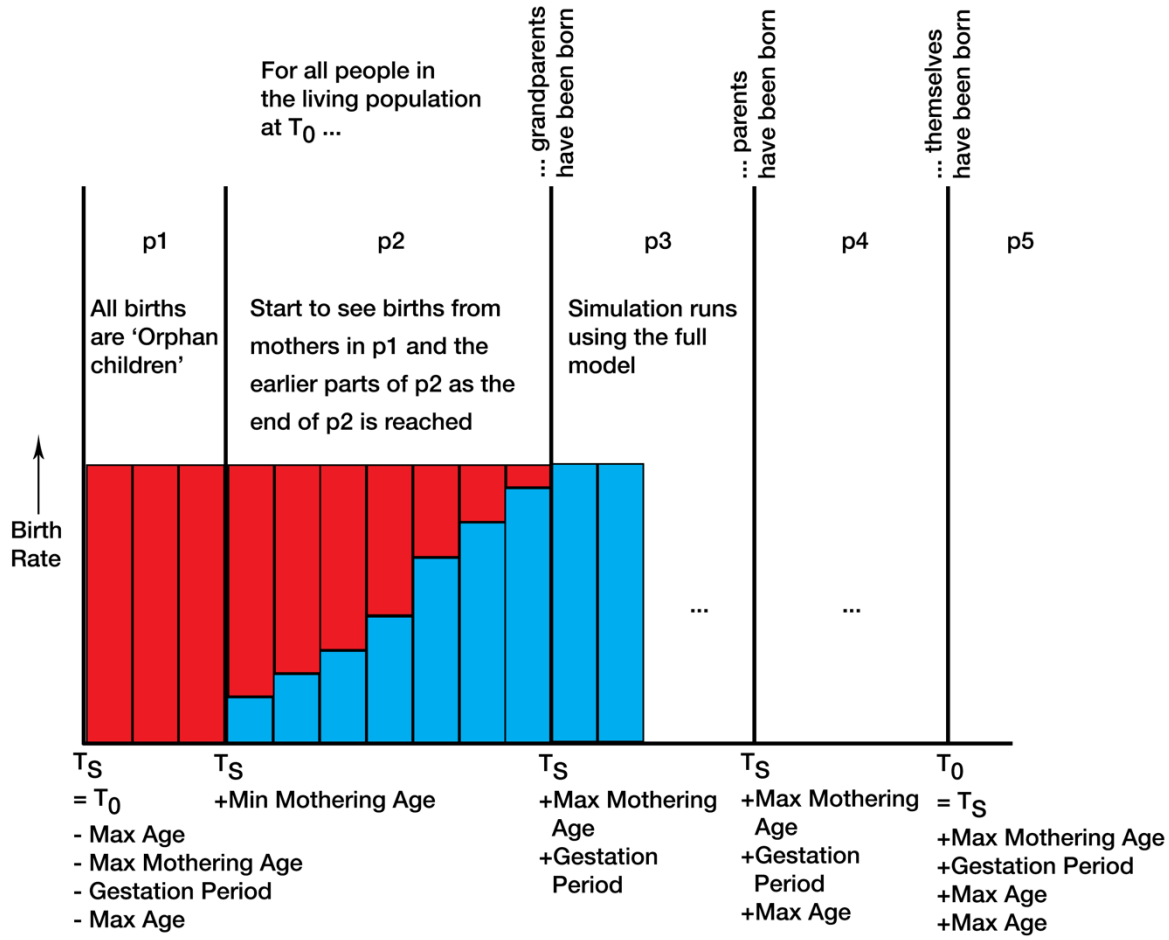


Figure 4 - Diagram of birth method changing with simulation time progression

We now have a concept that for the initial people in our population we only need to for them to mark the beginning of a family line and for us to be able to control all births that they are a parent to. This allows for us to imagine that these individuals could by spawned in the early stages of our simulation and then be used to create the parent generations and the generation who are alive at T_S and thus giving that generation its ***integrity.*** Let us term these individuals spawned without lineage as orphan children.

This introduces the question of how many orphan children should be created. This requires us to consider the birth rate and death rates the population will be exposed to between T_0 and T_S, from

this we can calculate the population growth rate (PGR). Using the PGR and the target population size at T_0 we can then calculate the expected population size at $T_{S+max\ age}$. This tells us the size of the population that the birth rate since T_S has lead to. We can consider the size of the population here in life table notation as **Σl(x)** (for the values of x from 0 to max age) and from which we can calculate **l(0)** - namely the number of people born in the first year following T_S.

2.2.1.4 How to transition between birthing approaches?

From T_S when orphan children begin to be spawned we also begin to run the full simulation model - checking for people to give birth, finding fathers, geographical movement, enforcing death. Obviously until any of the female orphan children reach the minimum mothering age there will not be any births. In this stage of the simulation we will keep track of how many natural births there are each year and by calculating the shortfall (considering the target PGR in this period and the death rate occurring in the simulation) add in the correct number of orphan children to the population each year. Up until time $T_{0+min\ mothering\ age}$ the births will be made up wholly of orphan children but between then and $T_{0+max\ mothering\ age}$ the number of natural births will increase meaning the shortfall decreases and thus the number of orphan children being spawned tapers away until in the year following $T_{0+max\ mothering\ age}$ the number is zero. From here on the full simulation runs without any correction from the orphan children adjustment.

2.2.1.5 How to know the orphan adjustment works?

// TO Fill IN – talk about the underlying maths?

2.2.2 Algorithmic integration

2.3 Birth

2.4 Multiple Births (twins, triplets, etc.)

2.5 Partnering

2.6 Separation

2.7 Death

3 Implementation Details

4 References

Kleinbaum, D. G., & Klein, M. (2006). *Survival analysis: a self-learning text*. Springer Science & Business Media.

Tierney, J. F., Stewart, L. A., Ghersi, D., Burdett, S., & Sydes, M. R. (2007). Practical methods for incorporating summary time-to-event data into meta-analysis. *Trials*, *8*(1), 16.

5 Appendix

5.1 Information in the Statutory records for Scotland

5.1.1 For Births

Information taken from: http://www.nrscotland.gov.uk/research/guides/statutory-registers/births

Most records include:

- forename of child
- surname of the child
- where born
- date of birth
- time of birth
- sex
- name, surname and occupation of father
- name and maiden surname of mother
- date and place of the parents' marriage
- signature, address (if not the place of birth) and relation of informant
- signature of registrar

For 1855 only information about:

- the parents' other children (living or dead)
- the ages and birthplaces of both parents was included.

5.1.2 For Marriages

Information taken from: http://www.nrscotland.gov.uk/research/guides/statutory-registers/marriages

Most records include:

- when, where and how married
- names of parties
- occupation of bride and groom
- whether parties were single, widowed or divorced
- their ages
- addresses of bride and groom
- name, surname and occupation of fathers
- name and maiden surname of mothers
- signature of witnesses
- where the marriage was registered and signature of registrar

In 1855 entries you will also find information about:

- the place and date of birth of both parties
- the number of any children by former marriages (whether living or dead)
- the number of previous marriages (if any)

From 1965 onwards:

- date of birth of both parties
- addresses of witnesses
- occupations of the mothers of the bride and groom has been recorded.

5.1.3 For Deaths

Information taken from: http://www.nrscotland.gov.uk/research/guides/statutory-registers/marriages

Most records include:

- forename of deceased
- surname of deceased
- occupation
- marital status
- if married the name of the spouse
- when and where died
- sex
- age
- name, surname and occupation of father
- name and maiden surname of mother
- if parents are deceased
- cause of death
- signature, address (if not where death occurred) and relation of informant
- where and when the death was registered and the signature of registrar.

For 1855 only the following information was included:

- the names and ages of all children of the deceased (whether living or dead)
- the place of burial and the undertaker involved
- the date and place of birth of the deceased
- how long they had been in the district.

From 1965 onwards:

- the occupation of the spouse (and later civil partner)
- the occupation of the mother of the deceased

5.2 Introduction to life segment diagrams

Life segment diagrams are a way of diagrammatically representing related individuals, the events occurring to them as individuals and also how they are related. Following are a couple of examples that represent different configurations of life segment diagrams and also how limited information is to be displayed.

The basic things to know first:

1. Red lines represent males
2. Blue lines represent females
3. Circles represent events, with the letter within representing the type:
    a. B – Birth
    b. D – Death
    c. F – Fathering
    d. M – Mothering
    e. W – Wedding
4. Black vertical lines represent the joining together of individuals at the events of:
    a. Wedding
    b. Childbearing
5. Time progresses from left to right

5.2.1 Nuclear Family Example

Here we can see two males and one female. The male born first produces a male with the female. The child experiences no events in life except for death.

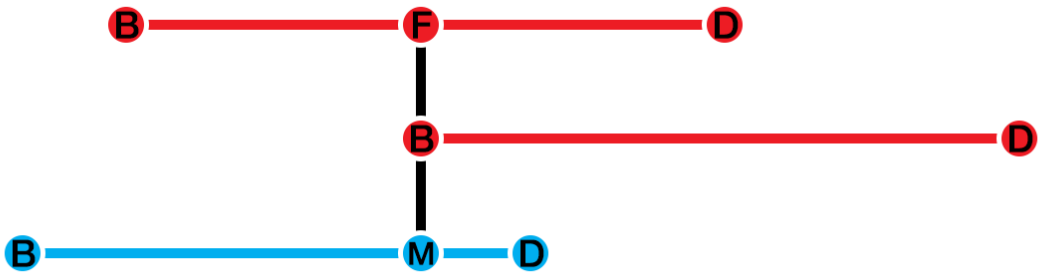


Figure 5 - Life segment diagram of a nuclear family

5.2.2 Marriage Example

Here we can see a male and a female who get married mid-way through the male's life. The lack of a continuing segment for the female indicates that this information is not, or needs not, be available to us.

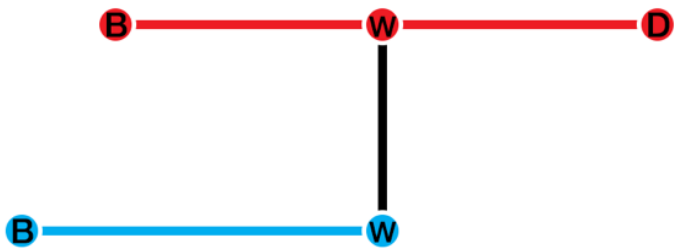


Figure 6 - A life segment diagram representing a marriage event

5.2.3 Posthumous Birth Example

Here we can see a male and a female producing a child. However, at the point of the birth of the child the father has died. Also, this diagram shows that we have no information about the mother other than her part as the mother of the child.

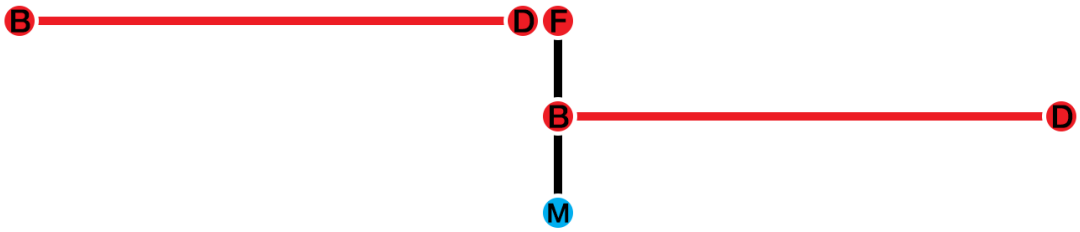


Figure 7 - A life segment diagram representing a posthumous birth event