ti_pedigree

TI-Nspire CX CAS II-T
TI-Nspire CX CAS Student Software

v. 1.0

Table of Contents

Introduction

In high school biology and introductory college biology classes, you are often asked to solve pedigree problems related to monogenic disorders and traits. ti_pedigree is a Python (better: MicroPython) module developed for the TI-Nspire CAS calculator that to a certain extent automates the sometimes burdensome task of solving a pedigree.

This Python module can handle most of the common cases when analyzing pedigrees and the most common modes of inheritance or inheritance patterns (autosomal dominant/recessive, sex-linked (X) dominant/recessive, mitochondrial DNA, and Y-chromosomal) but it cannot address e.g. incomplete penetrance or sex-influenced dominance. It gives you the complete table of weighted solutions, but you must use your judgment to decide which solution is correct.

The commonly accepted *prevalences of monogenic disorders* in the general population are [1]:
autosomal dominant = 3/1000 to 9.5/1000
autosomal recessive = 2/1000 to 2.5/1000
sex-linked (X) = 0.5/1000 to 2/1000
Y-chromosomal and mtDNA = rare

Around 8,000 monogenic disorders have been cataloged [2], and autosomal recessive disorders are the most numerous, followed closely by autosomal dominant disorders. Sex-linked (X) recessive disorders are more numerous than sex-linked (X) dominant, which are rare. mtDNA and Y-chromosomal disorders are extremely rare. mtDNA disorders can follow any other inheritance pattern if the microdeletion site is in the nuclear mtDNA.

Solving simple pedigrees

Here we have a simple pedigree (Pedigree 1).

Pedigree 1
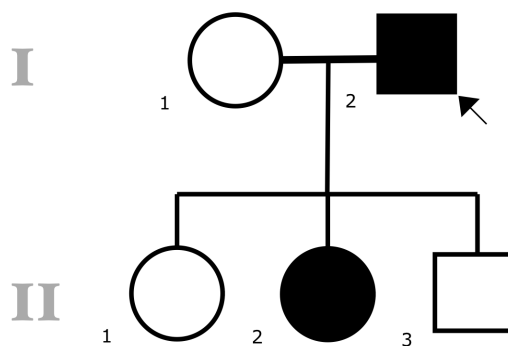
First, we must convert the pedigree into a suitable format. We create a new pedigree using the pedigree class. Use the following notation:

0 = unaffected (i.e. healthy)
1 = affected

We should use the syntax:

var = pedigree(father, mother, [boys], [girls])

Where:

father = 1 or 0 (*father must be defined* and it cannot be empty)
mother = 1 or 0 (*mother must be defined* and it cannot be empty)
boys = [a list of ones and/or zeros] (use [] for missing variable)
girls = [a list of ones and/or zeros] (use [] for missing variable)

The pattern of inheritance is later inferred entirely from the given phenotypes. Let's convert Pedigree 1 into a new pedigree and store it to a variable a:

```
>>> a = pedigree(1,0,[0],[1,0])
```

We can solve a simple pedigree by using the tbl() method. If you don't know if the pedigree is dominant or recessive, use the default argument (the default argument is "" with the tbl() method, see further below on page 8):

pedigree(father, mother, [boys], [girls]).tbl()

This method will return a table of all of the possible solutions. The following notation is used in the results:

M = mother
F = father
M+F = mother and father
het = heterozygous
hom = homozygous
**+(+)** = more likely
-(-) = less likely

Let's solve the pedigree that we created.

```
>>> a.tbl()
autosomal dominant
autosomal recessive (M het)
sex-linked (X) recessive (M het)
```

And that is our solution. As promised the solution is a complete table of all of the possible solutions. You would have to use contextual information to decide which inheritance pattern is the correct one. A plus sign would mean that the solution is more likely and a minus sign would mean that the solution is less likely.

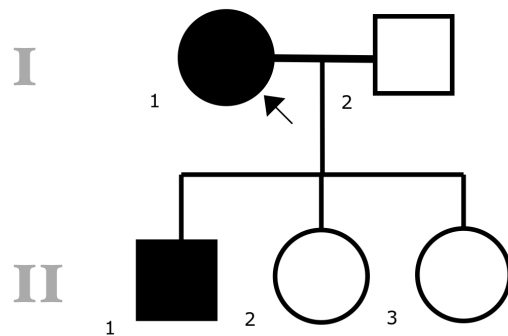Here are some more examples of solving simple pedigrees.
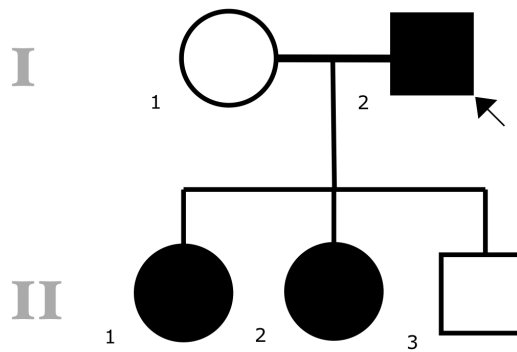
Pedigree 2

```
>>> pedigree(1,0,[0],[1,1]).tbl()
+sex-linked (X) dominant
autosomal dominant (F het)
-autosomal recessive (M het)
-sex-linked (X) recessive (M het)
```
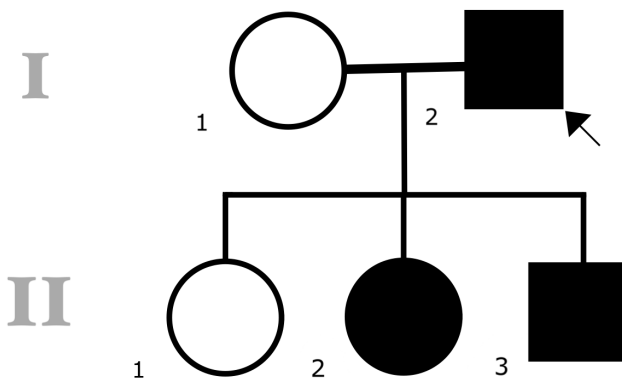
Pedigree 3

Pedigree 3:

```
>>> pedigree(0,1,[1],[0,0]).tbl()
+sex-linked (X) recessive
autosomal dominant
-sex-linked (X) dominant (M het)
-autosomal recessive (F het)
```
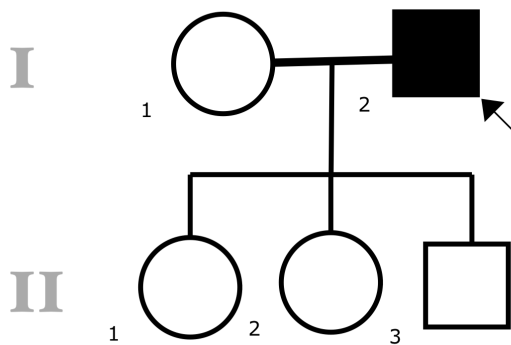
Pedigree 4

Pedigree 4:

```
>>> pedigree(1,0,[1],[0,1]).tbl()
autosomal dominant
autosomal recessive (M het)
sex-linked (X) recessive (M het)
```
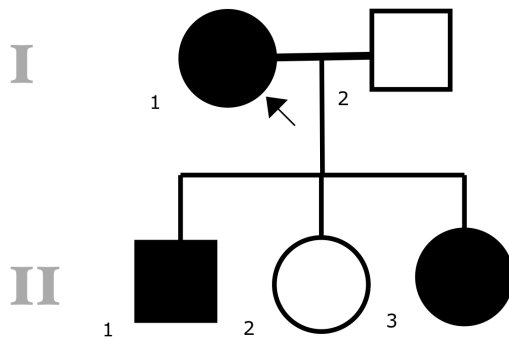
Pedigree 5

Pedigree 5:

```
>>> pedigree(1,0,[0],[0,0]).tbl()
autosomal dominant
autosomal recessive
sex-linked (X) recessive
(--mtDNA)
```
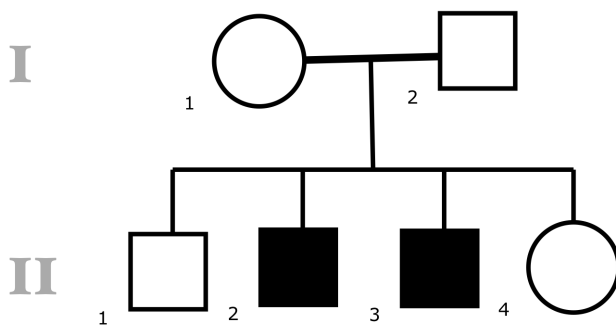
Pedigree 6

Pedigree 6:

```
>>> pedigree(0,1,[1],[0,1]).tbl()
+sex-linked (X) dominant
autosomal dominant
autosomal recessive
```

If you know for sure that the pedigree is *dominant or recessive*, you can use *"dominant"* or *"recessive"* as an argument with the *tbl()* method (Pedigrees 7 and 8).

pedigree(father, mother, [boys], [girls]).tbl("dominant")
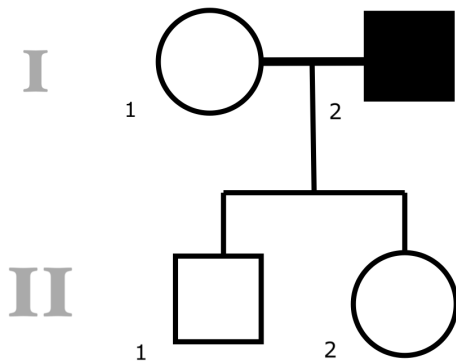**pedigree(father, mother, [boys], [girls]).tbl("recessive")**

Pedigree 7

Pedigree 7:

```
>>> pedigree(0,0,[0,1,1],[0]).tbl("recessive")
+sex-linked (X) recessive (M het)
```

```
autosomal recessive (M+F het)
```

Pedigree 8

```
>>> pedigree(1,0,[0],[0]).tbl("dominant")
autosomal dominant
```

You can also use the domrec() method, if it is applicable[1], to find out if the pedigree is dominant or recessive and then solve it.

dr = pedigree(father$_{dr}$, mother$_{dr}$, [boys$_{dr}$], [girls$_{dr}$]).domrec()
pedigree(father, mother, [boys], [girls]).tbl(dr)

If you want to print the result just use print(dr). Here is an example of the proper use of the *domrec() method* (Pedigree 9):

---

[1] The parent's phenotypes have to be the same and at least some of the children's phenotypes should differ from their parents.

Pedigree 9
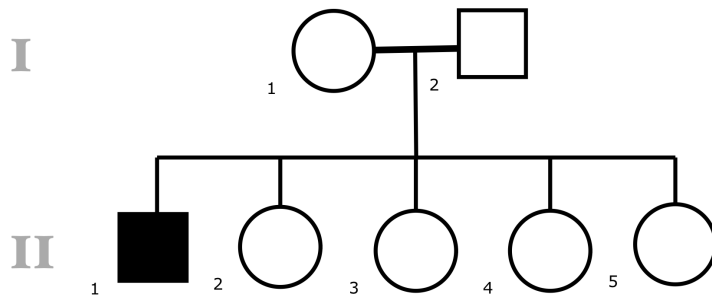
I
1 2

II
1 2 3 4 5

:

```
>>> dr = pedigree(0,0,[1],[0,0,0,0]).domrec()
>>> pedigree(0,0,[0,1,1],[0]).tbl(dr)
+sex-linked (X) recessive (M het)
autosomal recessive (M+F het)
```

Or:

```
>>>pedigree(0,0,[0,1,1],[0]).tbl(pedigree(0,0,[1],[0,0,0,0]).domrec
())
+sex-linked (X) recessive (M het)
autosomal recessive (M+F het)
```

The disorder is sex-linked (X) recessive.

Solving more complex pedigrees

If the pedigree contains two or more sub-pedigrees (simple pedigrees), you can solve the whole pedigree by compounding the results of the simple pedigrees. First, you must create the simple pedigrees. For *n sub pedigrees*:

a_1 = pedigree(father$_1$, mother$_1$, [boys$_a$], [girls$_a$])

...

a_n = pedigree(father$_n$, mother$_n$, [boys$_n$], [girls$_n$])

Then use the pdsolve() function to solve the whole pedigree. The first argument should be a list of the simple pedigrees. The default output format is a table, but it could be a list as well (you can leave out the second argument):

pdsolve([a$_1$,..., a$_n$], "tbl")

Pedigree 10 is a manageable example with just two sub-pedigrees.
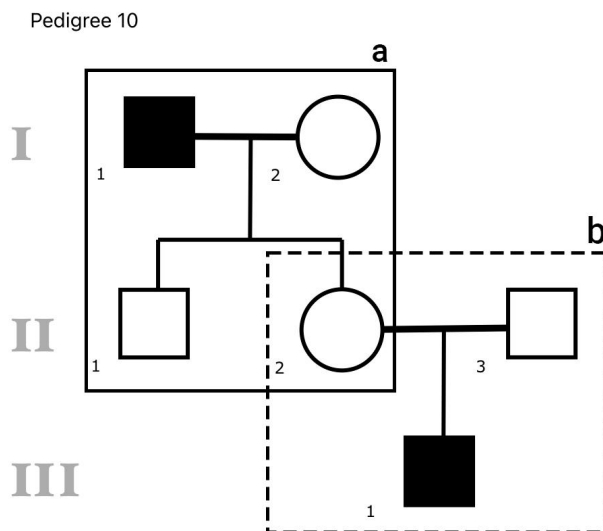
Pedigree 10

We can solve it like this:

```
>>> a = pedigree(1,0,[0],[0])
>>> b = pedigree(0,0,[1],[])
>>> pdsolve([a,b]))
Mode of inheritance:
++sex-linked (X) recessive
autosomal recessive
```

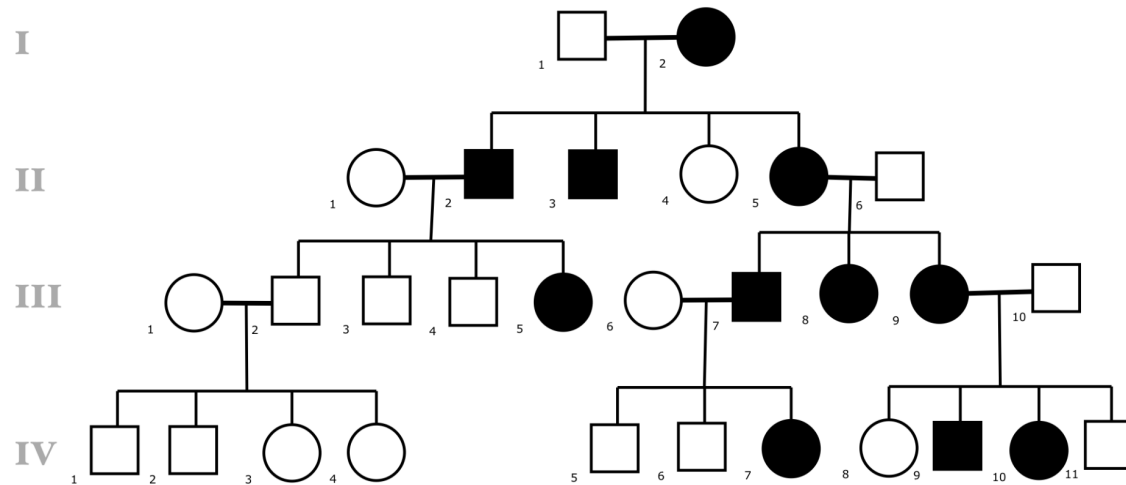Next, we go through some examples of solving more complex pedigrees with different modes of inheritance. The procedure is always similar, no matter how complicated the pedigree is.

Pedigree 11

Pedigree 11:

```
>>> a = pedigree(0,1,[1,1],[0,1])
>>> b = pedigree(1,0,[0,0,0],[1])
>>> c = pedigree(0,0,[0,0],[0,0])
>>> d = pedigree(0,1,[1],[1,1])
>>> e = pedigree(1,0,[0,0],[1])
>>> f = pedigree(0,1,[0,1],[0,1])
>>> pdsolve([a,b,c,d,e,f])
Mode of inheritance:
+sex-linked (X) dominant
+autosomal dominant
autosomal recessive
```

The disorder is autosomal dominant.

Pedigree 12

Pedigree 12:

```
>>> a = pedigree(0,1,[1],[0,1])
>>> b = pedigree(0,0,[],[0,0])
>>> c = pedigree(1,0,[1],[1])
>>> d = pedigree(0,1,[0,1],[])
>>> pdsolve([a,b,c,d])
Mode of inheritance:
+autosomal dominant
autosomal recessive
```
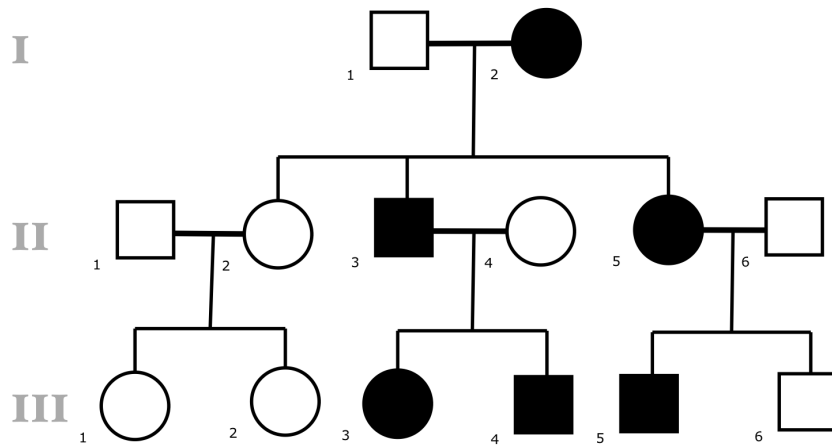
Pedigree 13

```
>>> a = pedigree(1,0,[1],[0,1])
>>> b = pedigree(1,0,[0],[0])
>>> c = pedigree(0,1,[1],[0,1])
>>> d = pedigree(0,0,[1],[0])
>>> e = pedigree(0,1,[1],[0])
>>> pdsolve([a,b,c,d,e])
Mode of inheritance:
autosomal recessive
```

Pedigree 14

Pedigree 14:

```
>>> a = pedigree(0,1,[0,1],[1])
>>> b = pedigree(0,1,[0,0],[0])
>>> c = pedigree(1,1,[1,1],[1])
>>> pdsolve([a,b,c])
Mode of inheritance:
+autosomal recessive
+autosomal dominant
sex-linked (X) dominant
```

Pedigree 14 is from an autosomal recessive disorder.

Pedigree 15
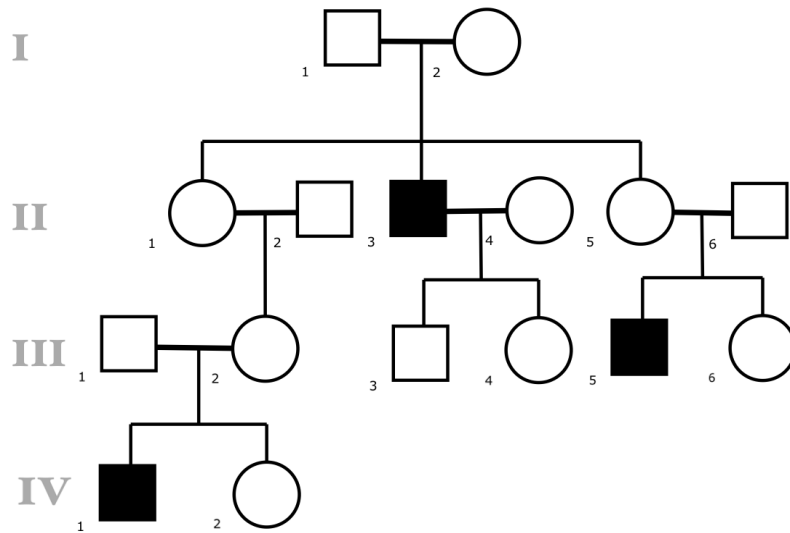
Pedigree 15:

```
>>> a = pedigree(0,0,[1],[0,0])
>>> b = pedigree(0,0,[],[0])
>>> c = pedigree(1,0,[0],[0])
>>> d = pedigree(0,0,[1],[0])
>>> e = pedigree(0,0,[1],[0])
>>> pdsolve([a,b,c,d,e])
Mode of inheritance:
++sex-linked (X) recessive
autosomal recessive
```
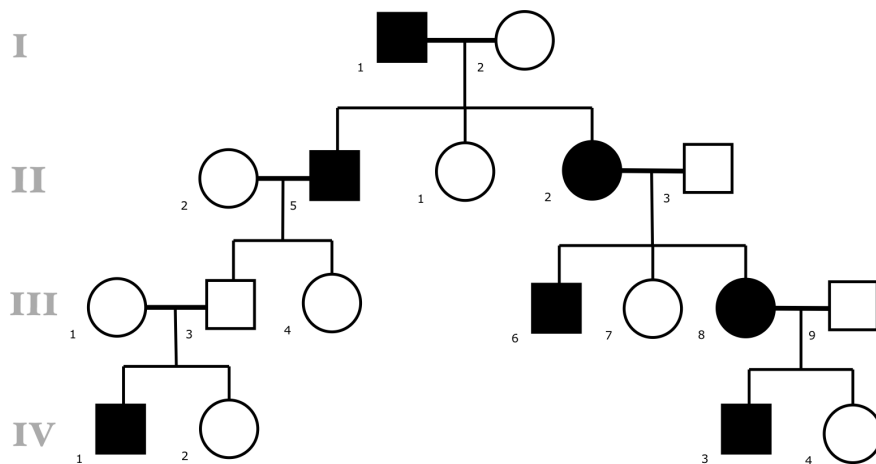
Pedigree 15 is from a sex-linked (X) recessive disorder.
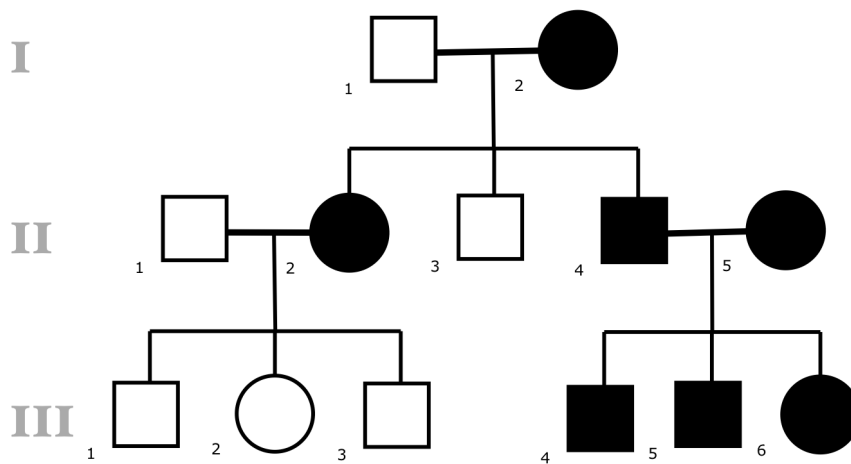
Pedigree 16:

```
>>> a = pedigree(0,0,[1],[0,0])
>>> b = pedigree(1,0,[0],[0])
>>> c = pedigree(0,0,[1],[0,0])
>>> d = pedigree(0,0,[0,1,1],[0])
>>> e = pedigree(1,0,[0],[0])
>>> f = pedigree(0,0,[1],[0])
>>> pdsolve([a,b,c,d,e,f])
Mode of inheritance:
++sex-linked (X) recessive
autosomal recessive
```

Pedigree 17.

We could solve the pedigree in the following manner, concentrating only on the generations V to VII:

```
>>> a = pedigree(1,0,[0],[0,0,0])
>>> b = pedigree(0,0,[0,1],[0])
>>> c = pedigree(0,0,[0,1,1],[0])
>>> pdsolve([a,b,c])
Mode of inheritance:
++sex-linked (X) recessive
autosomal recessive
```
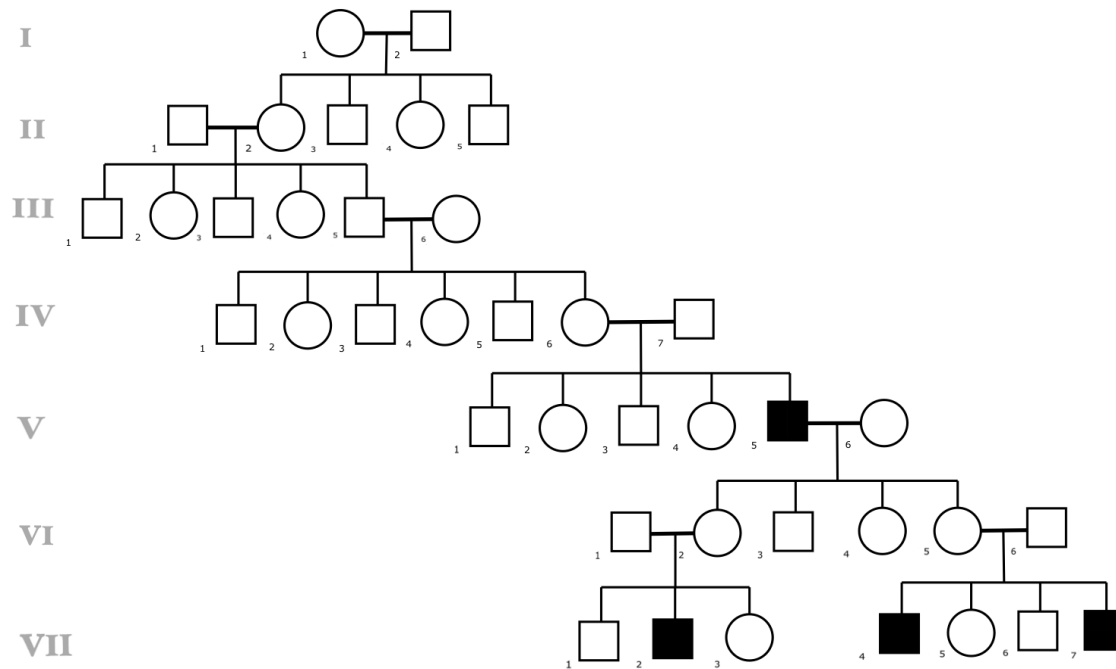
The disorder is most likely sex-linked (X) recessive, caused by *a loss-of-function mutation* in the X chromosome of the individual V5.
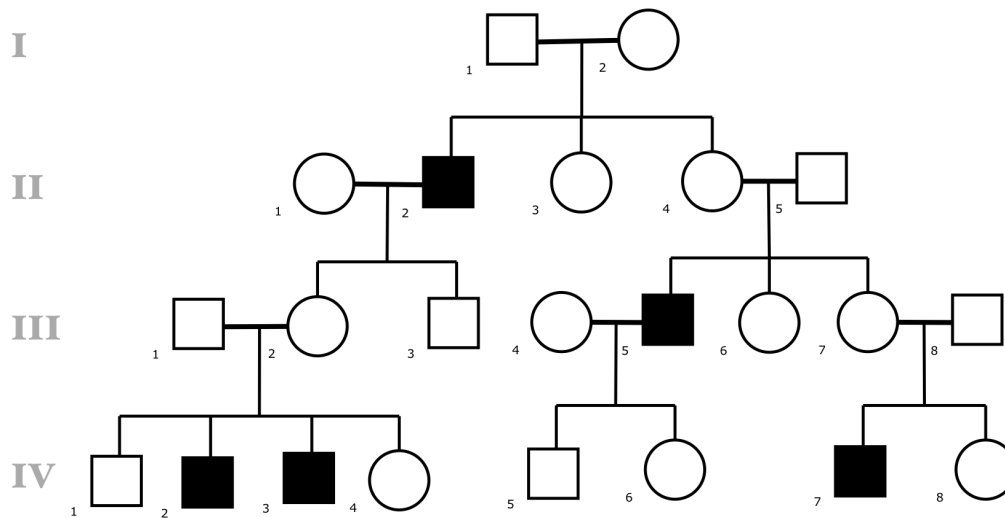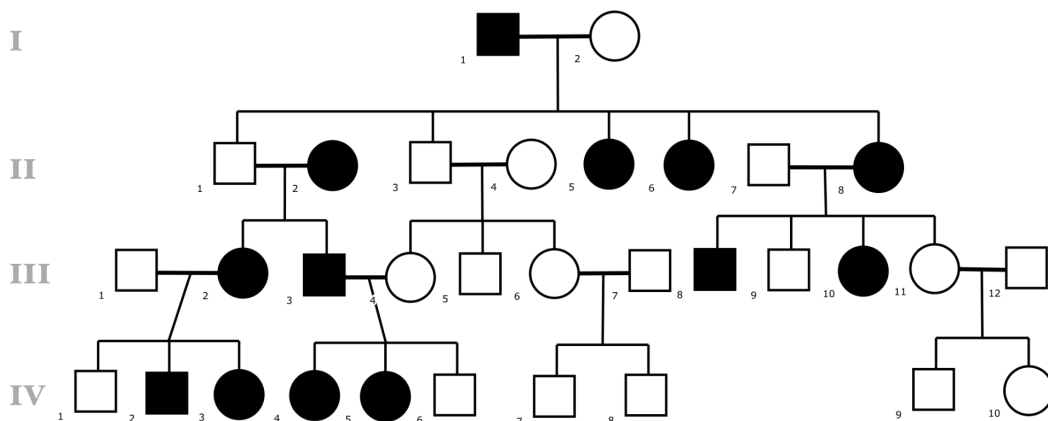
Pedigree 18:

```
>>> a = pedigree(1,0,[0,0],[1,1,1])
>>> b = pedigree(0,1,[1],[1])
>>> c = pedigree(0,0,[0],[0,0])
>>> d = pedigree(0,1,[1,0],[0,1])
>>> e = pedigree(0,1,[0,1],[1])
>>> f = pedigree(1,0,[0],[1,1])
```

```
>>> g = pedigree(0,0,[0,0],[])
>>> h = pedigree(0,0,[0],[0])
>>> pdsolve([a,b,c,d,e,f,g,h])
Mode of inheritance:
++sex-linked (X) dominant
+autosomal dominant
autosomal recessive
```

The disorder is in fact sex-linked (X) dominant. All of the girls of an affected man are affected.

Pedigree 19
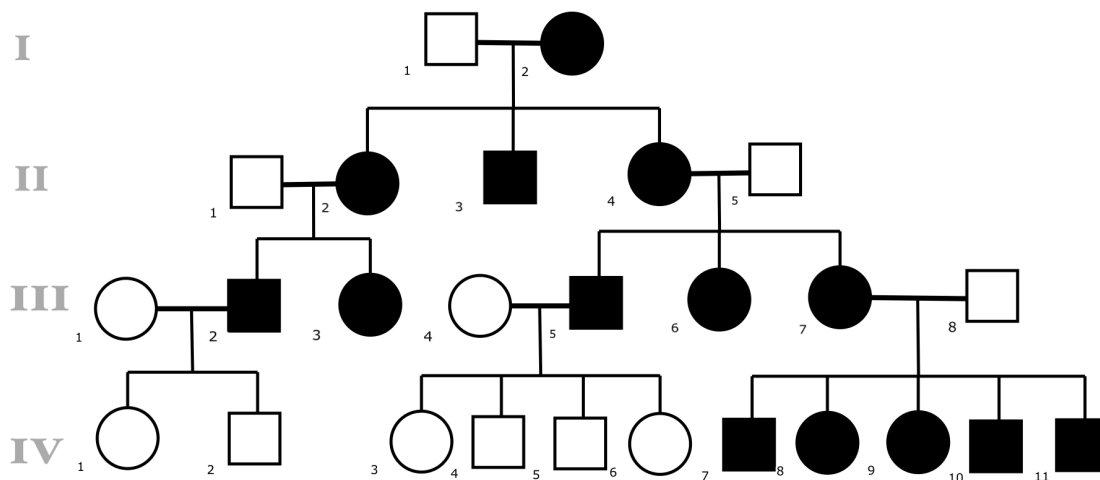
Pedigree 19:

```
>>> a = pedigree(0,1,[1],[1,1])
>>> b = pedigree(0,1,[1],[1])
>>> c = pedigree(0,1,[1],[1,1])
>>> d = pedigree(1,0,[0],[0])
>>> e = pedigree(1,0,[0,0],[0,0])
>>> f = pedigree(0,1,[1,1,1],[1,1])
>>> pdsolve([a,b,c,d,e,f])
Mode of inheritance:
++mtDNA
+autosomal dominant
autosomal recessive
```

Pedigree 20
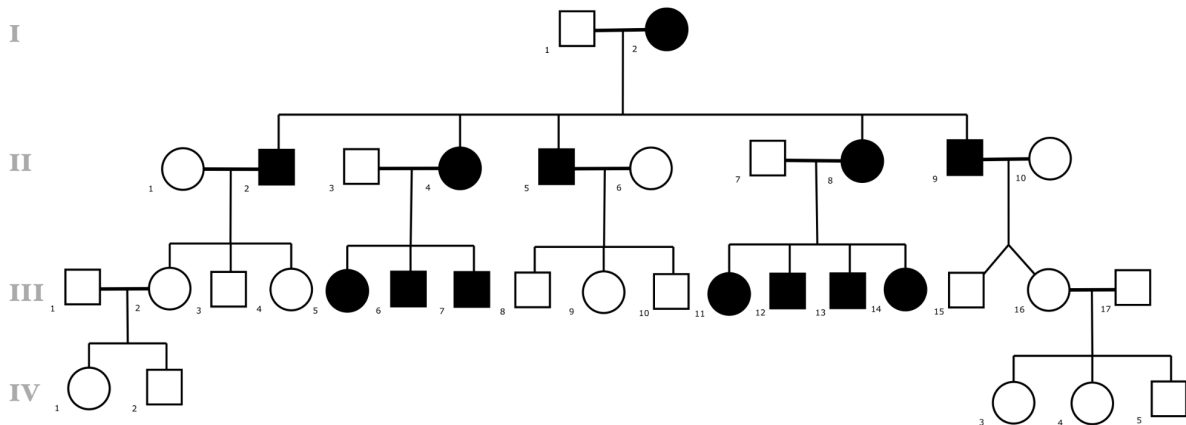
Pedigree 20:

```
>>> a = pedigree(0,1,[1,1,1],[1,1])
>>> b = pedigree(1,0,[0],[0,0])
>>> c = pedigree(0,1,[1,1],[1])
>>> d = pedigree(1,0,[0,0],[0])
>>> e = pedigree(0,1,[1,1],[1,1])
>>> f = pedigree(1,0,[0],[0])
>>> g = pedigree(0,0,[0],[0])
>>> h = pedigree(0,0,[0],[0,0])
>>> pdsolve([a,b,c,d,e,f,g,h])
Mode of inheritance:
++mtDNA
+autosomal dominant
autosomal recessive
```

The disorder is most likely inherited through maternal mtDNA. All of the children of an affected mother are affected, but none of the children of an affected father are affected.

Pedigree 21:

```
>>> a = pedigree(1,0,[1,1],[0,0])
>>> b = pedigree(1,0,[1],[0])
>>> c = pedigree(1,0,[1,1],[0])
>>> d = pedigree(1,0,[1,1],[0,0])
>>> e = pedigree(1,0,[1],[0])
>>> pdsolve([a,b,c,d,e])
Mode of inheritance:
++Y-chromosomal
+sex-linked (X) recessive
autosomal dominant
autosomal recessive
```

What if you are given two (or more) pedigrees that have the same mode of inheritance? Just proceed as before and use the *pdsolve()* function with the combined sub-pedigrees from the two large pedigrees.

Two pedigrees

I 1 2

II 1 2 3 4 5

III 1 2 3 4 5 6

IV 1 2 3 4

Pedigree 22A

I 3 4

II 6 7 8 9 10 11 12

III 7 8 9 10

IV 5 6 7 8

Pedigree 22B

Pedigree 22A:

```
>>> a = pedigree(0,1,[1,0],[1,0])
>>> b = pedigree(0,1,[1,0,0],[1,0])
>>> c = pedigree(0,0,[0],[0,0,0])
```

Pedigree 22B:

```
>>> d = pedigree(1,1,[1,1,1],[1,1,0,0])
>>> e = pedigree(1,1,[1,0],[1,0])
>>> f = pedigree(1,1,[1],[1,0,0])
```

Solve the super-pedigree with both pedigrees 22A and 22B:

```
>>> pdsolve([a,b,c,d,e,f])
Mode of inheritance:
autosomal dominant
```

You could have solved this problem also by solving just one sub-pedigree from the large pedigree 22B, in this case, the sub-pedigree in which the parents are II11 and II12:

```
>>> e.tbl(e.domrec())
autosomal dominant
```

Finally, we discuss some of the more convoluted cases and lastly a well-known historical example of pedigree analysis.

What if you have the same disorder but different modes of inheritance? This can be difficult to spot immediately. Let's solve one example from an entrance exam to a medical school (Pedigree 23 and 24). Arrows mark the probands. Observe how the different modes of inheritance affect the probands.
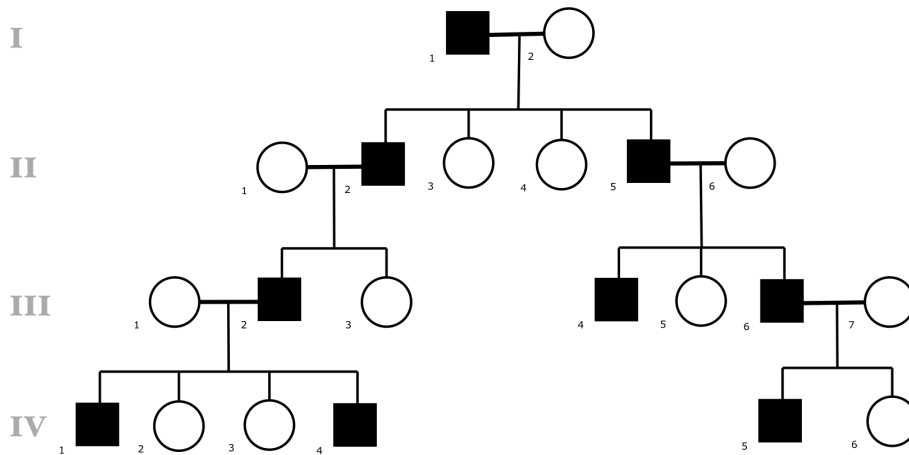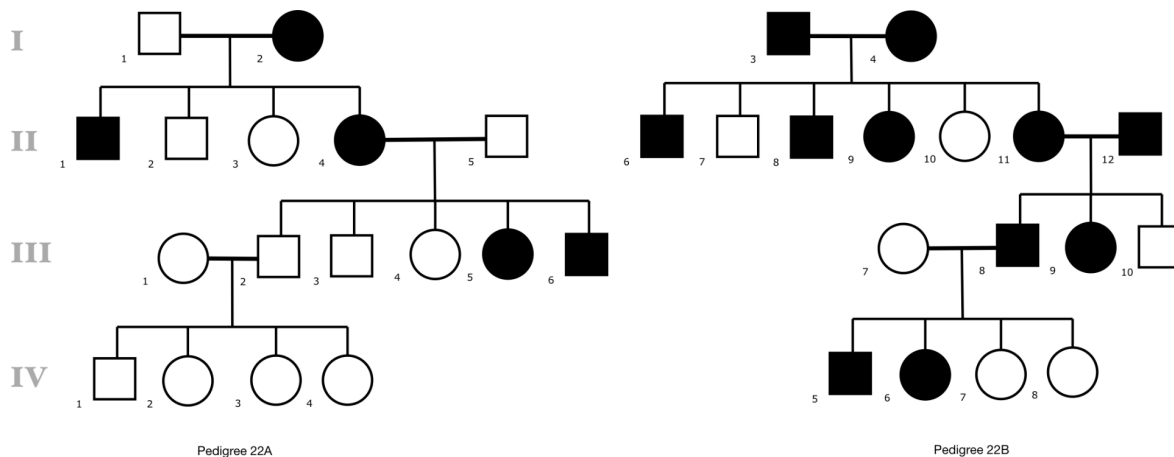
Pedigree 23
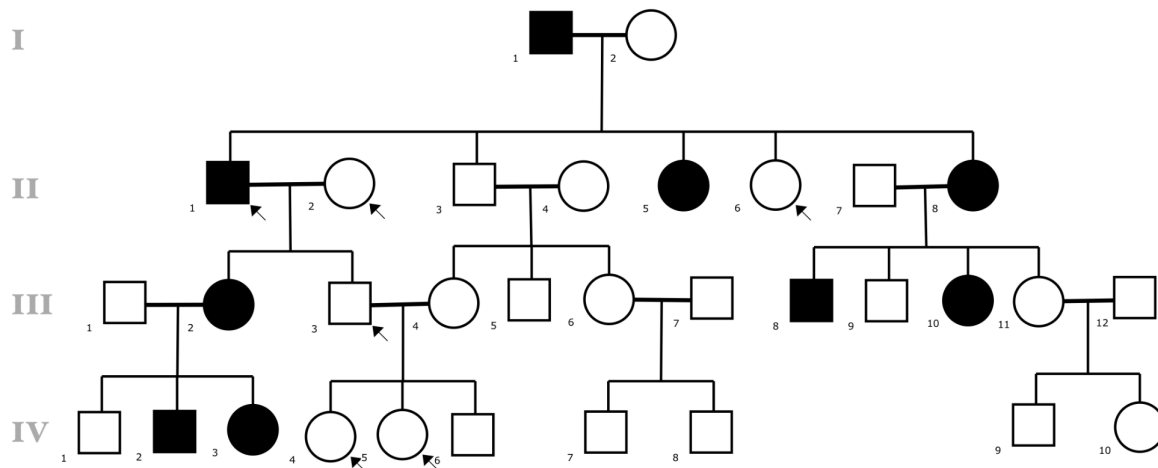
Pedigree 24
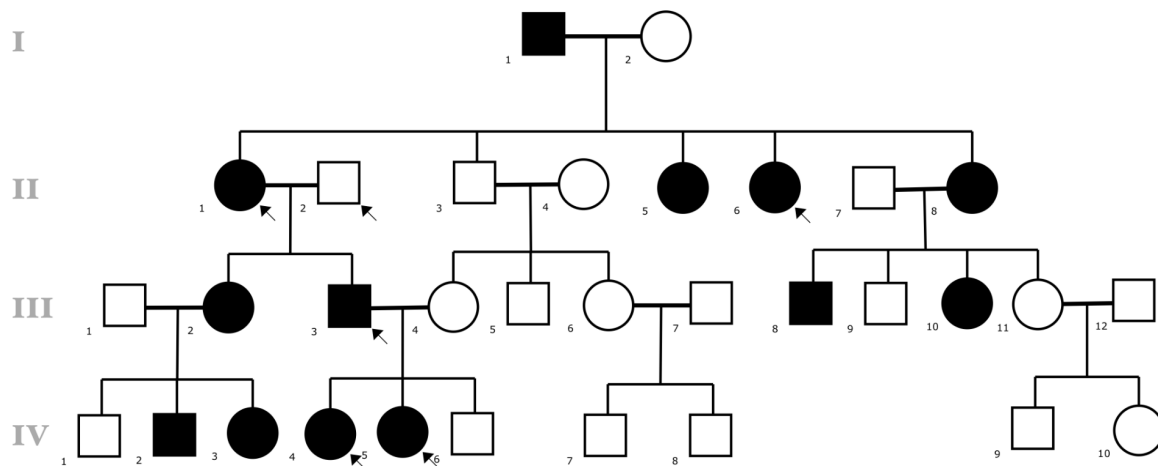
Pedigree 23:

```
>>> a = pedigree(1,0,[1,0],[1,1,0])
>>> b = pedigree(1,0,[0],[1])
>>> c = pedigree(0,0,[0],[0,0])
>>> d = pedigree(0,1,[1,0],[0,1])
>>> e = pedigree(0,1,[1,0],[1])
>>> f = pedigree(0,0,[0],[0,0])
>>> g = pedigree(0,0,[0,0],[])
```

```
>>> h = pedigree(0,0,[0],[0])
>>> pdsolve([a,b,c,d,e,f,g,h])
Mode of inheritance:
+autosomal dominant
autosomal recessive
```

```
>>> a = pedigree(1,0,[0],[1,1,1,1])
>>> b = pedigree(0,1,[1],[1])
>>> c = pedigree(0,0,[0],[0,0])
>>> d = pedigree(0,1,[1,0],[1])
>>> e = pedigree(0,1,[0,1],[1])
>>> f = pedigree(1,0,[0],[1,1])
>>> g = pedigree(0,0,[0,0],[])
>>> h = pedigree(0,0,[0],[0])
>>> pdsolve([a,b,c,d,e,f,g])
Mode of inheritance:
++sex-linked (X) dominant
+autosomal dominant
autosomal recessive
```

So, we have two different modes of inheritance (autosomal dominant and a very scarce sex-linked (X) dominant). Did you notice straight away that the modes of inheritance are different?

Next, we will discuss *incomplete penetrance* (or *variable penetrance*), which cannot currently be solved using this Python module (Pedigrees 25 and 26). First, look at the pedigree 25. We have a disorder that is known to be autosomal dominant. Only those individuals who are heterozygous are affected. Pedigree 25 is *the ideal case with complete penetrance*. The proband we are concerned with is the individual III3 (marked with an arrow), a female, and she is affected in pedigree 25. Let's solve the pedigree.

```
>>> a = pedigree(0,1,[0,1],[0,1])
>>> b = pedigree(1,0,[0],[1])
>>> c = pedigree(0,0,[0],[0])
>>> d = pedigree(0,0,[0],[0,0])
>>> e = pedigree(0,1,[0,0,1],[1])
>>> f = pedigree(0,0,[0,0,0],[0])
>>> g = pedigree(0,0,[0],[0,0])
>>> pdsolve([a,b,c,e,f,g])
```

```
Mode of inheritance:
++sex-linked (X) dominant
+autosomal dominant
autosomal recessive
```

Pedigree 25

As a solution, we find that the disorder is sex-linked (X) dominant and this is what would be suggested by the pedigree as more women are affected than men. We know from genetic analysis, however, that the disorder is autosomal dominant, and luckily, it is among the solutions.

Now, let's take a look at *the real-world pedigree with incomplete penetrance* (Pedigree 26). The proband (individual II3) is not affected, although she is heterozygous. Let's solve the pedigree. We expect a similar solution as before.
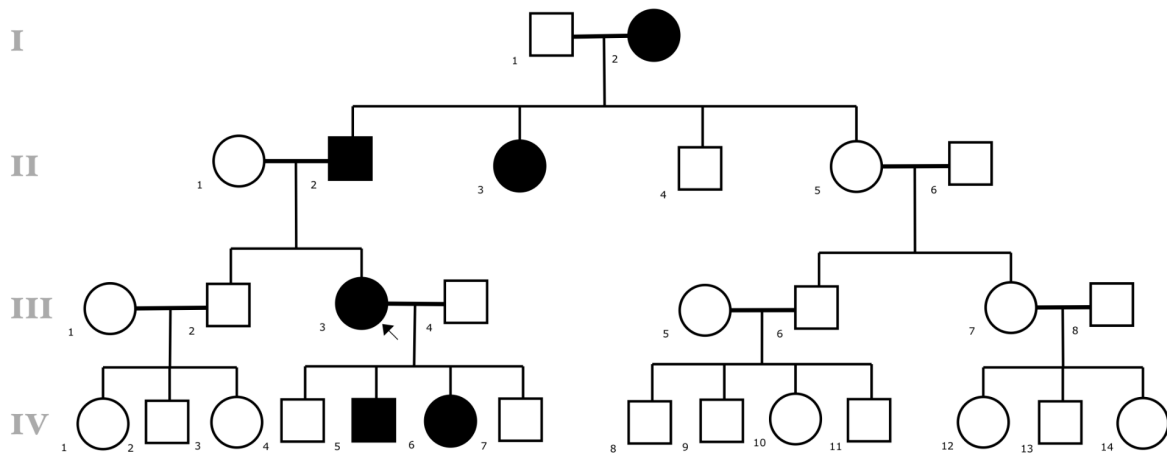
Pedigree 26:

```
>>> a = pedigree(0,1,[0,1],[0,1])
>>> b = pedigree(1,0,[0],[0])
>>> c = pedigree(0,0,[0],[0])
>>> d = pedigree(0,0,[0],[0,0])
>>> e = pedigree(0,0,[0,0,1],[1])
>>> f = pedigree(0,0,[0,0,0],[0])
>>> g = pedigree(0,0,[0],[0,0])
>>> pdsolve([a,b,c,e,f,g])
Mode of inheritance:
autosomal recessive
```

Pedigree 26

This time, we find out that the disorder is autosomal recessive, which is unfortunately incorrect as the disorder is autosomal dominant. Currently, the module can't handle incomplete penetrance.

Using the same procedure you can also solve highly unusual pedigrees with consanguineous marriages and a heavy degree of inbreeding (Pedigree 27, SB = stillbirth).

Pedigree 27:
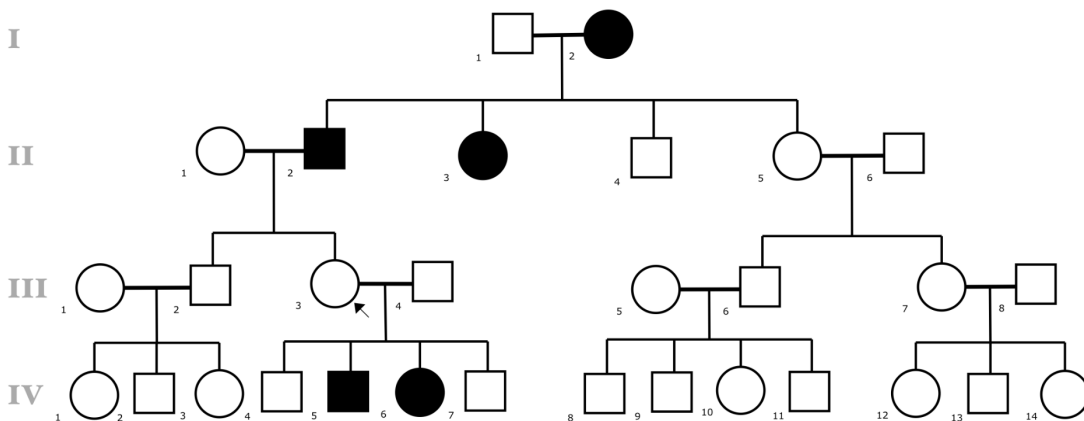
```
>>> a = pedigree(0,0,[0],[0])
>>> b = pedigree(0,0,[0],[0])
>>> c = pedigree(0,0,[0],[])
>>> d = pedigree(0,0,[],[0])
>>> e = pedigree(0,0,[0],[])
>>> f = pedigree(0,0,[1],[0,0,0])
>>> pdsolve([a,b,c,d,e,f])
Mode of inheritance:
++sex-linked (X) recessive
autosomal recessive
```

Pedigree 27
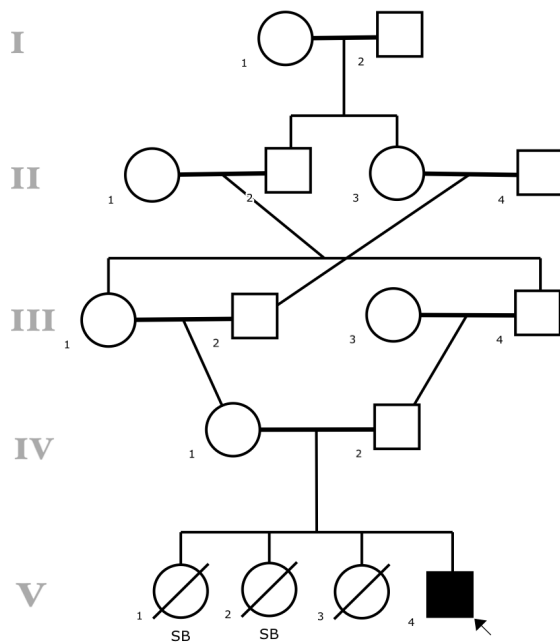
The disorder is sex-linked (X) recessive. It manifests due to heavy inbreeding. We can solve this problem more effortlessly:

```
>>> f.tbl()
+sex-linked (X) recessive (M het)
autosomal recessive (M&F het)
```

The result is very similar to the previous one. The disorder is more likely sex-linked (X) recessive than autosomal recessive.

Finally, let's discuss a classic historical case of pedigree analysis: *hemophilia in the Royal families of Germany, Spain, England, and Russia* [1] (see Fig below).

We could approach this problem from many different angles, but it seems to be quite straightforward. Using our Python module, you can only analyze observed phenotypes, and thus carrier status has no significance, and in our context, it is equal to the unaffected status. We can solve sub-pedigrees from the Prussian, Russian, and Spanish Royal Houses, and we should hopefully always reach the same conclusion. Notice that you don't have to enter the whole pedigree into *pdsolve()* to arrive at a solution.
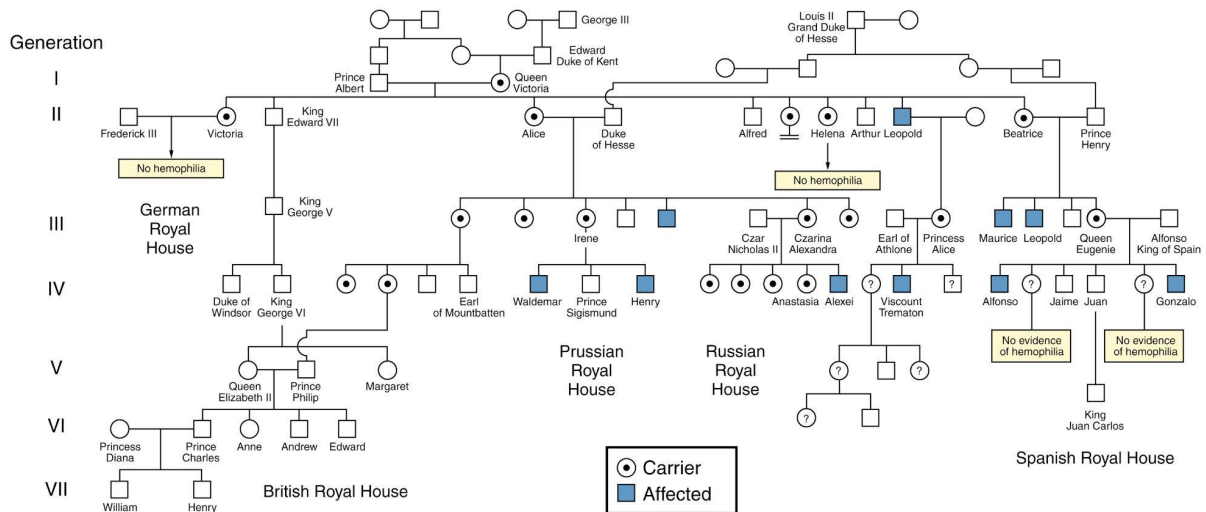
Fig. from [1].

Prussian Royal House:

```
>>> pedigree(0,0,[0,1,1],[]).tbl(pedigree(0,0,[0,1,1],[]).domrec())
+sex-linked (X) recessive (M het)
autosomal recessive (M&F het)
```

Russian Royal House:

```
>>> pedigree(1,0,[0],[0]).tbl(pedigree(0,0,[1],[0,0,0,0]).domrec())
autosomal recessive
sex-linked (X) recessive
```

Or:

```
>>>pedigree(0,0,[1],[0,0,0,0]).tbl(pedigree(0,0,[1],[0,0,0,0]).domr
ec())
+sex-linked (X) recessive (M het)
autosomal recessive (M&F het)
```

Spanish Royal House:

```
>>>pedigree(0,0,[0,1,1],[0]).tbl(pedigree(0,0,[1],[0,0,0,0]).domrec
())
+sex-linked (X) recessive (M het)
autosomal recessive (M&F het)
```

Or:

```
>>>
pedigree(0,0,[0,1,1],[0]).tbl(pedigree(0,0,[0,1,1],[0]).domrec())
+sex-linked (X) recessive (M het)
autosomal recessive (M&F het)
```

Or:

```
>>>
pedigree(0,0,[0,0,1,1],[0,0]).tbl(pedigree(0,0,[0,0,1,1],[0,0]).dom
rec())
+sex-linked (X) recessive (M het)
autosomal recessive (M&F het)
```

According to our in-depth analysis, the disorder seems to be sex-linked (X) recessive, and indeed it is. *Hemophilia B* is a sex-linked (X) recessive disorder that affects primarily men. It has plagued the Royal families of Europe, and Queen Victoria was the first known heterozygous carrier among these families.

Analyzing pedigrees

Consider the following pedigree (Pedigree 28). If the individuals III1 and III2 have children, will their children be healthy?

Pedigree 28
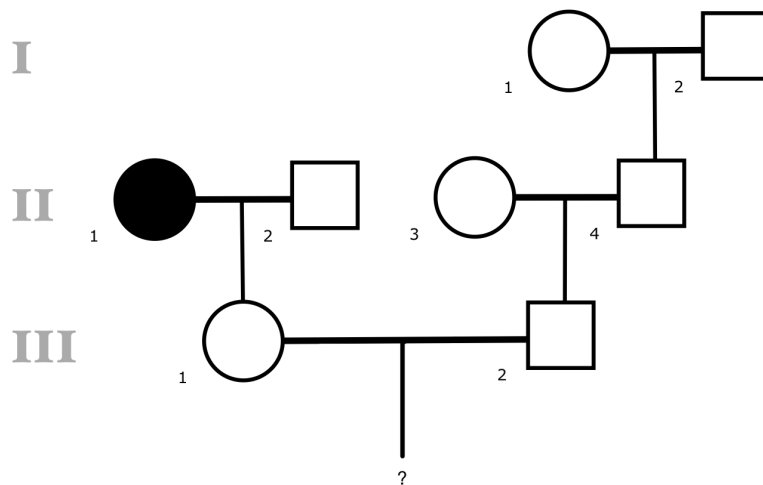
First, we must solve the pedigree.

```
>>> a = pedigree(0,0,[0],[])
>>> b = pedigree(0,1,[],[0])
>>> c = pedigree(0,0,[0],[])
>>> pdsolve([a,b,c])
Mode of inheritance:
autosomal dominant
sex-linked (X) dominant
```

The disorder is either autosomal or sex-linked (X) dominant. Now we can use the prob() function to determine the average probability of inheriting a disorder. The *prob()* function has the following syntax:

prob(mofi, father, mother, hz, sex, status)

Where:

mofi (= mode of inheritance) =
"ad" = autosomal dominant
"ar" = autosomal recessive

"sd" = sex-linked (X) dominant

"sr" = sex-linked (X) recessive

"mt"= mitochondrial DNA (maternal inheritance)

"y" = Y-chromosomal

father = 0, 0.5, or 1 (notice that *carrier status* can be indicated with 0.5)

mother = 0, 0.5, or 1 (notice that *carrier status* can be indicated with 0.5)

hz (= heterozygosity of the parents) =

""，

"F hom",

"M hom",

"F het",

"M het",

"M+F hom",

"F+M hom",

"M+F het",

"F+M het"

sex = "boy" or "girl"

status = 1 or 0

In this case, the mode of inheritance is *"ad"* or *"sld"*. The average probability that a boy would be affected can be estimated as follows:

```
>>> prob("ad",0,0,"","boy", 1)
Probability of a boy being affected:
autosomal dominant : 0
>>> prob("sld",0,0,"","boy", 1)
Probability of a boy being affected:
sex-linked (X) dominant : 0
```

Similarly for a girl:

```
>>> prob("ad",0,0,"", "girl", 1)
Probability of a girl being affected:
autosomal dominant : 0
>>> prob("sld",0,0,"", "girl", 1)
Probability of a girl being affected:
sex-linked (X) dominant : 0
```

In both cases the probability is zero, so the couple will have healthy children in every case. The pedigree originates from a family with *achondroplasia* which is an autosomal dominant disorder.

Let's move on to another example (Pedigree 29). Observing just the phenotypes (although we could figure out the genotypes and calculate the actual probabilities), what is the probability that a child born to the parents marked with arrows (parents III2 and III3) would be affected?

Let's solve the pedigree first.

```
>>> a = pedigree(1,0,[0,0],[0,0])
>>> b = pedigree(0,0,[1],[0])
>>> c = pedigree(0,1,[0,0],[])
>>> pdsolve([a,b,c])
Mode of inheritance:
autosomal recessive
```

Pedigree 29
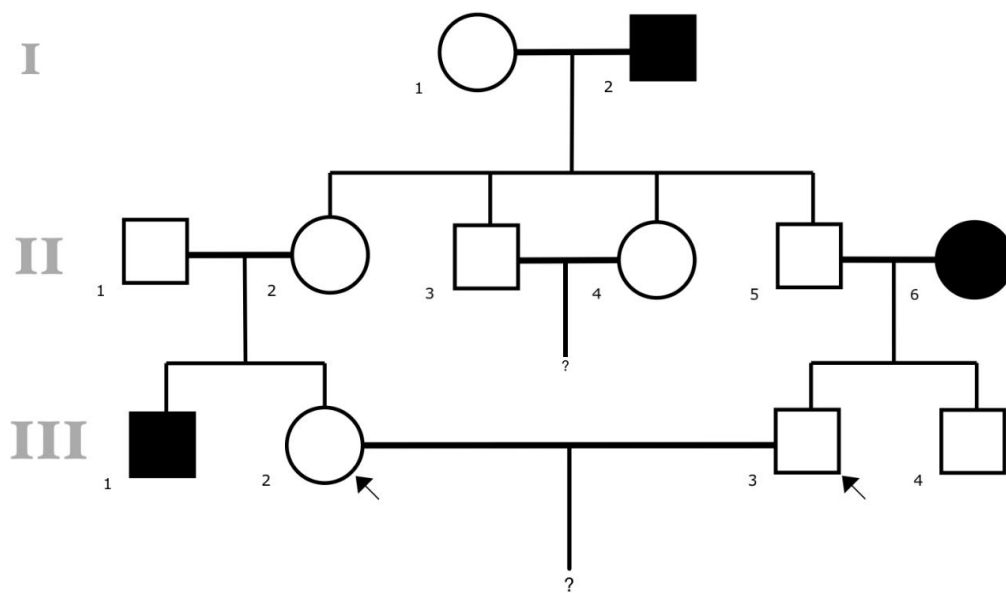
The disorder is autosomal recessive. What is the probability that a child born to the siblings II3 and II4 is affected?

```
>>> prob("ar",0,0,"M&F het", "boy", 1)
Probability of a boy being affected:
autosomal recessive : 0.25
```

As the brother and sister are both carriers the probability is the same as the average probability of being affected, that is 25%.

Next, we run into trouble, however. What is then the probability that a child born to the first cousins III2 and III3 is affected:

```
>>> prob("ar",0,0,"M&F het", "boy", 1)
Probability of a boy being affected:
autosomal recessive : 0.25
>>> prob("ar",0,0,"M hom", "boy", 1)
 Probability of a boy being affected:
autosomal recessive : 0
```

The probability seems to be 0.125 (12.5%) and that is the average probability of a child being affected by the disorder in this case. From the pedigree, we can conclude that both of the siblings (II2 and II5) are carriers, the probability that III2 is a carrier is ~67% (III3 has to be a carrier), and thus the actual probability calculated using genotypes is around 17%. The module cannot currently handle conditional probabilities in complex pedigrees using genotypes. In the first case, we were lucky and got the correct result (Pedigree 28).

The *prob()* function thus allows you to query the average tabulated probabilities of being affected concerning the different modes of inheritance.

Sometimes you know that the mother has for example an autosomal recessive disorder but you know nothing about the father's phenotype. Still, you want to know how the children will be affected by the illness. Then you have to check both cases, when the father is affected and when the father is not affected.

Here is an example of how a boy is affected by the father's status (with autosomal recessive disorders there is no difference between the sexes):

```
>>> prob("ar",1,1,"", "boy", 1)
Probability of a boy being affected:
autosomal recessive : 1.0
>>>  prob("ar",0,1,"", "boy", 1)
Probability of a boy being affected:
autosomal recessive : 0.25
```

If we don't know the father's phenotype the probability that a boy is affected is 0.625 (62.5%). There's more than a 50% chance of having a disorder.

You may want to know what is the probability of the next child being affected once you have solved a pedigree. If you have a simple pedigree you can find out the average probability of being affected using the pr() method with that pedigree:

pedigree(father, mother, [boys], [girls]).pr(dr, hz, sex, status)

Where:

dr =   "",
      "dominant",
      "recessive"
hz (= heterozygosity of the parents) =
      "",
      "F hom",
      "M hom",
      "F het",
      "M het",
      "M+F hom",
      "F+M hom",
      "M+F het",
      "F+M het"
sex = "boy" or "girl"
status = 1 or 0

We can analyze Pedigree 6 like so:

```
>>> pedigree(0,1,[1],[0,1]).pr("", "", "boy", 1)
Probability of a boy being affected:
sex-linked (X) dominant: 0.75
autosomal dominant: 0.75
autosomal recessive : 0.25
>>> pedigree(0,1,[1],[0,1]).pr("", "", "girl", 0)
Probability of a girl being healthy:
sex-linked (X) dominant: 0.25
autosomal dominant: 0.25
autosomal recessive : 0.75
```

Here are a few quick analyses from Pedigree 9:

```
>>> pedigree(0,0,[1],[0,0,0,0]).pr("", "", "boy", 0)
Probability of a boy being healthy:
```

```
sex-linked (X) recessive: 0.75
autosomal recessive : 0.9375
>>> pedigree(0,0,[1],[0,0,0,0]).pr("", "", "girl", 1)
Probability of a girl being affected:
sex-linked (X) recessive : 0
autosomal recessive : 0.0625
```

Consider finally this rather uncomplicated pedigree (Pedigree 30):

Pedigree 30
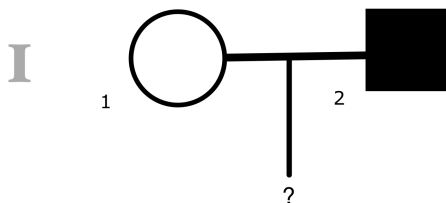
What is the probability that a boy born to parents I1 and I2 is affected?

```
>>> pedigree(1,0,[],[]).pr("","", "boy",1)
Probability of a boy being affected:
autosomal dominant: 0.75
autosomal recessive : 0.25
sex-linked (X) dominant : 0
sex-linked (X) recessive: 0.25
Y-chromosomal: 1.0
```

What if the mother is homozygous regarding the disorder alleles?

```
>>> pedigree(1,0,[],[]).pr("", "M hom", "boy",1)
Probability of a boy being affected:
autosomal dominant: 0.75
autosomal recessive : 0
sex-linked (X) dominant : 0
sex-linked (X) recessive : 0
Y-chromosomal: N/A
```

Short user guide

Create a new pedigree using the pedigree class.

Use the following notation:

0 = unaffected (i.e., healthy)
1 = affected

Then, create a pedigree with

a = pedigree(father, mother, [boys], [girls])

Where:

father = 1 or 0 (father must be defined and it cannot be empty)
mother = 1 or 0 (mother must be defined and it cannot be empty)
boys = [a list of ones and/or zeros] (use [] for missing variable)
girls = [a list of ones and/or zeros] (use [] for missing variable)

Examples:

```
>>> a = pedigree(1,0,[0,0],[1,1])
>>> a = pedigree(0,0,[],[0,1])
```

Solve a simple pedigree using *the tbl() method.*

If you don't know if the pedigree is dominant or recessive, use the default argument with the tbl() method.

pedigree(father, mother, [boys], [girls]).tbl()

Examples:

```
>>> pedigree(1,0,[0,1],[0]).tbl()
>>> pedigree(1,0,[1,1],[]).tbl()
```

This method will return a table of all the possible solutions. The following notation is used when outputting results:

M = mother
F = father
M+F = mother and father
het = heterozygous
hom = homozygous
+(+) = more likely
-(-) = less likely

Example:

```
>>> pedigree(1,0,[1,1],[]).tbl()
autosomal dominant
autosomal recessive (M het)
sex-linked (X) recessive (M het)
-Y-chromosomal
```

If you know for sure that the pedigree is dominant or recessive, use *"dominant"* or *"recessive"* with the tbl() method.

pedigree(father, mother, [boys], [girls]).tbl("dominant")
pedigree(father, mother, [boys], [girls]).tbl("recessive")

Examples:

```
>>> pedigree(0,0,[0,1,1],[0]).tbl("recessive")
+sex-linked (X) recessive (M het)
autosomal recessive (M&F het)
>>> pedigree(1,0,[0],[0]).tbl("dominant")
autosomal dominant
```

You can also use the domrec() method if it is applicable. The *domrec()* method can be applied if the parent's phenotypes are the same and they differ from at least some of the children's phenotypes.

dr = pedigree(father$_{dr}$, mother$_{dr}$, [boys$_{dr}$], [girls$_{dr}$]).domrec()
pedigree(father, mother, [boys], [girls]).tbl(dr)

Examples:

```
>>> dr = pedigree(0,0,[1],[0,0,0,0]).domrec()
>>> pedigree(0,0,[0,1,1],[0]).tbl(dr)
+sex-linked (X) recessive (M het)
```

```
autosomal recessive (M&F het)
```

or

```
>>>
pedigree(0,0,[0,1,1],[0]).tbl(pedigree(0,0,[1],[0,0,0,0]).domrec())
+sex-linked (X) recessive (M het)
autosomal recessive (M&F het)
```

Solve a more complex pedigree.

If there are two or more sub pedigrees (simple pedigrees) you can solve the whole pedigree by combining the results of the simple pedigrees. First, you must create simple pedigrees.

> **a_1 = pedigree(father$_1$, mother$_1$, [boys$_1$], [girls$_1$])**
>
> **…**
>
> **a_n = pedigree(father$_n$, mother$_n$, [boys$_n$], [girls$_n$])**

Then you have to use the pdsolve() function to solve the whole pedigree. The first argument should be a list of the simple pedigrees (the default argument of the output format is a table).

> **pdsolve([a_1,…,a_n])**

Examples:
```
>>> a = pedigree(0,1,[0,1],[0,0])
>>> b = pedigree(0,0,[0,0],[0])
>>> c = pedigree(1,0,[0],[0])
>>> d = pedigree(0,0,[0,0],[0])
>>> e = pedigree(0,0,[1,1],[])
>>> pdsolve([a,b,c,d,e])
Mode of inheritance:
+autosomal recessive
>>>
>>> a = pedigree(1,1,[1,0,0],[1])
>>> b = pedigree(1,0,[1],[0,0])
>>> pdsolve([a,b])
Mode of inheritance:
autosomal dominant
```

Analyzing pedigrees.

You can analyze the average probabilities of being affected associated with simple pedigrees using the pr() method.

pedigree(father, mother, [boys], [girls]).pr(dr, hz, sex, status)

Where:

dr =   "",
           "dominant",
           "recessive"
hz (= heterozygosity of the parents) =
           "",
           "F hom",
           "M hom",
           "F het", "
            M het",
           "M+F hom",
           "F+M hom",
           "M+F het",
           "F+M het"
sex = "boy" or "girl"
status = 1 or 0

Examples:

```
>>> pedigree(0,1,[],[]).pr("recessive","","boy", 1)
Probability of a boy being affected:
sex-linked (X) recessive: 1.0
autosomal recessive : 0.25
>>>
>>> pedigree(0,1,[0,0],[]).pr("","","boy",0)
Probability of a boy being healthy:
autosomal dominant: 0.25
sex-linked (X) dominant: 0.25
autosomal recessive : 0.75
```

You can also query the average probabilities related to a mode of inheritance using the prob() function without creating a pedigree.

prob(mofi, father, mother, hz, sex, status)

Where:

mofi (= mode of inheritance) =

> "ad" = autosomal dominant
>
> "ar" = autosomal recessive
>
> "sd" = sex-linked (X) dominant
>
> "sr" = sex-linked (X) recessive
>
> "mt"= mitochondrial DNA (maternal inheritance)
>
> "y" = Y-chromosomal

father = 0, 0.5, or 1 (notice that carrier status can be indicated with 0.5)

mother = 0, 0.5, or 1 (notice that carrier status can be indicated with 0.5)

hz = (heterozygosity of the parents) =

> "",
>
> "F hom",
>
> "M hom",
>
> "F het",
>
> "M het",
>
> "M+F hom",
>
> "F+M hom",
>
> "M+F het",
>
> "F+M het"

sex = "boy" or "girl"

status = 1 or 0

Examples:

```
>>> prob("ad",1,0,"","boy",1)
Probability of a boy being affected:
autosomal dominant: 0.75
>>>
>>> prob("ad",1,0,"","girl",1)
Probability of a girl being affected:
autosomal dominant: 0.75
```

Afterword

Unfortunately, I have had insufficient data to debug the whole module. Inaccuracies may remain in the code: the errors and exceptions may not be handled with due diligence, and in some cases, the output may be missing information. The weights in the results are empirical and would need to be checked against a database. Especially 'Analyzing pedigrees' section is still under construction. The probability estimators should also take into account the genotype and not just offer average probabilities. The code includes some redundancy so that the algorithm reaches the correct conclusion. It is also easier to debug, you can try it for yourself. Caveat emptor.

Notation

HH = handheld
Student software = TI-Nspire CX CAS Student Software

boys = [a list of ones and/or zeros] (use [] for missing variable)
      e.g. [0,0,1,1] or [0]
girls = [a list of ones and/or zeros] (use [] for missing variable)
      e.g. [0,1,1] or []
dr = "dominant" or "recessive"
father = 1 or 0 (father must always be defined; the father can be 0.5 with the prob() function)
hz (= heterozygosity of the parents) =
      "",
      "F hom",
      "M hom",
      "F het",
      "M het",
      "M+F hom",
      "F+M hom",
      "M+F het",
      "F+M het"
mofi (= mode of inheritance) =
      "ad" = autosomal dominant
      "ar" = autosomal recessive
      "sd" = sex-linked (X) dominant
      "sr" = sex-linked (X) recessive
      "mt"= mitochondrial DNA (maternal inheritance)
      "y" = Y-linked
mother = 1 or 0 (mother must always be defined; the mother can be 0.5 with prob())
sex = "boy" or "girl"
status = 1 or 0

pedigree(father, mother, [boys], [girls])
pedigree(father$_{dr}$, mother$_{dr}$, [boys$_{dr}$], [girls$_{dr}$]).domrec()
pedigree(father,mother,[boys],[girls]).tbl(pedigree(father$_{dr}$, mother$_{dr}$, [boys$_{dr}$], [girls$_{dr}$]).domrec())
pedigree(father, mother, [boys], [girls]).pr(dr, hz, sex, status)
HH: **pedigree(father, mother, [boys], [girls]).tbl()**
Student software: pedigree(father, mother, [boys], [girls]).tbl(dr)
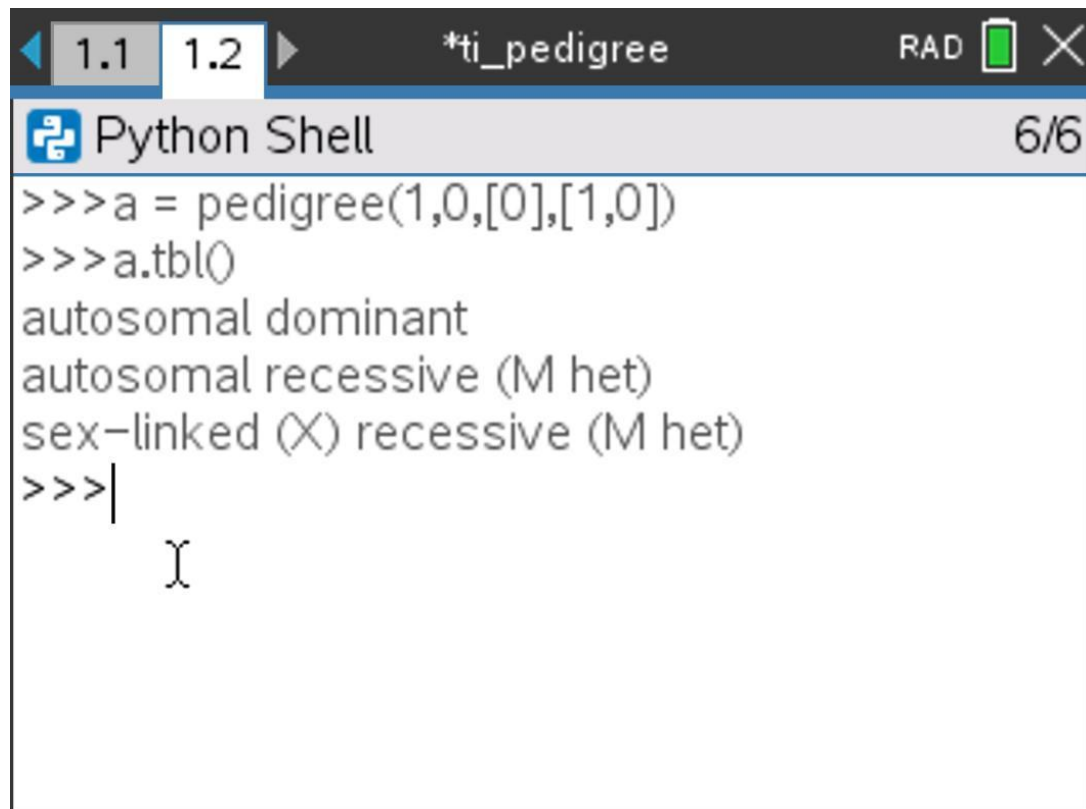
HH: **pdsolve([n$_1$, …, n$_k$])**
OR: pdsolve([n$_1$, …, n$_k$], "tbl")
prob(mofi, father, mother, hz, sex, status)

References

[1] Lynn B. Jorde et al., Medical Genetics, Elsevier, 6th ed., 2020
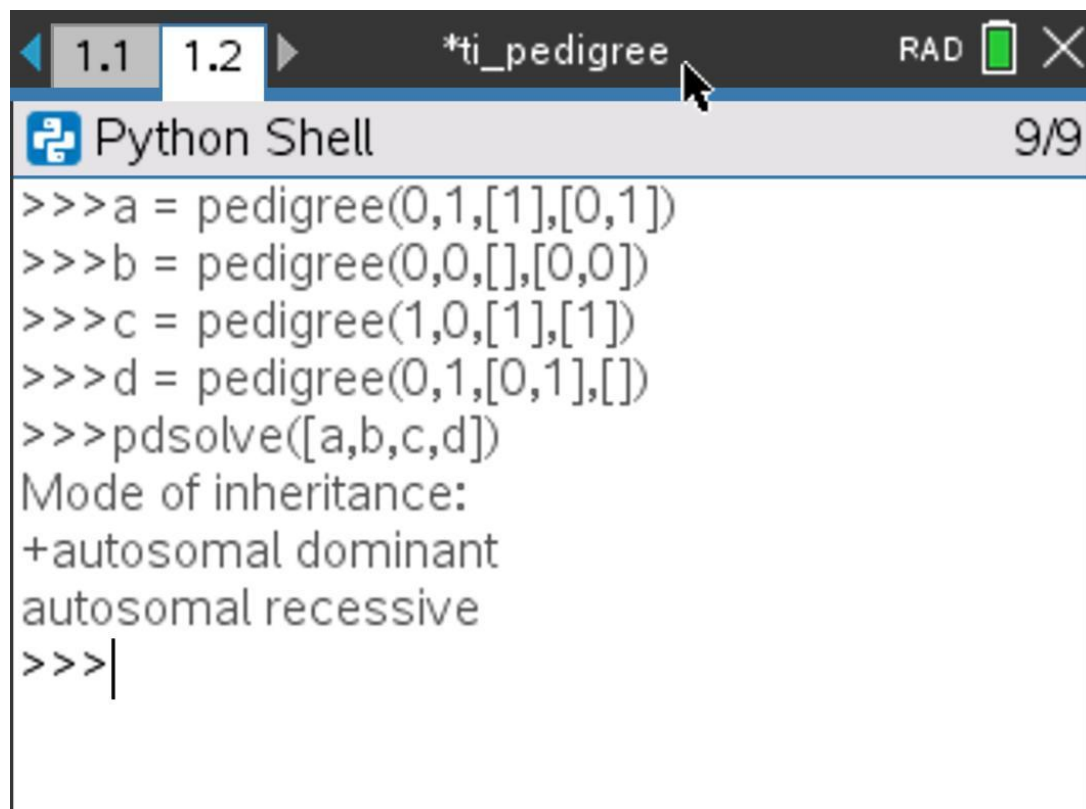[2] OMIM - Online Mendelian Inheritance in Man, https://omim.org

Screenshots

🐍 Python Shell   6/6

```
>>>a = pedigree(1,0,[0],[1,0])
>>>a.tbl()
autosomal dominant
autosomal recessive (M het)
sex−linked (X) recessive (M het)
>>>|
```

🐍 Python Shell   9/9

```
>>>a = pedigree(0,1,[1],[0,1])
>>>b = pedigree(0,0,[],[0,0])
>>>c = pedigree(1,0,[1],[1])
>>>d = pedigree(0,1,[0,1],[])
>>>pdsolve([a,b,c,d])
Mode of inheritance:
+autosomal dominant
autosomal recessive
>>>|
```

```
>>>a = pedigree(1,0,[1,1],[0,0])
>>>b = pedigree(1,0,[1],[0])
>>>c = pedigree(1,0,[1,1],[0])
>>>d = pedigree(1,0,[1,1],[0,0])
>>>e = pedigree(1,0,[1],[0])
>>>pdsolve([a,b,c,d,e])
Mode of inheritance:
++Y−chromosomal
+sex−linked (X) recessive
autosomal dominant
autosomal recessive
```

```
>>>dr = pedigree(0,0,[1],[0,0,0,0]).domrec()
>>>pedigree(0,0,[0,1,1],[0]).tbl(dr)
+sex−linked (X) recessive (M het)
autosomal recessive (M&F het)
>>>pedigree(0,1,[1],[0,1]).pr("","","boy",1)
Probability of a boy being affected:
sex−linked (X) dominant : 0.75
autosomal dominant : 0.75
autosomal recessive : 0.25
>>>
```

🐍 Python Shell   9/9

```
>>>prob("sld",0,0,"","girl",1)
Probability of a girl being affected:
sex-linked (X) dominant : 0
>>>pedigree(0,1,[1,1],[0,1,1]).pr("","","boy",1)
Probability of a boy being affected:
sex-linked (X) dominant : 0.75
autosomal dominant : 0.75
autosomal recessive : 0.25
>>>
```

🐍 Python Shell   8/8

```
>>>pedigree(0,0,[0,1,1],[0]).tbl("recessive")
+sex-linked (X) recessive (M het)
autosomal recessive (M&F het)
>>>pedigree(1,0,[0],[0]).tbl("dominant")
autosomal dominant
>>>pedigree(0,0,[1],[]).domrec()
'recessive'
>>>
```