Applied programming for Biosciences

Thor Wessel Lindberg vjx107@alumni.ku.dk

Introduction

This project seeks to answer various questions about a database of people, by reading the database in Python, and producing functions which analyze the database. This report contains a description of the database and its structure, a thorough description of the functions produced to solve analytical questions, and a section containing questions and the results produced to answer these questions.

Reading and analyzing the database is motivated by the opportunity for statistical insight into the differences between people, their children, and how their attributes change over periods of time. This information gives insight into how the different attributes impact our evolution in different ways.

In addition to a description of the materials, methods and results, the following files are appended with this report, and are required for understanding this project:

- Jupyter notebook (Methods.ipynb) containing all of the functions.
- Python file (methods.py) containing all of the functions. It can be imported for analysis.
- Jupyter notebook (Notebook.ipynb) which imports the functions and produces the results which answer the questions given in this report. It is designed to be user-friendly and to avoid common errors such as an incorrect file path.
- Database file (people.db) containing the data for the analysis.
- ZIP file (Results.zip) containing the results produced by the notebook. Not all results from this file are presented in the results section of this report, as some are too extensive to present within the limits given for this report, but all results are accessible through this file.

Materials

The database is stored as a .db file (people.db), and read line-by-line using Python. It consists of a header which describes the data, and paragraphs which represent a person. Each line in a paragraph is an attribute of the person it describes. These attributes are **CPR number, first name, last name, height, weight, eye color, blood type, and the CPR numbers of their children (if they have any)**. All information in the database was recorded in the year 2000, and all people present in the database were alive at the time of recording. Rules about the construction of CPR numbers were not followed, as publishing CPR numbers is illegal, but the date is in the 20th century, and the last digit represents a male if the number is odd, or a female if the number is even.

Below we look at the structure of a paragraph representing a person in the database, and discuss the impact of its structure on the functions. Each line begins with a description of the line's data, and the description and data are separated by a colon followed by a space. The lines will be read into a dictionary, where each key represents a person, and each value is a dictionary itself, which keys are the description for each attribute, and its values the data corresponding to the attribute.

- CPR (Central Person Registry) consists of 6 numbers representing a date of birth (date/month/year), and 4 numbers, separated by a hyphen. It is worth noting that the last number in the CPR is an odd number if the person's gender is male, and an even number if the person's gender is female. Due to the hyphen, the CPR attribute can't be read into Python as a integer, unless it is separated at the hyphen. This isn't worth doing, as we don't need to add, subtract, multiply etc. any CPR numbers.

- A CPR number always has the same length and structure, and for this reason we will read it into Python as a string, and index it to determine other attributes like age and gender.
- The first and last names are read into Python as strings, and are combined, as having them separate is not relevant for any analysis.
- Height and weight are read into Python as integers, as they are attributes which would be useful as integers for analyzing averages, percentages, correlations etc.
- Eye color and blood type are read into python as strings, since they are only letters and symbols.

Aside from the database, some functions use information, such as the average height for an age for each gender, or which blood types can donate and receive blood from which blood types. The sources of this information are within the description of the respective functions in the section on Methods.

Methods

All functions for this project are contained in the same Python file (methods.py), which is also appended as a jupyter notebook (Methods.ipynb), which is separated into different sections, with each function having its own section. All functions are called in a jupyter notebook (Notebook.ipynb), where they're imported from the Python file. This notebook requires its user to input the path to the folder containing the database file, which is also appended. It then calls the imported functions, if the input is correct. All results are saved as .csv or .png files for tables or plots respectively, in the same folder as the database file. All functions with an output call the save_csv or save_plot function at their end.

There are four different function inputs in the notebook containing functions.

- Output is a list of lists, where each line represents a row in a table.
- Directory is the path to the folder containing the database, and the destination for saving files.
- Filename is the given name for a file being saved, excluding its file format.
- Database is the name of the database file, including its file format.

save_csv(output, directory, filename)

- Loops through a list of lists, converting each element in the list to a string, and writing each list into a .csv file, with each element being separated by a comma, and appended with a newline.
- Prints a message confirming that the output has been saved in the given directory.

save_plot(plot, directory, filename)

- Saves a pyplot plot in a given directory with a given filename in the format PNG, and closes the plot.
- Prints a message confirming that the plot has been saved in the given directory.

read_file(directory, filename)

- Creates a dictionary to contain the people in the database, and a temporary dictionary to contain the attributes for each person in the database, which is reset for each person.
- Reads the database file line-by-line, adding its data to the dictionary for a person, if the line is an attribute of a person. A line containing an attribute is split, and the data following the colon is added. The dictionary for a person is added to the dictionary for people whenever an empty line is encountered, and the dictionary for a person is reset.

- Returns the dictionary for people, to be used for analysis by other functions.

count_people(directory, database)

- Outputs the length of the database, which corresponds to the amount of people in the database.

count_parents(directory, database)

- Creates a list of parents, which will contain their CPR numbers.
- Loops through the keys and values in the database, appending the key to the list, if the value contains children, which means the person has children and is a parent.
- Outputs the length of the list, which corresponds to the amount of parents in the database.

count_children(directory, database)

- Creates a list of children, which will contain their CPR numbers.
- Loops through the values in the database, appending the CPR numbers of a parent's children, if the CPR number isn't already in the list, to avoid duplicates from multiple parents.
- Outputs the length of the list, which corresponds to the amount of children in the database.

parents_with_multiple_partners(directory, database)

- Creates a list of parents with multiple partners, which will contain their CPR numbers.
- Loops through the keys and values in the database, creating a list of a parent's children's other parent's CPR numbers. This list resets for each parent in the database, and if the parent has children with multiple partners, their CPR number will be appended to the list of parents with multiple partners.
- Outputs the length of the list of parents with multiple, which corresponds to the amount of parents in the database who have children with multiple partners.

count_grandparents(directory, database)

- Creates a list of grandparents, which will contain their CPR numbers.
- Loops through the keys and values in the database, appending the CPR number of everyone whose children have children, which corresponds to that person being a grandparent.
- Outputs the length of the list of grandparents, which corresponds to the amount of grandparents.

age_gender_distribution(directory, database)

- Creates a list of ages, which will contain the ages of people in the database.
- Creates a list of women's age and a list of men's ages.
- Loops in a range from the minimum to the maximum age in the list, jumping 20 years each loop, and appending the percentage of women whose age is in the given range to a table of output, and their ages to the lists of women's ages and men's ages respectively.
- Appends the average age, average age for women, and average age for men to a table of output.
- Outputs the table, with descriptive headers.

first_time_father(directory, database)

- Creates a list of ages, which will contain the ages of fathers when they had their first child.
- Loops in a range from the minimum to the maximum age in the list, jumping 3 years each loop, appending the percentage of first-time fathers in the given age range to a list for a pie plot.
- Creates a pie plot, and outputs it with a descriptive title.
- Appends the average age for first-time fatherhood to a table of output.
- Outputs the table, with descriptive headers.

first_time_mother(directory, database)

- Creates a list of ages, which will contain the ages of mothers when they had their first child.
- Loops in a range from the minimum to the maximum age in the list, jumping 3 years each loop, appending the percentage of first-time mothers in the given age range to a list for a pie plot.
- Creates a pie plot, and outputs it with a descriptive title.
- Appends the average age for first-time motherhood to a table of output.
- Outputs the table, with descriptive headers.

family_patterns(directory, database)

- Creates a list of ages, which will contain the ages of parents in the database.
- Loops in a range from the maximum to the minimum age in the list, jumping 10 years each loop, appending the average amount of children for a parent in the given age range to a list for a line plot.
- Reverses the list, to plot the ages in descending order and the decades in ascending order.
- Creates a line plot, with a description of each axis, and outputs it.

average_cousins(directory, database)

• Creates a dictionary which keys is people in the database and values is cousins in the database.

• Loops through the keys and values in the database, adding their CPR number to the dictionary if they have cousins, along with a list of the CPR numbers of their cousins.

• Converts the lists of cousins from the dictionary into a list of the amounts of cousins for each person.

• Outputs the average amount of cousins in the list.

height_correlation(directory, database)

• Creates a list of heights, which will contain the heights of people in the database.

• Loops in a range from the minimum to the maximum height in the list, jumping 10 cm. each loop, appending the percentage of people with children and a height in the given range, and the percentage of people without children and a height in the given range, to a table of output.

• Outputs the table, with descriptive headers.

weight_correlation(directory, database)

• Creates a list of weights, which will contain the weights of people in the database.

• Loops in a range from the minimum to the maximum weight in the list, jumping 10 kg each loop, appending the percentage of people with children and a weight in the given range, and the percentage of people without children and a weight in the given range, to a table of output.

• Outputs the table, with descriptive headers.

tall_parents(directory, database)

• Defines a dictionary which keys are ages from 0 to 20, and values are dictionaries of the average height in cm. for women and for men in the given age. This dictionary is sourced from the links below.

https://www.sundhed.dk/borger/patienthaandbogen/boern/illustrationer/tegning/vaekstkurver-piger-0-20/

https://www.sundhed.dk/borger/patienthaandbogen/boern/illustrationer/tegning/vaekstkurve-drenge-0-20/

• Creates a list of heights of parents with a height above the average for their age.

• Loops in a range from the minimum to the maximum height in the list, jumping 5 cm. each loop, appending the percentage of parents with a height in the given range and a child with a height above the average for their age, and the percentage of parents with a height in the given range and a child with a height below the average for their age to a table of output.

• Outputs the table, with descriptive headers.

overweight_parents(directory, database)

- Creates a list of the Body Mass Indexes of overweight parents in the database.
- Loops in a range from the minimum to the maximum BMI in the list, jumping 5 BMIs each loop, appending the percentage of overweight parents with a BMI in the given range and an overweight child, and the percentage of overweight parents with a BMI in the given range and a non-overweight child to a table of output.
- Outputs the table, with descriptive headers.

blood_to_children(directory, database)

- Defines a dictionary which keys are blood types, and values lists of blood types they can donate to.
- Creates a dictionary of parents who can donate blood to their children, by looping through the keys and values in the database, adding the CPR number of a parent and a list of the CPR numbers of children they can donate blood to, if they have such grandchildren. This dictionary is sourced from this link http://www.thebloodcenter.org/Donor/BloodFacts.aspx
- Loops through the dictionary of parents who can donate blood to their children, appending the names and bloodtypes of the parent and their children to a table of output.
- Outputs the table, with descriptive headers.

blood_to_grandparents(directory, database)

- Defines a dictionary which keys are blood types, and values lists of blood types they can donate to.
- Creates a dictionary of grandparents who can receive blood from their grandchildren, by looping through the keys and values in the database, adding the CPR number of a grandparent and a list of the CPR numbers of grandchildren they can receive from, if they have such grandchildren.
This dictionary is sourced from this link http://www.thebloodcenter.org/Donor/BloodFacts.aspx
- Loops through the dictionary of grandparents who can receive blood from their grandchildren, appending the names and bloodtypes of the grandparent and their grandchildren to a table of output.
- Outputs the table, with descriptive headers.

eye_color_heritage(directory, database)

- Loops through the keys and values in the database, finding the parents of a child, and checking if both parents have the eye color grey or blue, appending their eye color to a list. If both parents have grey or blue eyes, the name, eye color, parents’ names, and parents’ eye colors, are added to a table of output. It then outputs the table, with descriptive headers.

height_evolution(directory, database)

- Creates a list of ages, which will contain the ages of people in the database.
- Loops in a range from the minimum to the maximum age in the list, jumping 10 years each loop, and appending the ages of people with an age within the given range to a list for a line plot.
- Reverses the list, to plot the ages in descending order and the decades in ascending order.
- Creates a line plot, with a description of each axis, and outputs it.

weight_evolution(directory, database)

- Creates a list of weights in kg, which will contain the weights of people in the database.
- Loops in a range from the minimum to the maximum weight in the list, jumping 10 kg each loop, and appending the weights of people with an age within the given range to a list for a line plot.
- Reverses the list, to plot the weights in descending order and the decades in ascending order.
- Creates a line plot, with a description of each axis, and outputs it.

Results

How many people are in the database?	500
How many parents are in the database?	236
How many children are in the database?	305
How many parents have children with more than one person?	0
How many grandparents are in the database?	104
What is the average amount of cousins for people with cousins?	3.326086957
What is the average age for first-time fatherhood?	23.06779661
What is the average age for first-time motherhood?	22.79661017

Is the distribution of first-time fatherhood age as expected?

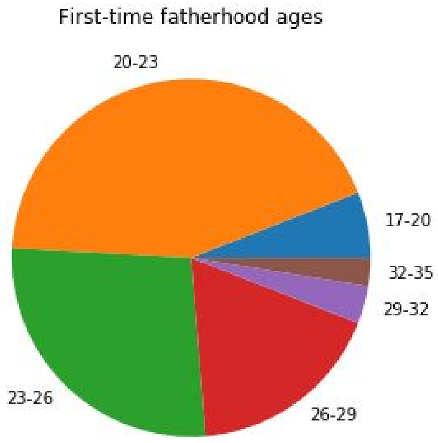
According to statistics from Danmarks Statistik, these numbers are lower than expected, as men on average become first-time fathers closer to the beginning of their 30s than the beginning of their 20s.

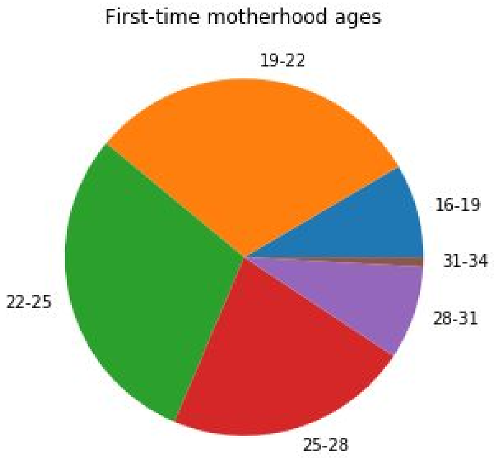
Source: https://www.dst.dk/da/Statistik/Publikationer/gennemsnitsdanskeren

Is the distribution of first-time motherhood age as expected?

According to statistics from Danmarks Statistik, these numbers are lower than expected, as women on average become first-time mothers in the beginning of their 30s and not the beginning of their 20s.

Source: https://www.dst.dk/da/Statistik/Publikationer/gennemsnitsdanskeren





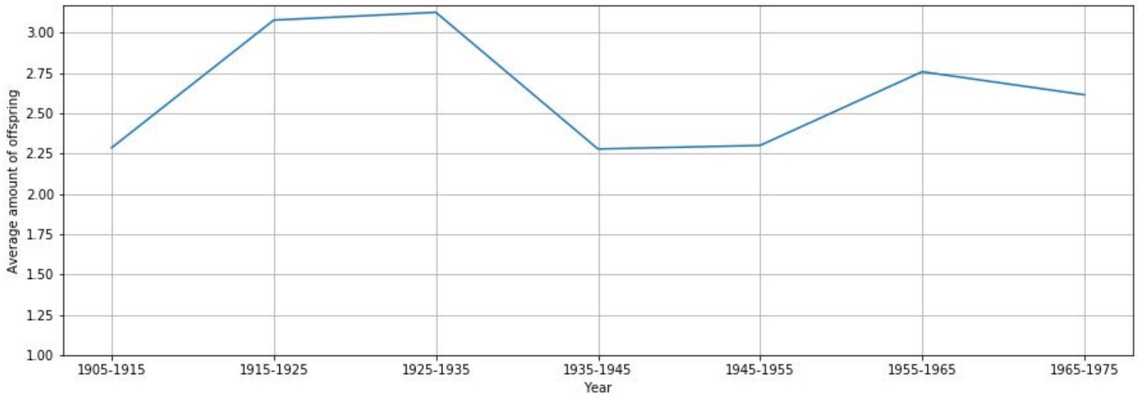
Is the distribution of age and gender as expected? (include average)

According to statistics from Danmarks Statistik, there is roughly an equal amount of women and men in each age range. The results below differ from the statistics, in that the percentage of women decreases with age, rather than increasing.

Source: https://www.dst.dk/da/Statistik/emner/befolkning-og-valg/befolkning-og-befolkningsfremskrivning/folketal

Age	**% of all**	**% of women in range**	**% of men in range**
1-21	26.4	54.54545455	45.45454545
21-41	29.8	53.69127517	46.30872483
41-61	23.8	51.2605042	48.7394958
61-81	12.4	46.77419355	53.22580645
81-101	7.6	39.47368421	60.52631579
Average age	38.848		
Average age for women	36.95330739		
Average age for men	40.85185185		

How do family patterns change over time?



What is the correlation between height and having children?

Height (cm)	% with children	% without children
160-170	53.60824742	46.39175258
170-180	52.29357798	47.70642202
180-190	40.77669903	59.22330097
190-200	42.55319149	57.44680851
200-210	46.39175258	53.60824742

What is the correlation between weight and having children?

Weight (kg)	% with children	% without children
55-65	49.5049505	50.4950495
65-75	49.33333333	50.66666667
75-85	41.75824176	58.24175824
85-95	46.08695652	53.91304348
95-105	49.15254237	50.84745763

Do tall people have tall children? (taller than average for their age)

Height of parents (cm)	% of children with height above average	% of children with height of average or below
171-176	91.66666667	8.333333333
176-181	82.22222222	17.77777778
181-186	83.87096774	16.12903226
186-191	69.23076923	30.76923077
191-196	80.43478261	19.56521739
196-201	81.13207547	18.86792453
201-206	82.69230769	17.30769231
206-211	73.33333333	26.66666667

Do overweight people have overweight children? (overweight = 25 BMI or over)

BMI of parents	% of children with BMI of 25 or over in range	% of children with BMI under 25 in range
25-30	49.64539007	50.35460993
30-35	44.82758621	55.17241379
35-40	38.0952381	61.9047619

Which people can donate blood to their children?

Results are too extensive to display here, and can be found in the file blood_to_children.csv

Which people can donate blood to their grandparents?

Results are too extensive to display here, and can be found in the file blood_to_grandparents.csv

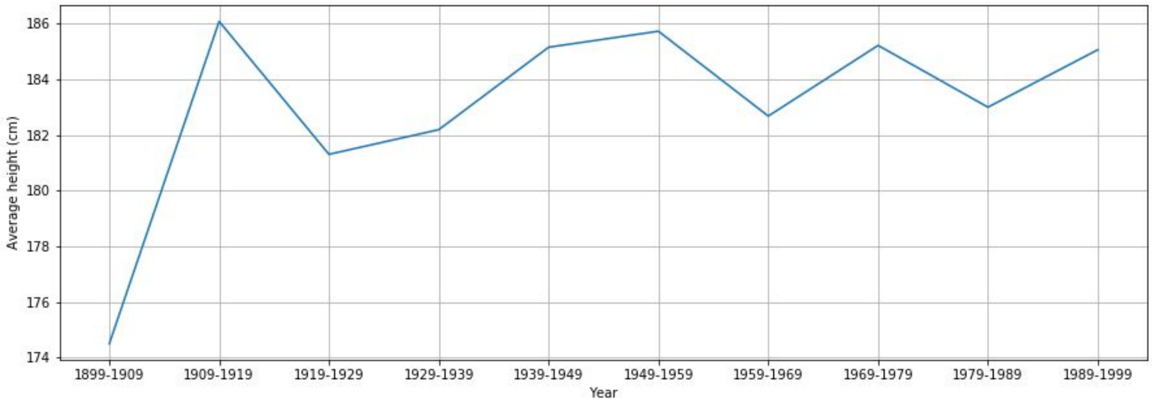
Are there any children with dark eyes whose parents both have light eyes? (blue or grey)

Results are too extensive to display here, and can be found in the file eye_color_hertiage.csv

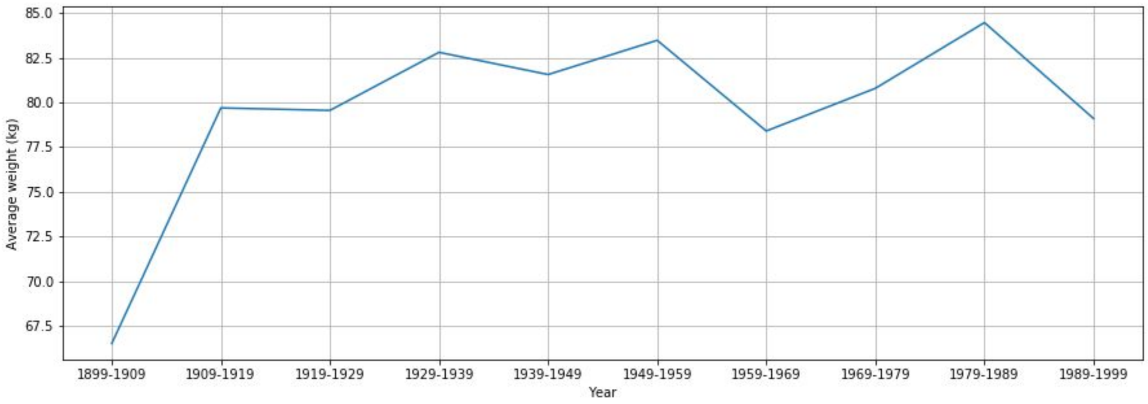
Below is a small sample, showing that there are some people with dark eyes, despite both of their parents having blue or grey eyes. This should not be the case according to the theory of heritage.

Child	Eye color	Parent color	Parent color
Abelone Ibsen	Black	Blue	Grey
Odin Rapacki	Black	Grey	Grey
Sanne Winther	Black	Blue	Blue
Ellen Karlsen	Green	Blue	Grey
Jens Pedersen	Green	Blue	Grey
Ben Karlsen	Red	Blue	Grey

How has average height changed over time?



How has average weight changed over time?


Average weight (kg)
Year