

Visualizing Double-Entry Bookkeeping: A Digital History Methodology

Jesse Sadler
Virginia Tech

@vivalosburros
jessesadler.com
github.com/jessesadler

Slides: jessesadler.com/slides/chico2021.pdf



DANIEL VANDER
MEULEN.
A. 1583.



HESTER DELLA
FAILLE.
A. 1583.



Solving niche historical problems with Digital Humanities: Non-decimal currencies

- Finding my footing in Digital Humanities
- debkeepr: Analysis of Non-Decimal Currencies in R
- Example: The estate of Jan della Faille de Oude (1515–1582)

Finding my footing in Digital Humanities



Find a project



Power of coding



Just pick a project

How to draw an owl

1.



2.



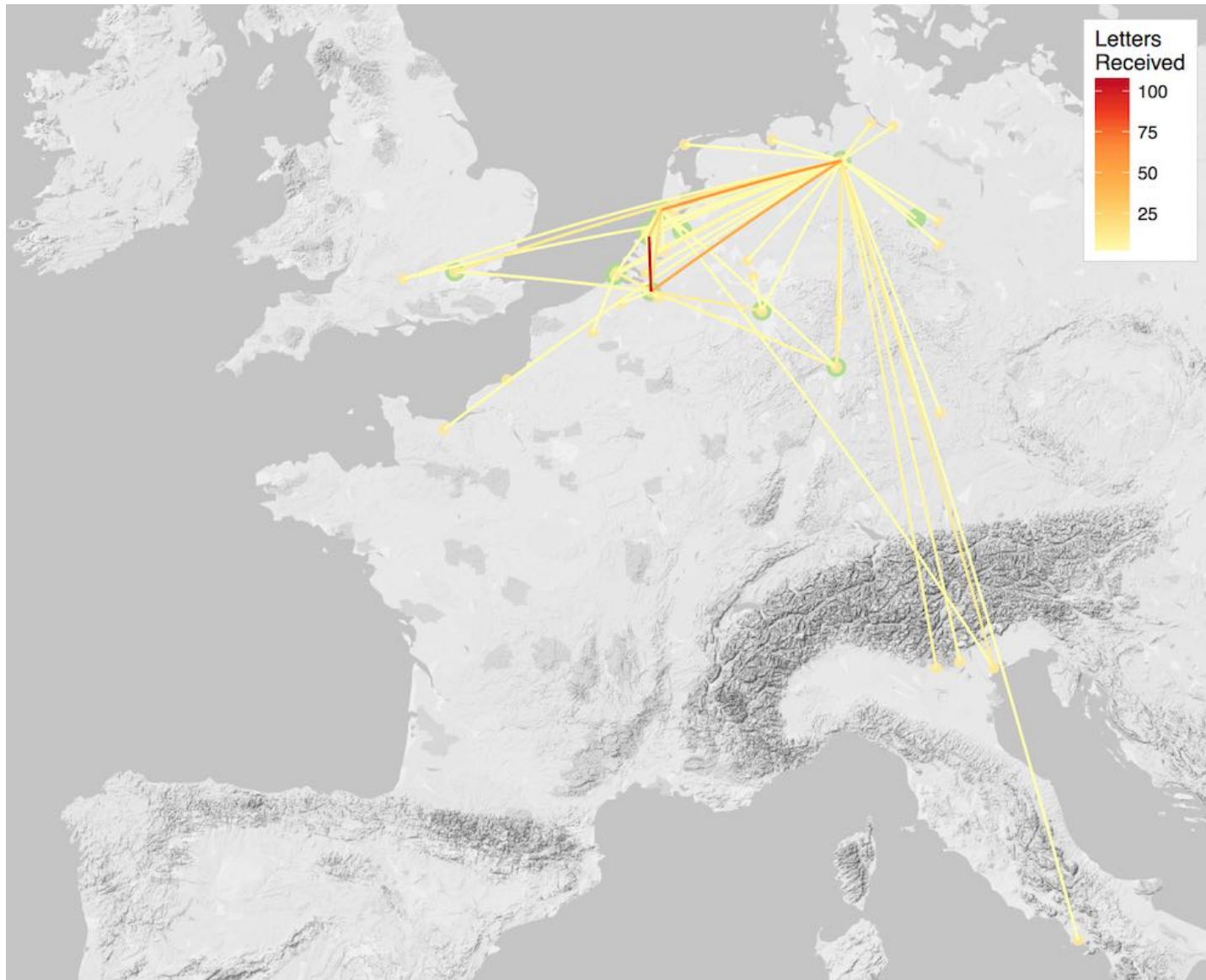
1. Draw some circles

2. Draw the rest of the fucking owl



6,000 letters sent
to Daniel van der
Meulen between
1578 and 1600

Letters received by Daniel van der Meulen, 1578–1591



[jessesadler.com/
project/dvdm-
correspondence/](http://jessesadler.com/project/dvdm-correspondence/)

Wickham and Grolemund, R for Data Science

R for Data Science

Table of contents

Welcome

1 Introduction

Explore

2 Introduction

3 Data visualisation

4 Workflow: basics

5 Data transformation

6 Workflow: scripts

7 Exploratory Data Analysis

8 Workflow: projects

Wrangle

9 Introduction

10 Tibbles

11 Data import

12 Tidy data

13 Relational data

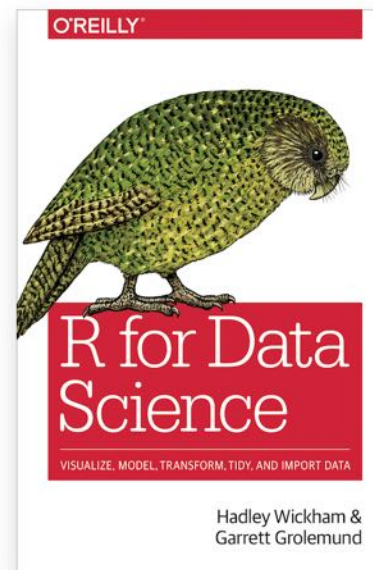
14 Strings

15 Factors

Welcome

This is the website for “**R for Data Science**”. This book will teach you how to do data science with R: You’ll learn how to get your data into R, get it into the most useful structure, transform it, visualise it and model it. In this book, you will find a practicum of skills for data science. Just as a chemist learns how to clean test tubes and stock a lab, you’ll learn how to clean data and draw plots—and many other things besides. These are the skills that allow data science to happen, and here you will find the best practices for doing each of these things with R. You’ll learn how to use the grammar of graphics, literate programming, and reproducible research to save time. You’ll also learn how to manage cognitive resources to facilitate discoveries when wrangling, visualising, and exploring data.

This website is (and will always be) **free to use**, and is licensed under the **Creative Commons Attribution-NonCommercial-NoDerivs 3.0 License**. If you’d like a **physical copy** of the book, you can order it from **amazon**; it was published by O’Reilly in January 2017. If you’d like to **give back** please make a donation to **Kākāpō Recovery**: the **kākāpō** (which appears on the cover of R4DS) is a critically endangered native NZ



[https://
r4ds.had.co.nz](https://r4ds.had.co.nz)

Jesse Sadler

A blog about early modern history and digital humanities

Introducing debkeepr

An R package for the analysis of non-decimal currencies

Posted on September 18, 2018

After an extensive period of iteration and a long but rewarding process of learning about package development, I am pleased to announce the release of my first R package. The package is called **debkeepr**, and it derives directly from my historical **research on early modern merchants**. **debkeepr** provides an interface for working with non-decimal currencies that use the tripartite system of pounds, shillings, and pence that was used throughout Europe in the medieval and early modern periods. The package includes functions to apply arithmetic and financial operations to single or multiple values and to analyze account books that use **double-entry bookkeeping** with the latter providing the basis for the name of **debkeepr**. In a later post I plan to write about the package development process, but here I want to discuss the motivation behind the creation of the package and provide some

My approach to Digital Humanities

- Develop transferable skills
- Build small-scale projects
- Use open-source tools that facilitate reproducible research
- Use skills to solve niche problems

Champagne fairs



Network of Florentine banks






Renaissance Italy



Contributing to data science

- Approaching learning digital skills from a humanistic perspective
- Blogging: Learning in public
- Becoming a part of the open-source community

opencage: Open-source geocoding

 **opencage v0.2.2.9000**

Search...

 [Get started](#) [Reference](#) [Articles ▾](#) [Changelog](#) 

rOpenSci: The *opencage* package

Geocode with the [OpenCage](#) API, either from place name to longitude and latitude (forward geocoding) or from longitude and latitude to the name and address of the location (reverse geocoding).

Installation

Install the package with:

```
install.packages("opencage")
```

Or install the development version using [remotes](#) with:


```
remotes::install_github("ropensci/opencage")
```

Quickstart

For the best experience, we recommend that you read through the "Introduction to opencage" vignette (`vignette("opencage")`), but if you are in a hurry:

1. Register at opencagedata.com/users/sign_up.
2. Generate an API key at the [OpenCage dashboard](#).
3. Save your API key as an [environment variable](#) like `OPENCAGE_KEY=yourkey` in `.Renviron`. See `help(oc_config)` for alternative ways to set your OpenCage API key.

Now you are ready to turn place names into latitude and longitude coordinates:



Links

Download from CRAN at <https://cloud.r-project.org/package=opencage>

Browse source code at <https://github.com/ropensci/opencage/>

Report a bug at <https://github.com/ropensci/opencage/issues>

License

GPL (>= 2)

Developers

Daniel Posenriede
Author, maintainer 

Jesse Sadler
Author 

Maëlle Salmon
Author 

[All authors...](#)

Dev status

CRAN

0.2.2

CRAN

NOTE

<https://docs.ropensci.org/opencage/>

debkeepr: Analysis of Non-Decimal Currencies in R

debkeepr 0.0.5.9000



Reference

Get started

Articles ▾

News



debkeepr: Analysis of Non-Decimal Currencies

`debkeepr` integrates non-decimal currencies that use the tripartite system of pounds, shillings, and pence into the methodologies of Digital Humanities and the practices of reproducible research. The package makes it possible for historical non-decimal currencies to behave like decimalized numeric values through the implementation of the `deb_lsd` and `deb_decimal` vector classes or types. These types are based on the infrastructure provided by the [vctrs package](#). `debkeepr` simplifies the process of performing arithmetic calculations with non-decimal currencies — such as adding £3 13s. 4d. sterling to £8 15s. 9d. sterling — and also provides a basis for analyzing account books with thousands of transactions recorded in non-decimal currencies. The name of the `debkeepr` package derives from this latter capability of analyzing historical account books that often used [double-entry bookkeeping](#).

Installation

You can install `debkeepr` from GitHub with [remotes](#):

```
# install.packages("remotes")
remotes::install_github("jessesadler/debkeepr")
```

Please open an [issue](#) if you have any questions, comments, or requests.

Historical Background

The `debkeepr` package uses the nomenclature of [l](#), [s](#), and [d](#) to represent pounds, shillings, and pence units in non-decimal currencies. The abbreviations derive from the Latin terms [libra](#), [solidus](#), and [denarius](#). The libra was a Roman measurement of weight, while the solidus and denarius were both Roman coins. The denarius was a silver coin from the era of the Republic, in contrast to the golden solidus that was issued in the Late Empire. As the production of silver coins overtook that of gold by the 8th century, a solidus came to represent 12 silver denarii coins, and 240 denarii were — for a time — made from one libra or pound of silver. The custom of

Links

Browse source code at

<https://github.com/jessesadler/debkeepr/>

Report a bug at

<https://github.com/jessesadler/debkeepr/issues>

License

[Full license](#)

MIT + file [LICENSE](#)

Developers

[Jesse Sadler](#)

Author, maintainer

Dev status

build

codecov 99%

lifecycle

<https://jessesadler.github.io/debkeepr/>

GitHub: <https://github.com/jessesadler/debkeepr/>

— + Anno 1585. —

Jacques della faille de jonghe. is — rediteur		Aa 15. februarij a. 1585.				
26	4	0.	voor onkosten van onse bruyloft voor soz veel lymaet a 10 p deen			
			aen welken monninx. betaet ende 19 p. voor vracst van des Goffers met			
			7. parckhuis rompt tjanng. — — — — —	26	4	—
—			Ditto 2 0 0. 0. voor de voorsp. voor scrijpvracst van 20. Septer deela faille			
			met diuifse parckhuis int vertuut van dit vaxp. — — — — —	0	—	—
—			Ditto 2 5 1 10. voor de voorsp. voor een siliure scale wettende 11 Oct 15. 15.			
			ady 20. Septer deela faille in october ouergelaten. — — — — —	5	1	10
—			Ditto 2 28 10. 0. voor de voorsp. voor 4. minen ranssen wijn voor de bruyloft			
			tot dort ydop rompt met de vaten. — — — — —	20	10	—
—			Ditto 2 4 7 0. voor de voorsp. voor diuers lachten ende ranssen ady 20.			
			Septer deela faille ouer lange ouergelaten. — — — — —	4	7	0
—			Ditto 2 3 69. 4. 4. voor de voorsp. voor een rekening van onkosten op de			
			bruyloft godaer als breeder ghescein Kerkme is blykende. — — — — —	3 69	4	4
20			Ditto 2 88 4. 11. voor de voorsp. voor een rekening van onkosten tot			
			besoef van de bruyloft by Jacques noiot in dit vaxp. betaet. — — — — —	0 5	4	11
—			Ditto 2 15 100. 10. voor de voorsp. voor soo veel ingan deela faille voor			
			de runderen vant bruyloft welck. int gach ende romen betaet. — — — — —	15	18	10
—			Ditto 2 2 18. 6. voor de voorsp. voor soo veel lichte timmerman voor			
			op den deelen ende ardyt betaet. — — — — —	2	18	6
—			Ditto 2 0. 3. 4. voor de voorsp. voor vracst ende licencie van een gheue			
			amw asijn romende van londen. — — — — —	—	3	4
10			Mart. 2 155. 1. 4. voor tarwe in grunde van Robert noiot in Simiclie			
			voordat ich inder 235. quarter tarwe geleds int scrij de voors			
			de reys gemaect schiedt. — — — — —			

Non-decimal currency nomenclature

lsd

libra	solidus	denarius
£	s.	d.
Pound	shilling	penny (pence)

Names for non-decimal currencies

- Pound sterling: 1 pound = 20 shillings; 1 shilling = 12 pence
- Pound Flemish: 1 pound = 20 schellingen; 1 schelling = 12 groten
- Holland guilders: 1 guilder = 20 stuivers; 1 stuiver = 16 penningen
- French crowns: 1 crown = 60 sous; 1 sous = 12 deniers
- Polish florins: 1 florin = 30 gros; 1 gros = 18 denar
- Portuguese real: 1 milréis = 1,000 réis

Problem space

Compound unit arithmetic

	£	s.	d.
	28	15	8
	32	8	11
	54	18	7
	18	12	9
Answer	£134	15s.	11d.
Unit total	132	53	35
Divide by base	-	53 / 20	35 / 12
Carried forward	2	2	-
Remainder	-	13	11

- Three separate units make up one value
- The units have non-decimal bases
- Need to use compound-unit arithmetic to normalize values
- The non-decimal bases differed by currency

Arithmetic by hand

$$\begin{array}{r}
 2.12.2 \frac{1}{2} \\
 2.2 \frac{1}{2} \\
 624.5 \\
 14642.12.2 \frac{1}{2} \\
 14640 \\
 12.12.2 \frac{1}{2} \\
 52. \\
 1304
 \end{array}$$

$$\begin{array}{r}
 4537/4= \\
 126 \\
 1039 \overset{1}{0} \overset{1}{7} \\
 265126 \\
 \hline
 1291.13.1 \\
 1304
 \end{array}$$

$$\begin{array}{r}
 2 \frac{1}{2} \\
 48 \quad 50 \\
 5.7.2 \frac{1}{2} \\
 107 \\
 104 \quad 52 \\
 1830.6.6 \\
 525 \quad 13.4 \quad 13/16 \\
 1304 \quad 13.1 \quad 8/16
 \end{array}$$

$$\begin{array}{r}
 3241.5.9 \\
 533.6.8 \\
 \hline
 3774.12.5 \\
 6.5 \\
 52 \\
 2102.13.7 \\
 2102 \quad 137 \\
 \hline
 4205 \quad 72
 \end{array}$$

1 part = 1830.6.6 $\frac{7}{16}$

4 parts = 7321.6.1 $\frac{1}{4}$

3 parts = 5490.19.6 $\frac{15}{16}$

$$1134.8.1 \frac{1}{8}$$

Profits

~~3241~~

Pirella	(94, 74b)	7168
Florette	(13, 72b)	4281
Naples	(6b, 43a)	1314
Fire rough	(40, 20b)	782
Ardassa	(81, 68a)	34
Linen		
Ardassa	(87, 68a)	44
Linen		
Linen		

$$\begin{array}{r}
 38.125 \\
 1.19.1 \frac{1}{2} \\
 \hline
 8 \\
 2 \\
 81.3.5
 \end{array}$$

1501
4.9 18.0

7009. Silk

$$\begin{array}{r}
 13,626.4.5 \\
 45000 \quad 17 \quad 7 \frac{1}{4} \\
 300 \\
 \hline
 44694 \quad 24 \frac{3}{4}
 \end{array}$$

Winninge ende Verlies

Debit 324.2

$$\begin{array}{r}
 324.16.0 \\
 100 \\
 90.10.0 \\
 10.16.8 \\
 39.16.33 \\
 0.3.1 \frac{1}{2} \\
 0.4.5 \\
 464.13.6 \\
 682 \quad 0.1 \frac{1}{2}
 \end{array}$$

Small Debt 1713.0.2

$$\begin{array}{r}
 22880.15.0 \frac{1}{2} \\
 3584.17.4
 \end{array}$$

$28178.12.6 \frac{1}{2}$

Came on Both Credit and debit

Small Debits + Business expenses =

$$5297.17.6$$

Design principles for a solution

- A class that maintains the tripartite structure of non-decimal currencies
- Decimalized class as fall back
- Track the bases of shillings and pence units
- Vectors with different bases cannot be combined
- Choose and track unit represented by decimalized class
- Vectors with different units can be combined but need coercion path

Structure of the classes

Tripartite structure

```
deb_lsd(l = c(17, 32, 18),  
        s = c(16, 7, 12),  
        d = c(6, 9, 3))
```

```
#> <deb_lsd[3]>  
#> [1] 17:16s:6d 32:7s:9d  
#> [3] 18:12s:3d  
#> # Bases: 20s 12d
```

Decimalized

```
deb_decimal(x = c(17.8250,  
                  32.3875,  
                  18.6125))
```

```
#> <deb_decimal[3]>  
#> [1] 17.8250 32.3875  
#> [3] 18.6125  
#> # Unit: pounds  
#> # Bases: 20s 12d
```

Normalization

Compound unit arithmetic

	£	s.	d.
	28	15	8
	32	8	11
	54	18	7
	18	12	9
Answer	£134	15s.	11d.
Unit total	132	53	35
Divide by base	-	53 / 20	35 / 12
Carried forward	2	2	-
Remainder	-	13	11

```
deb_normalize(c(132, 53, 35))
```

```
#> <deb_lsd[1]>
```

```
#> [1] 134:15s:11d
```

```
#> # Bases: 20s 12d
```

Multiplication

RULE II. "If the multiplier be a composite number, whose component parts do not exceed 12, multiply first by one of these parts, then multiply the product by the other. Proceed in the same manner if there be more than two."

Ex. 1st.] L. 15 3 8 by 32 = 8 × 4
8

L. 121 9 4 = 8 times.
4

L. 485 17 4 = 32 times.

Multiply £15 3s. 8d.
sterling by 32

deb_lsd(15, 3, 8) * 32

#> <deb_lsd[1]>

#> [1] 485:17s:4d

#> # Bases: 20s 12d

Division

RULE I. "When the dividend only consists of different denominations, divide the higher denomination, and reduce the remainder to the next lower, taking in (p. 296. Rule V.) the given number of that denomination, and continue the division."

Examples.

Divide L. 465 : 12 : 8
by 72.

L.	s.	d.	L.	s.	d.
72) 465	12	8	(6	9	4
432	..	.			

33
20

72) 672
648

24
12

72) 296
288

8 Rem.

Or we might divide by the component parts of 72, (as explained under *Thirdly*, p. 298).

Divide 345 cwt. 1 q. 8 lb.
by 22.

Cwt.	q.	lb.	Cwt.	q.	lb.
22) 345	1	8	(15	2	21
22	..	.			

125
110

15
4

22) 61
44

17
28

144
34

22) 484
44

44
44

0

Divide 345cwt. 1q. 8lbs.
by 22

```
x <- deb_lsd(345, 1, 8,
              bases = c(4, 28))
```

```
x / 22
```

```
#> <deb_lsd[1]>
```

```
#> [1] 15:2s:22d
```

```
#> # Bases: 4s 28d
```

Arithmetic

```
deb_lsd(15, 15, 9) + deb_lsd(6, 13, 4)
```

```
#> <deb_lsd[1]>
```

```
#> [1] 22:9s:1d
```

```
#> # Bases: 20s 12d
```

Comparison

```
deb_lsd(15, 15, 9) < deb_lsd(10, 128, 432)
```

```
#> TRUE
```


debkeepr: Analysis of Non-Decimal Currencies in R

debkeepr 0.0.5.9000



Reference

Get started

Articles ▾

News



debkeepr: Analysis of Non-Decimal Currencies

debkeepr integrates non-decimal currencies that use the tripartite system of pounds, shillings, and pence into the methodologies of Digital Humanities and the practices of reproducible research. The package makes it possible for historical non-decimal currencies to behave like decimalized numeric values through the implementation of the `deb_lsd` and `deb_decimal` vector classes or types. These types are based on the infrastructure provided by the [vctrs package](#). debkeepr simplifies the process of performing arithmetic calculations with non-decimal currencies — such as adding £3 13s. 4d. sterling to £8 15s. 9d. sterling — and also provides a basis for analyzing account books with thousands of transactions recorded in non-decimal currencies. The name of the debkeepr package derives from this latter capability of analyzing historical account books that often used [double-entry bookkeeping](#).

Installation

You can install debkeepr from GitHub with [remotes](#):

```
# install.packages("remotes")
remotes::install_github("jessesadler/debkeepr")
```

Please open an [issue](#) if you have any questions, comments, or requests.

Historical Background

The debkeepr package uses the nomenclature of [l](#), [s](#), and [d](#) to represent pounds, shillings, and pence units in non-decimal currencies. The abbreviations derive from the Latin terms [libra](#), [solidus](#), and [denarius](#). The libra was a Roman measurement of weight, while the solidus and denarius were both Roman coins. The denarius was a silver coin from the era of the Republic, in contrast to the golden solidus that was issued in the Late Empire. As the production of silver coins overtook that of gold by the 8th century, a solidus came to represent 12 silver denarii coins, and 240 denarii were — for a time — made from one libra or pound of silver. The custom of

Links

Browse source code at

<https://github.com/jessesadler/debkeepr/>

Report a bug at

<https://github.com/jessesadler/debkeepr/issues>

License

[Full license](#)

MIT + file [LICENSE](#)

Developers

[Jesse Sadler](#)

Author, maintainer

Dev status

build

codecov 99%

lifecycle

<https://jessesadler.github.io/debkeepr/>

GitHub: <https://github.com/jessesadler/debkeepr/>

THE
MERCHANTS
MIRROUR.

OR,
DIRECTIONS
For the perfect Ordering and Keeping of his
ACCOUNTS.

Framed by way of DEBITOR and CREDITOR, after the
(so termed) *Italian Manner*: Containing 250 *Rare Questions*,
With their Answers, in forme of a DIALOGUE.

AS LIKEWISE

A WASTE-BOOK, with a complete JOURNAL and LEGER thereunto appertaining;
Unto the which I have annexed two other *Waste-Books* for exercise of the *Students*: and at the
end of each is entred the brief Contents of the *Leagers Accounts*, arising from thence.

AND ALSO

A MONETH-BOOK, very requisite for Merchants, and commodious for all other
SCIENCE-LOVERS of this Famous Art.

The Third Edition, Corrected and Amended.

Compiled by RICHARD DAFFORNE of Northampton, Accountant, and
Teacher of the same, after an Exquisite Method, in the
English and Dutch Language.

J. Vanden V.

Soo eenigh licht-verispens' Pan,
Myn werk beracht, of soecht 'onteerren,
Die maecthet beeter, Soo hy kan:
'Ikhebb groote lust noch meer te Leeren.

H. L. S.

Dordeelt iemant 'dooz het Leezeh;
Niet goet kan syn Dordel wezen.

Leerende, leere ick.

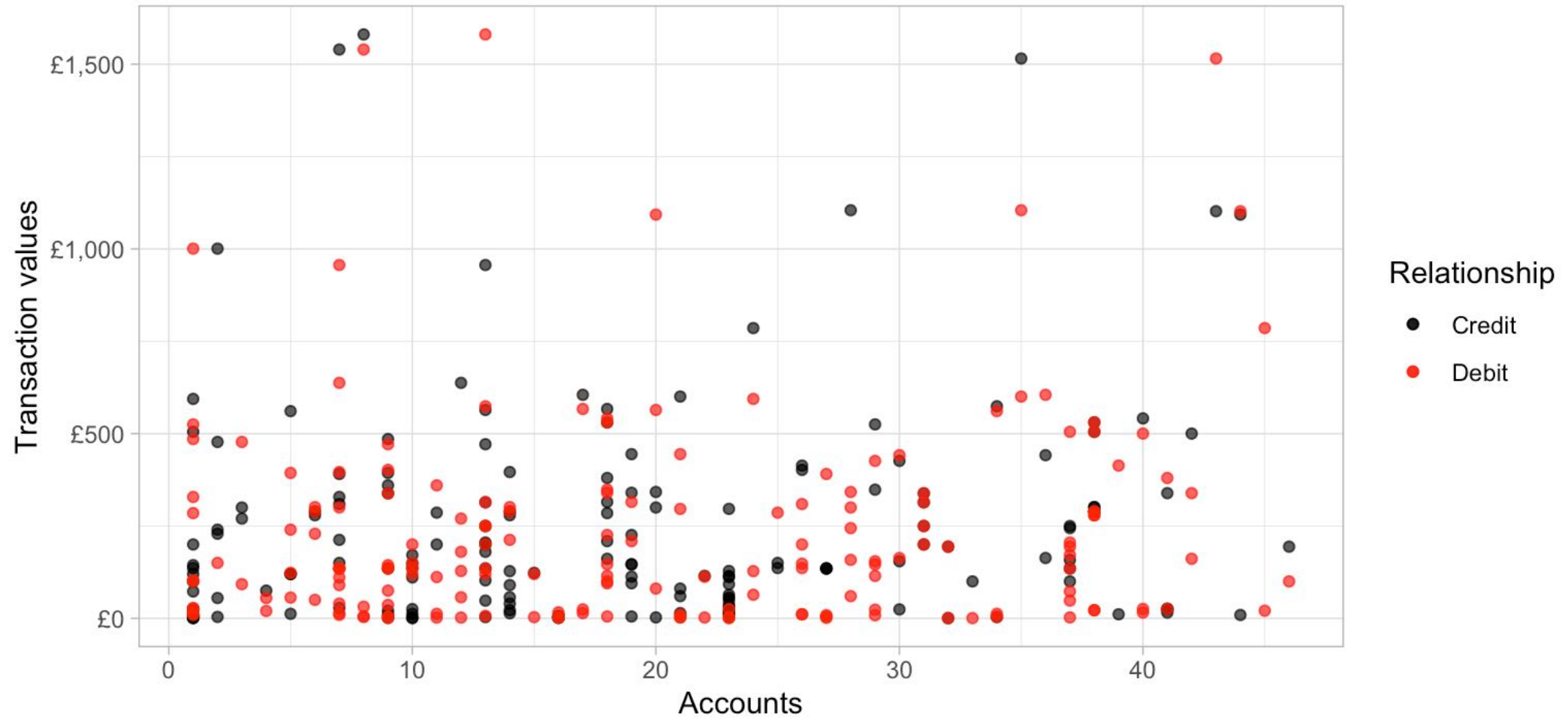
The Contents are immediately prefixed before the Book.

LONDON,

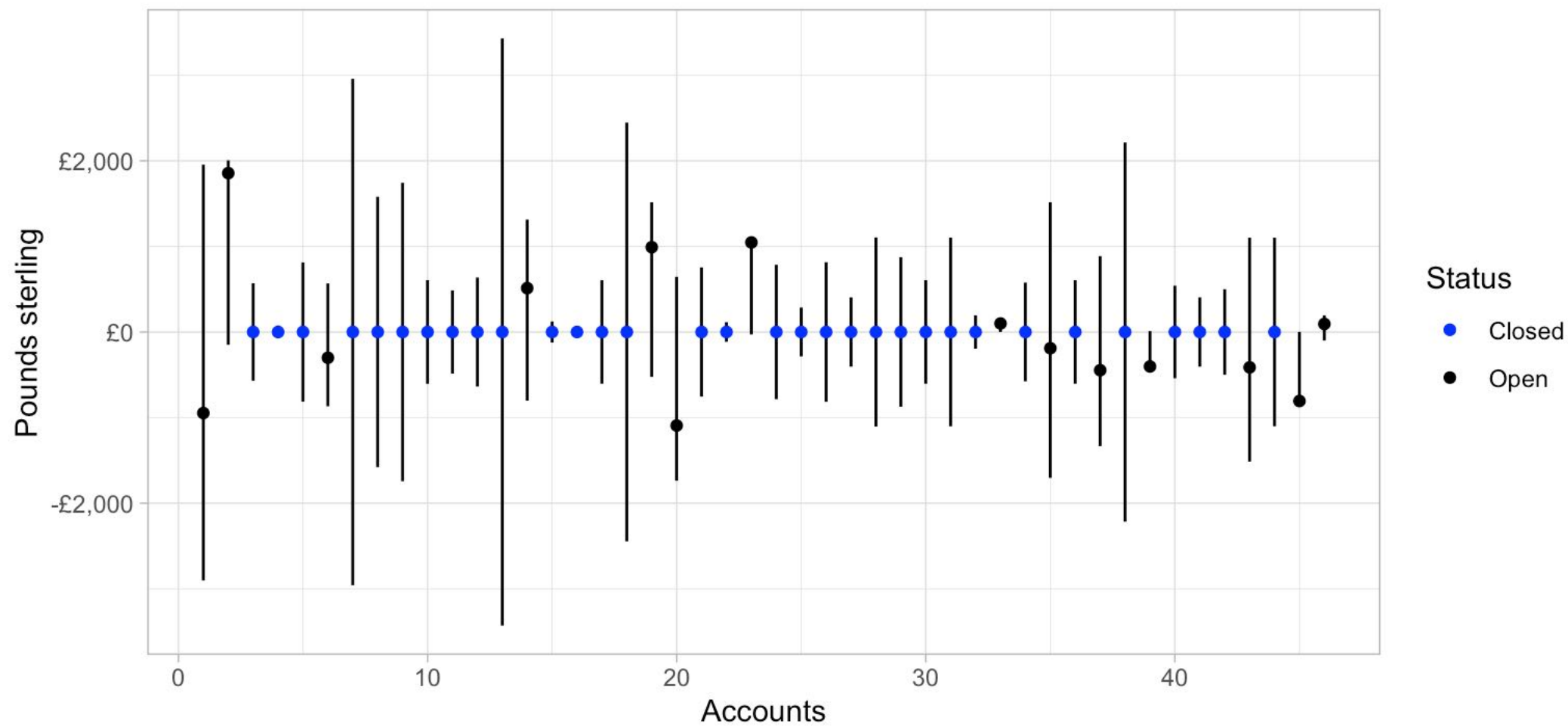
Printed by R. H. and J. G. for Nicholas Bourn, at the South-entrance of the
Royall Exchange, 1660.

Richard Dafforne,
*The Merchant's
Mirrour
(1660)*

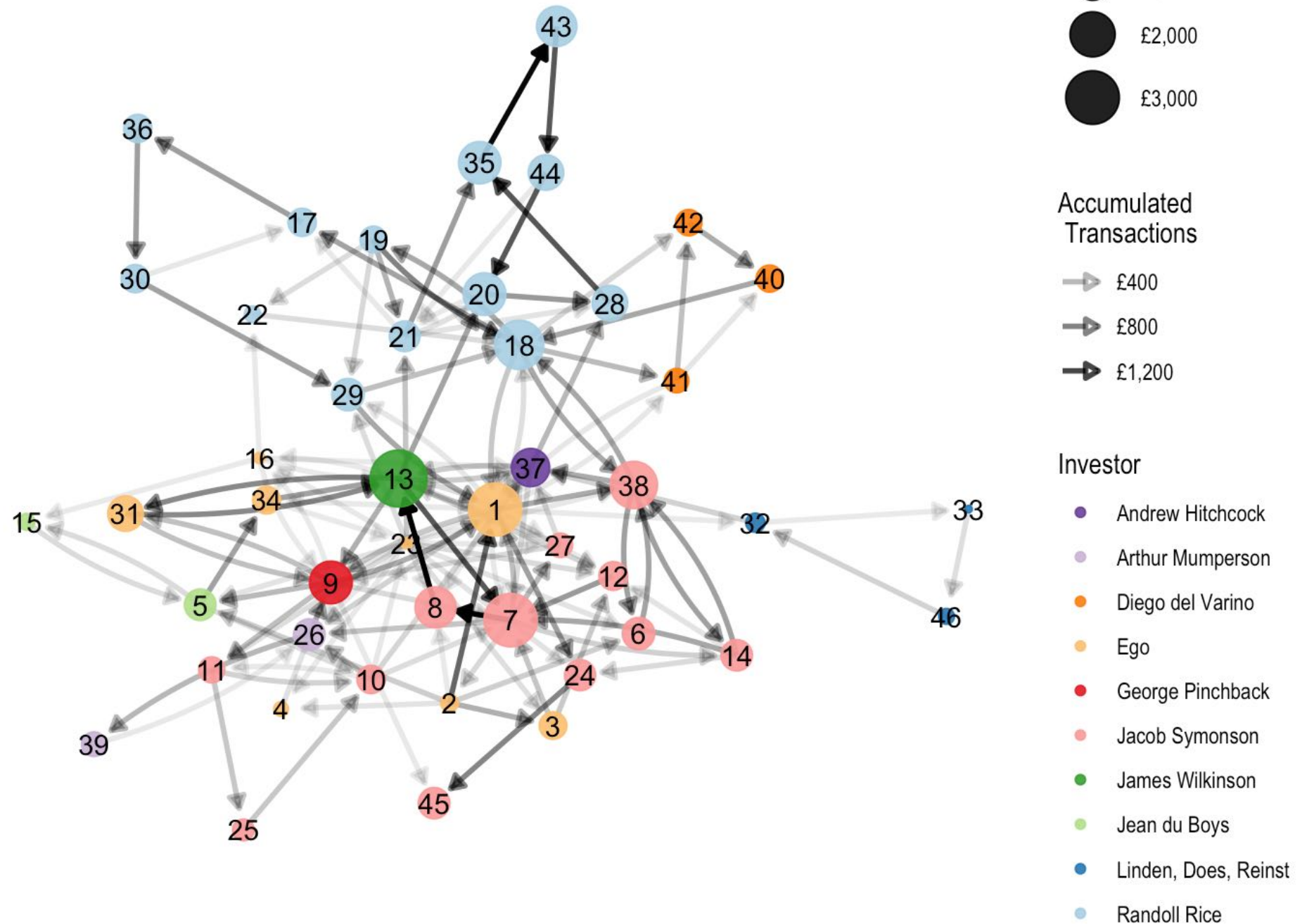
Value of Transactions by Accounts



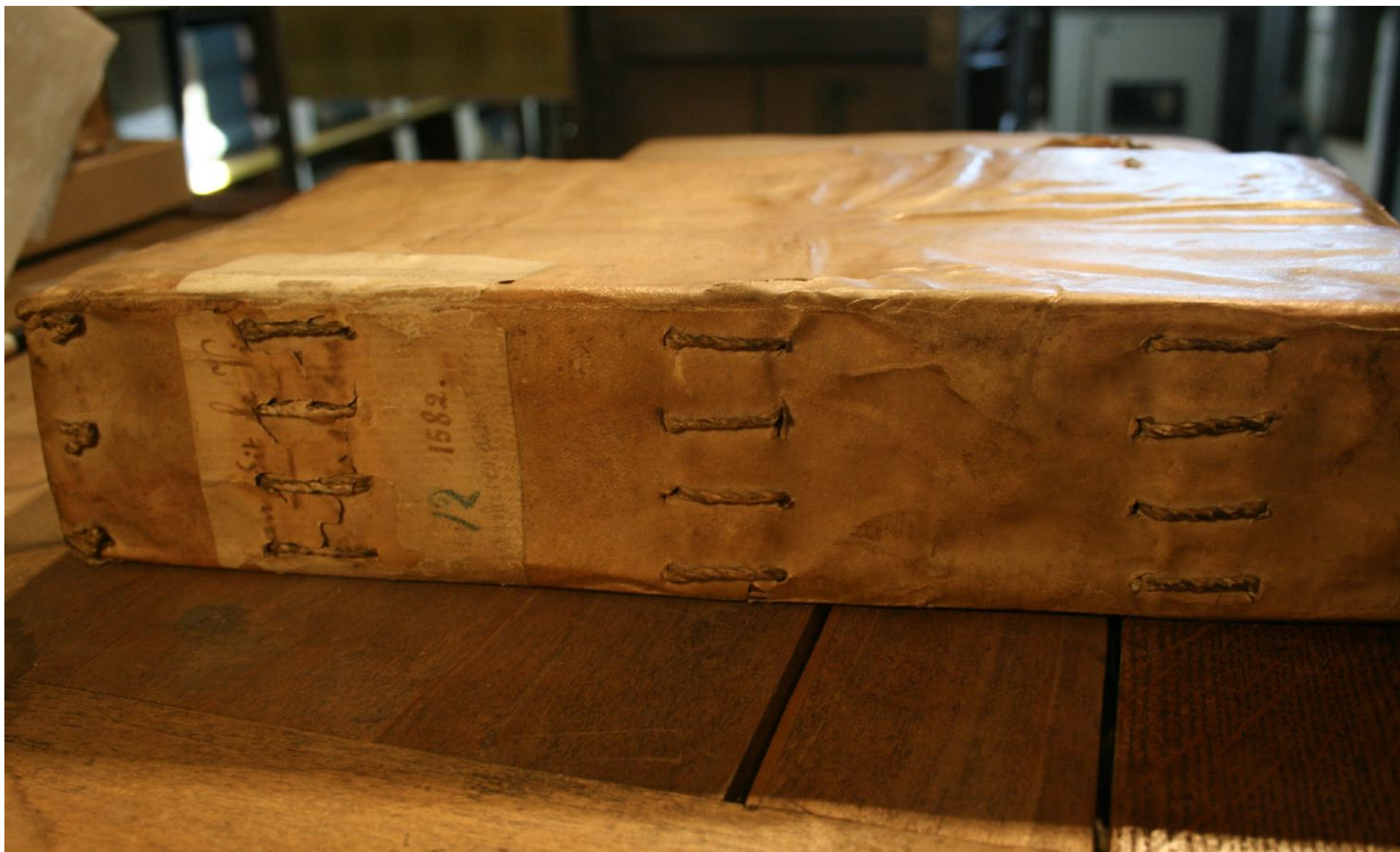
Summary of the accounts



Network of Dafforne's Journal



Example: The estate of Jan della Faille de Oude (1515–1582)



sal Worden ——— £ 39: 16: 3:

Beloopende alsoodt, v. g. v. g.
credit ——— £ 1337: 17: 3:

Inde alsoo £ 24: merck dat, elc
voors. debet is bedragende 1 Maer
gemixet de voers. partijes is credit
gedraget, v. g. v. g. is debet ende
tot laste vande voers. rekeninges
aldaer die onder andere partijes
tot proffijte gebracht is v. g. v. g.
voort Worden is compt. d. g. v. g.
gatende alhier tot proffijte ——— Met

De £ 24: die de voers. p. m. v. g.
als boude is credit aduanteur, als
op 26 december 1583 noch open staet
Gulles, ten naostere rekeninge v. g.
ant. voort Worden is p. x. ——— Memoria

alm Item Jan van der Leide f. 9 o. fte
staet fo. 150: debiteur van ———
£ 140: 0: 0:
— Day gemixet die op 26: december

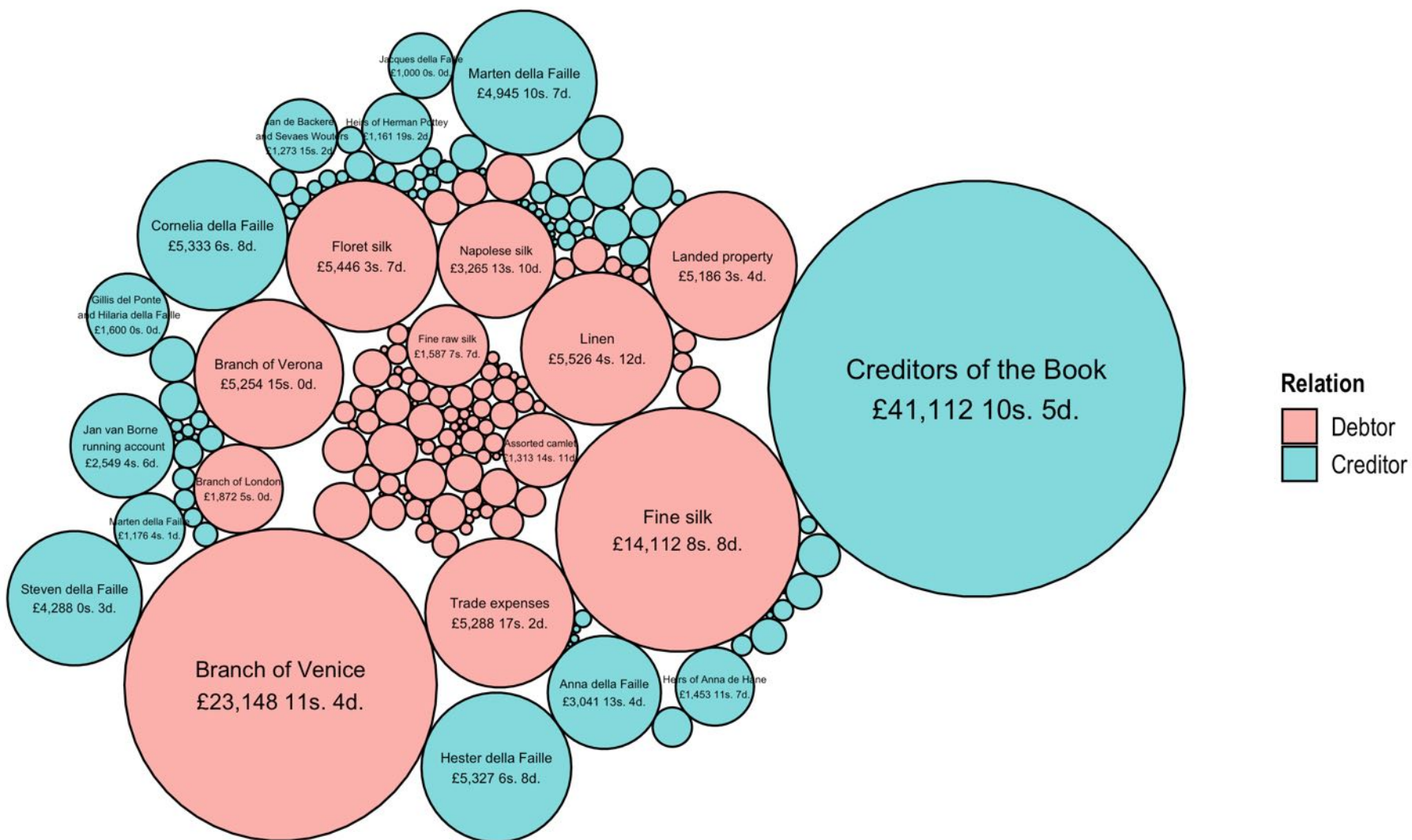
Accounts of the estate of Jan della Faille de Oude

- Date: 8 November 1582 to 31 December 1594
- 8 November 1582 to 26 December 1583
- 26 December 1583 to 31 December 1594

- **Jan** (c. 1542–1618): Executor of estate but not successor to his father.
- **Anna** (c. 1543–1622): Married Robert van Eeckeren, a close associate of her father.
- **Marten** (c. 1544–1620): Chosen by his father as his primary successor.
- **Carlo** (c. 1546–1617): Constant source of trouble for his siblings.
- **Jacques** (c. 1549–1615): Executor of estate. Moved to Haarlem in 1584.
- **Steven** (c. 1550–1621): Married twice only to have his marriages annulled through the will of his father.
- **Maria** (1555–1578): Married and had three children before her early death.
- **Hester** (c. 1558–1643): Married the merchant Daniel van der Meulen.
- **Cornelia** (c. 1563–1582): Died shortly after her father.

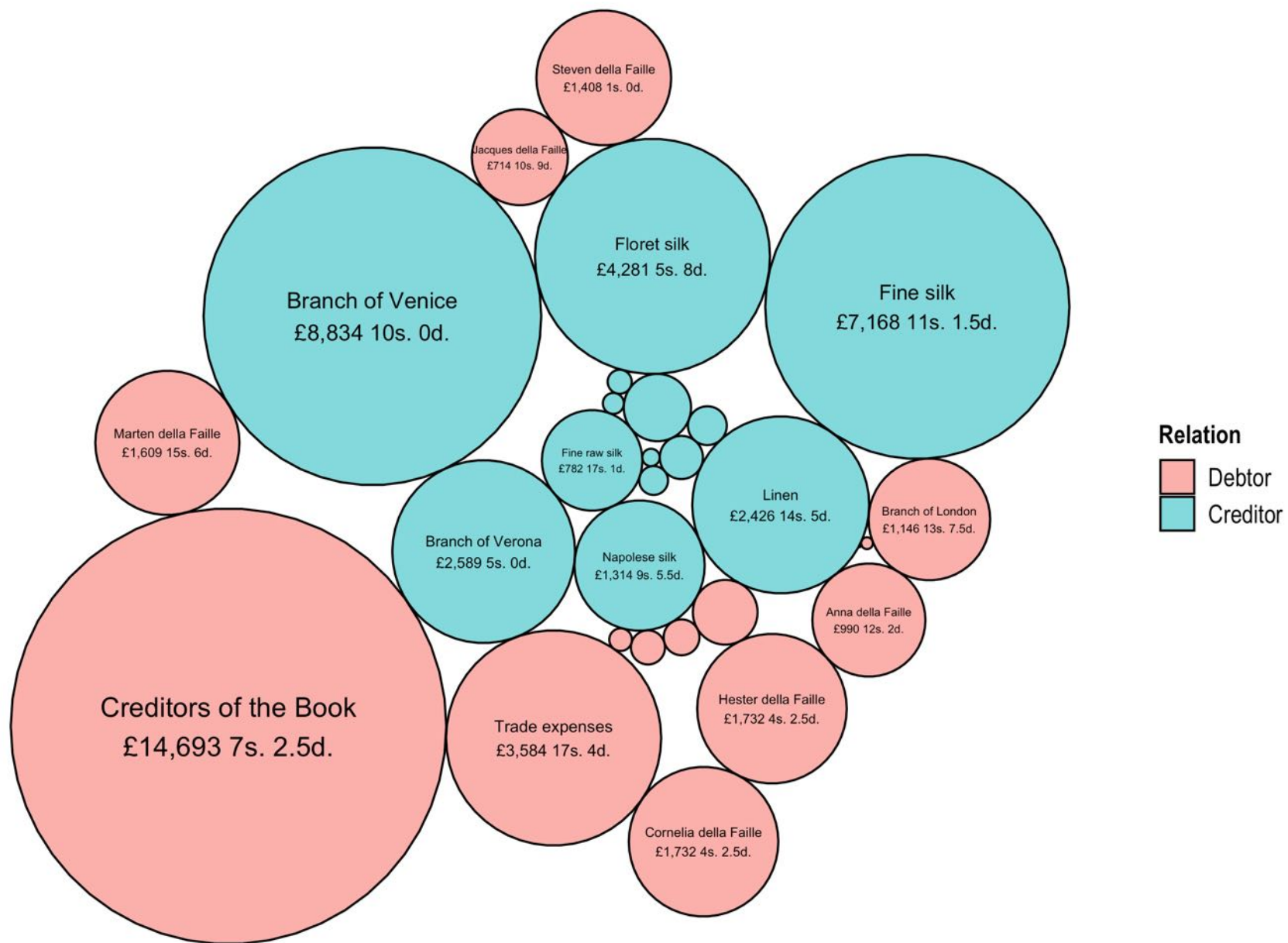
Value of accounts in the estate of Jan della Faille de Oude, 8 December 1582

Opening value of the estate: £82,813 5s. 8d.



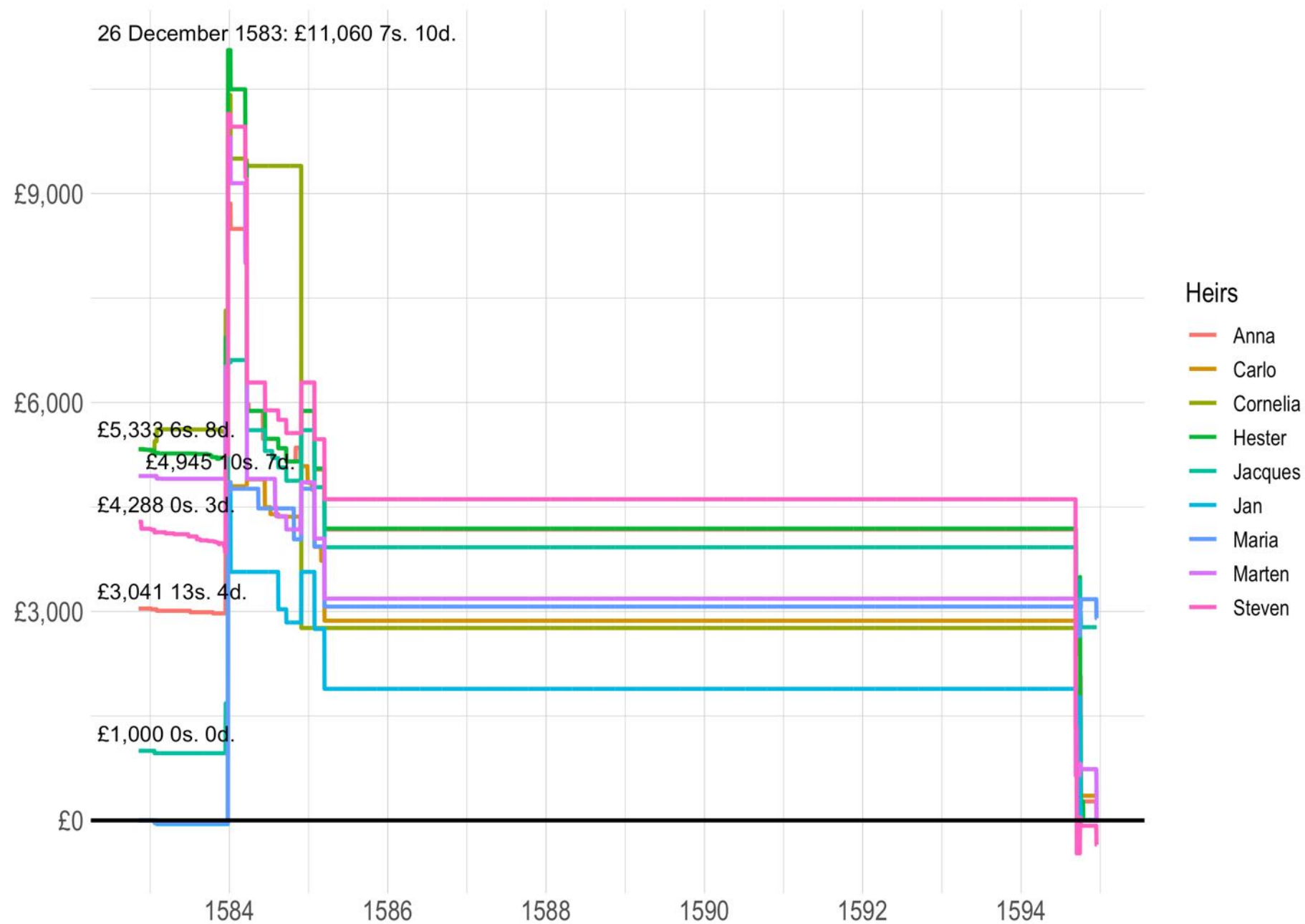
Profits and losses in the trade of Jan de Oude

1 January 1579 to 26 December 1583



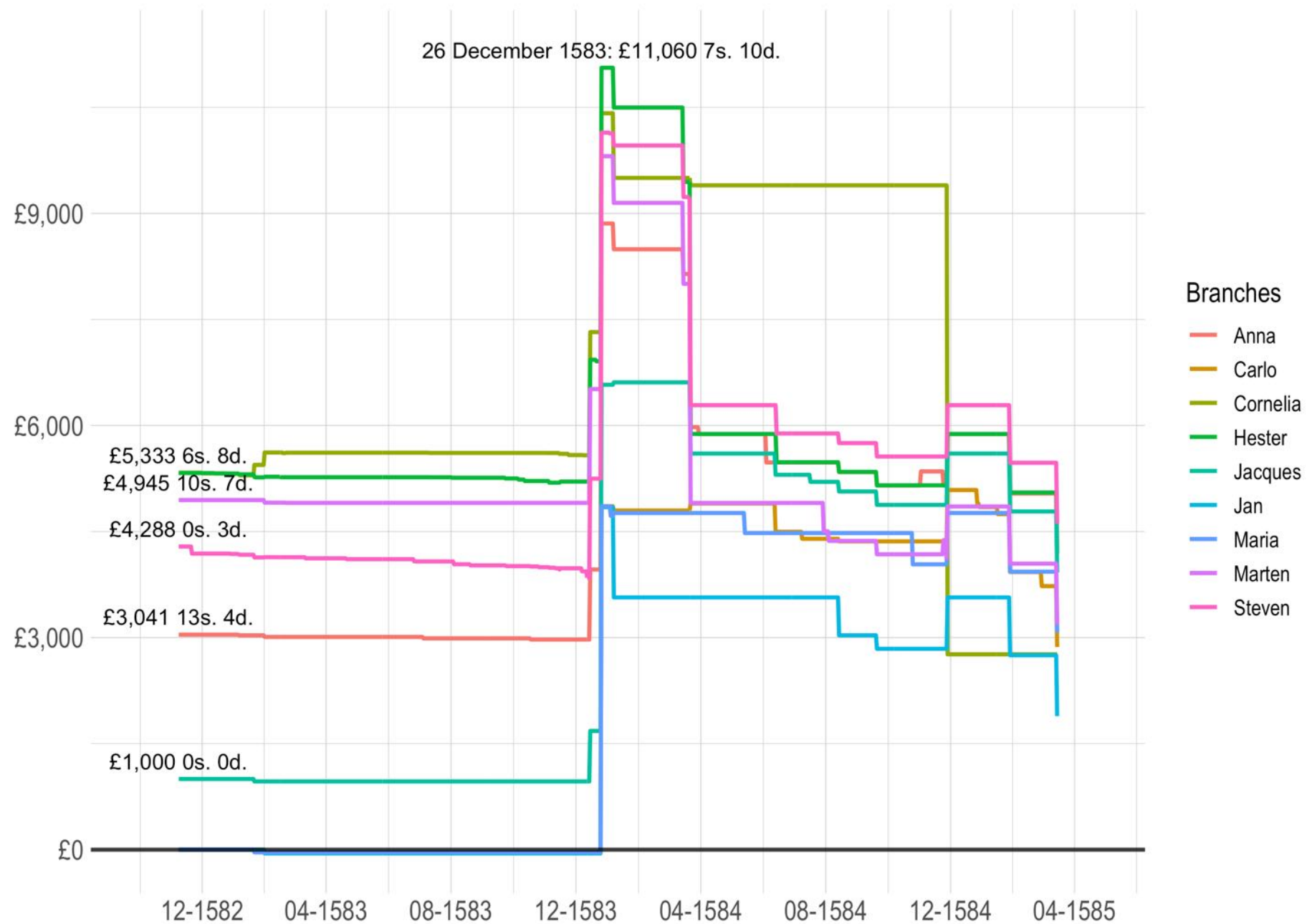
Inheritance due to the heirs of Jan de Oude

Estate of Jan della Faille de Oude, 1582–1594



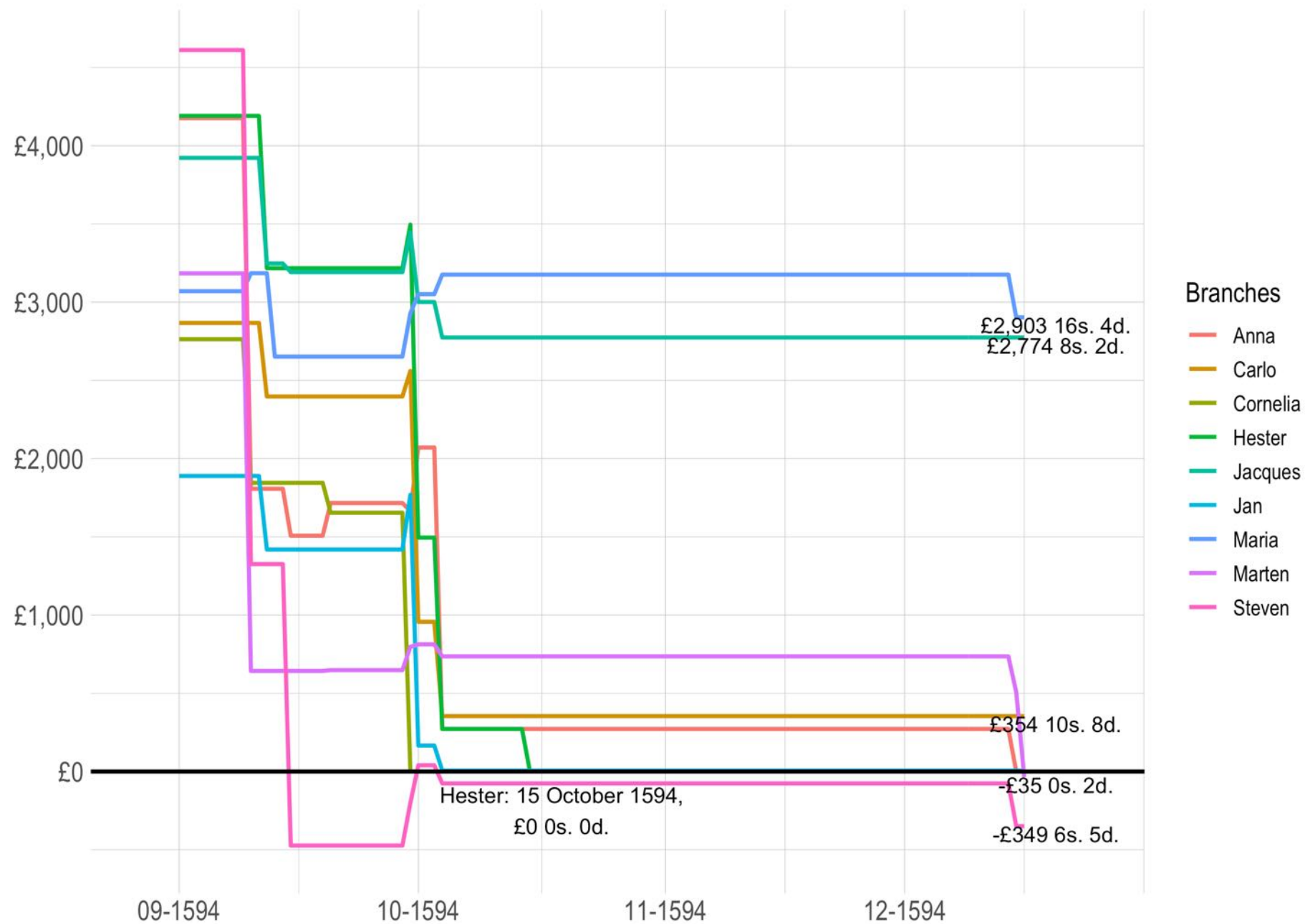
Inheritance due to the heirs of Jan de Oude

November 1582 to March 1585



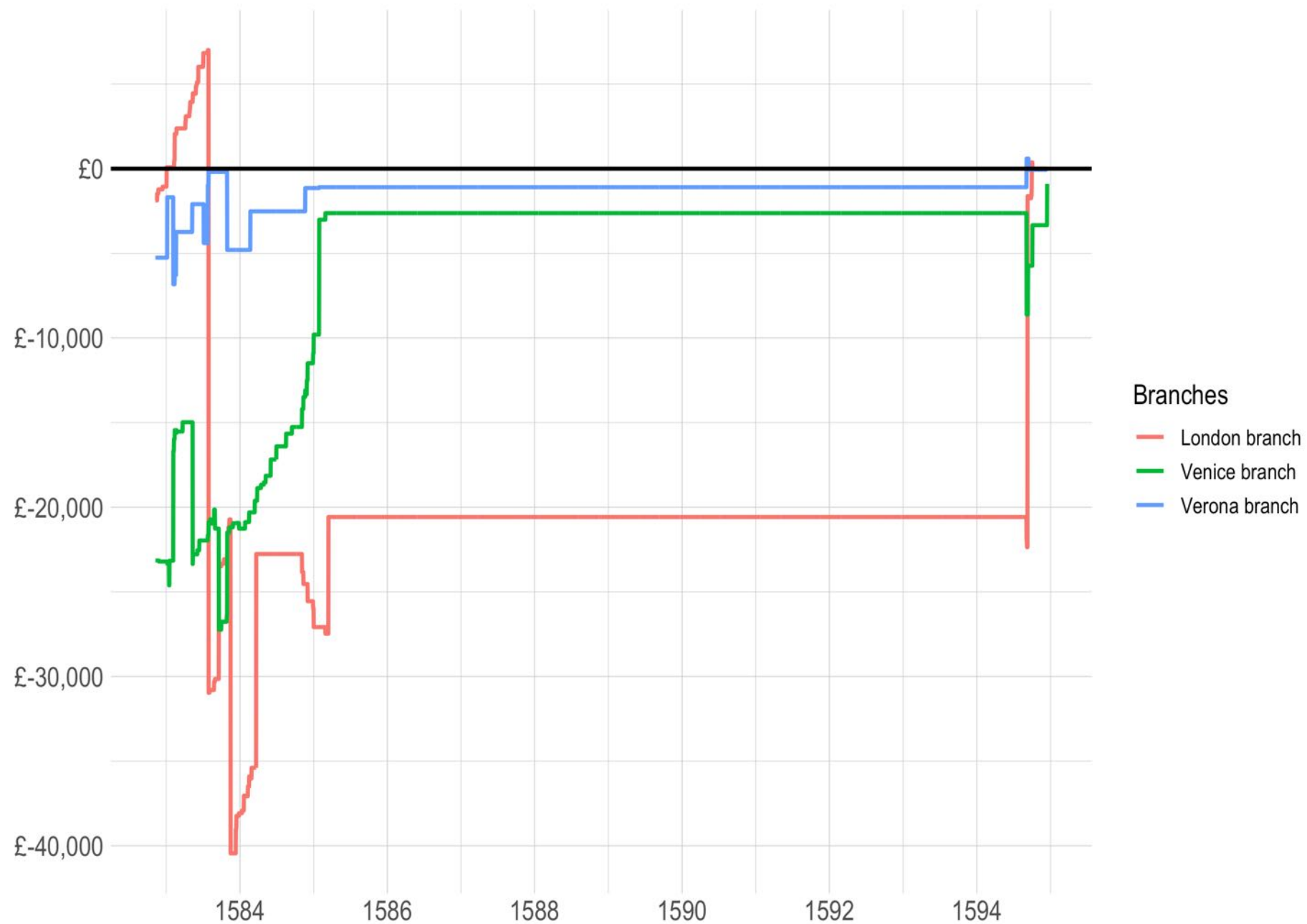
Inheritance due to the heirs of Jan de Oude

September 1594 to 16 December 1594



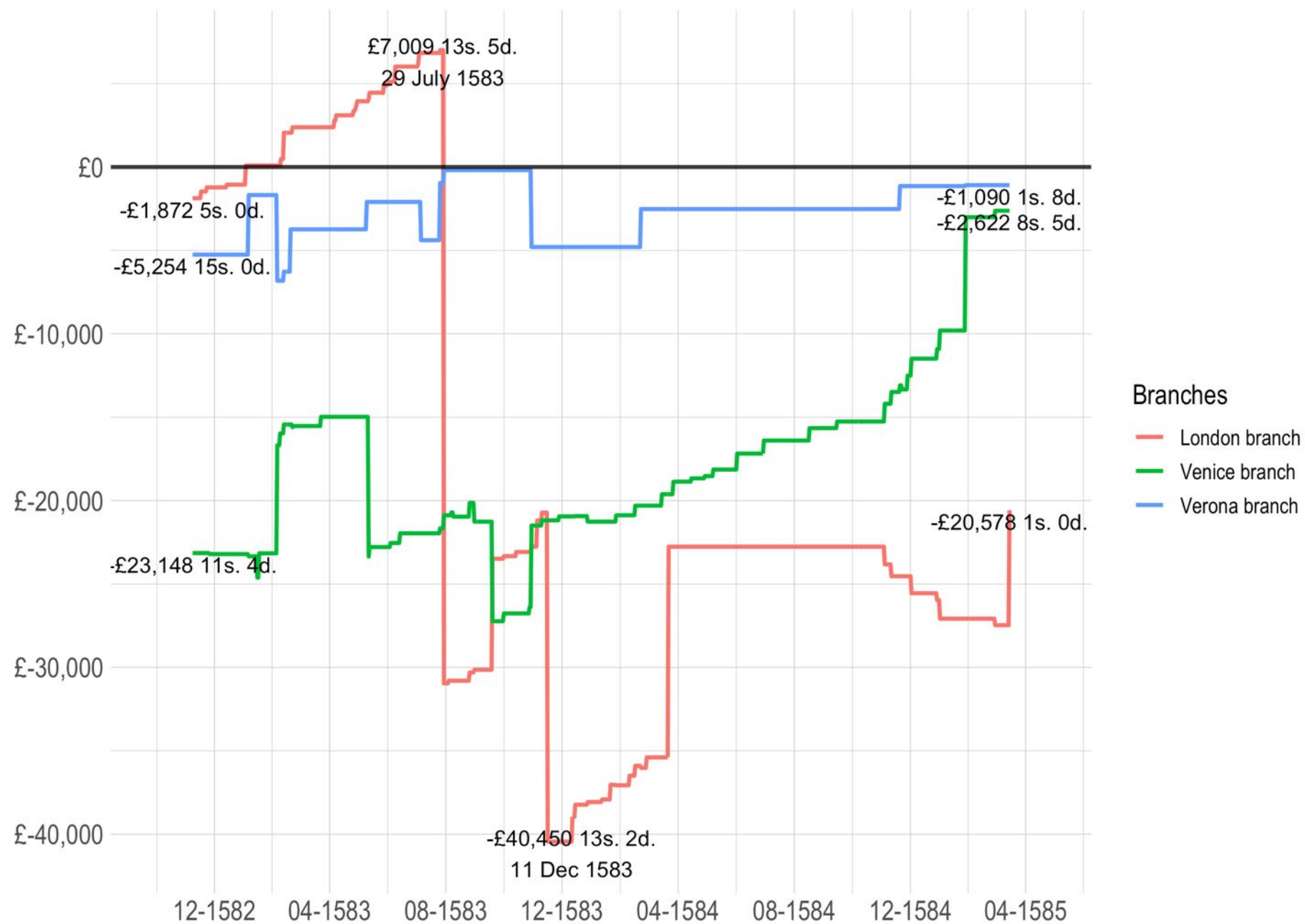
Running values of the branches in the trade of Jan de Oude

Estate of Jan della Faille de Oude, 1582–1594



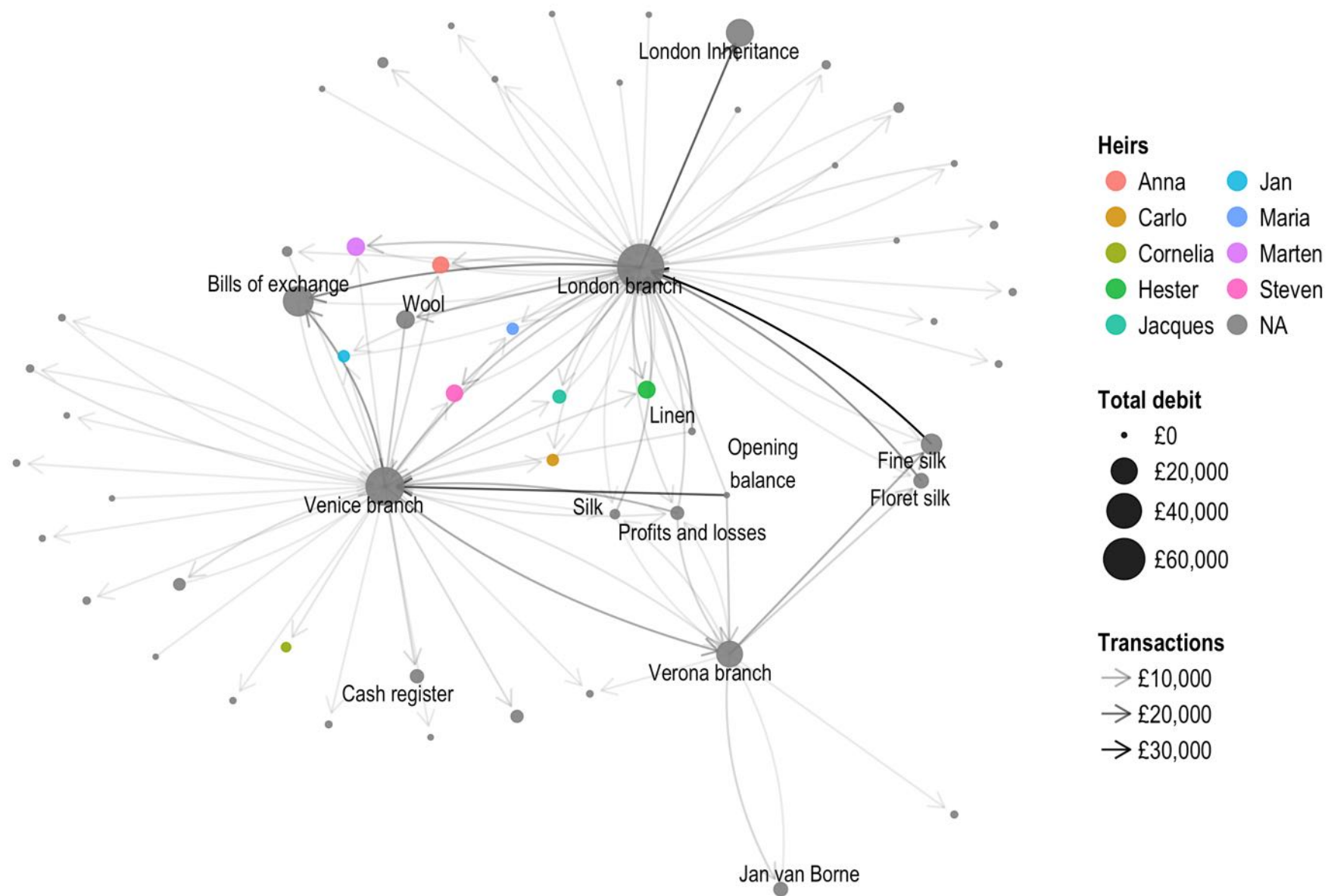
Running values of the branches in the trade of Jan de Oude

November 1582 to March 1585



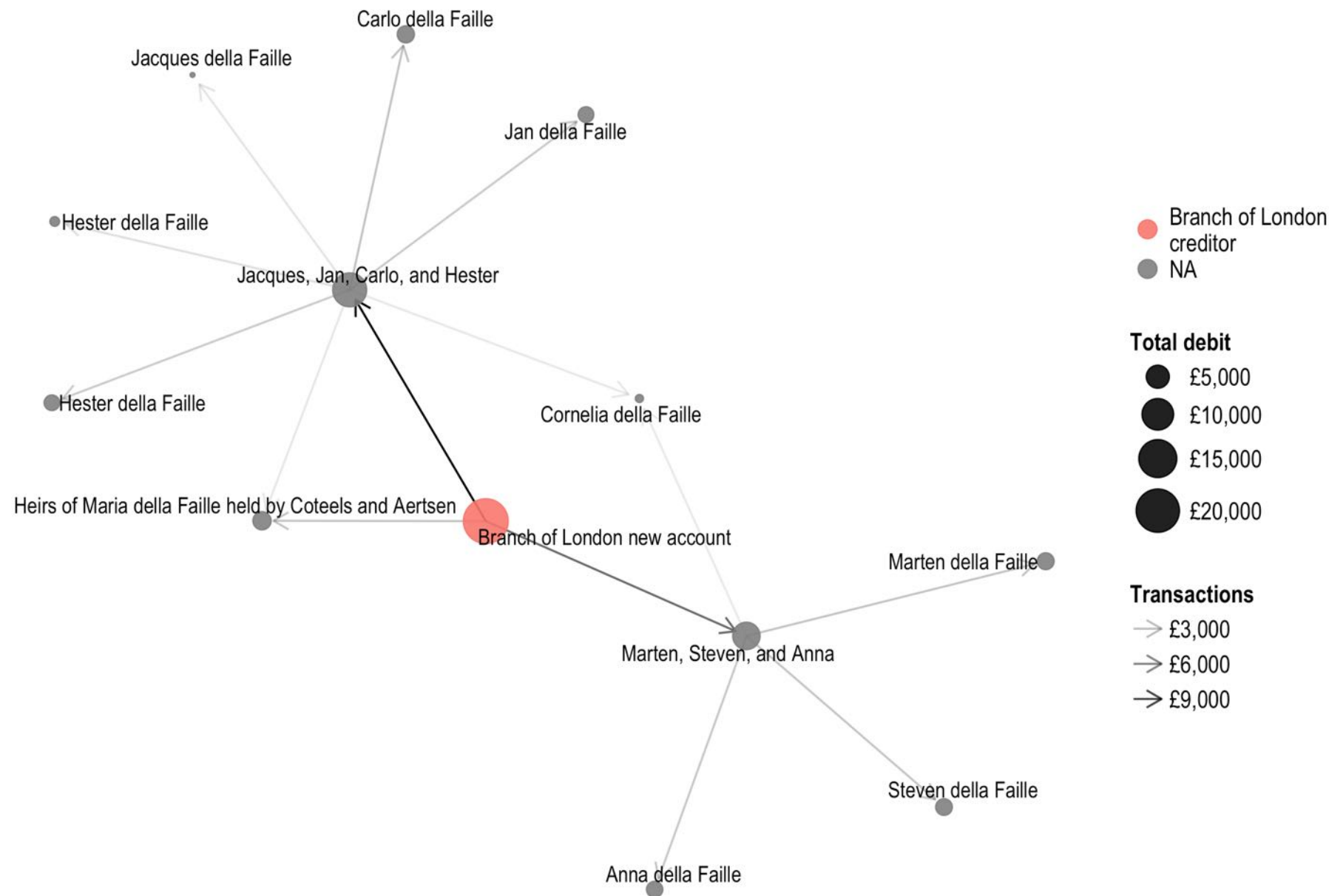
Subgraph of the branches in the trade of Jan de Oude

Estate of Jan della Faille de Oude, 1582–1594

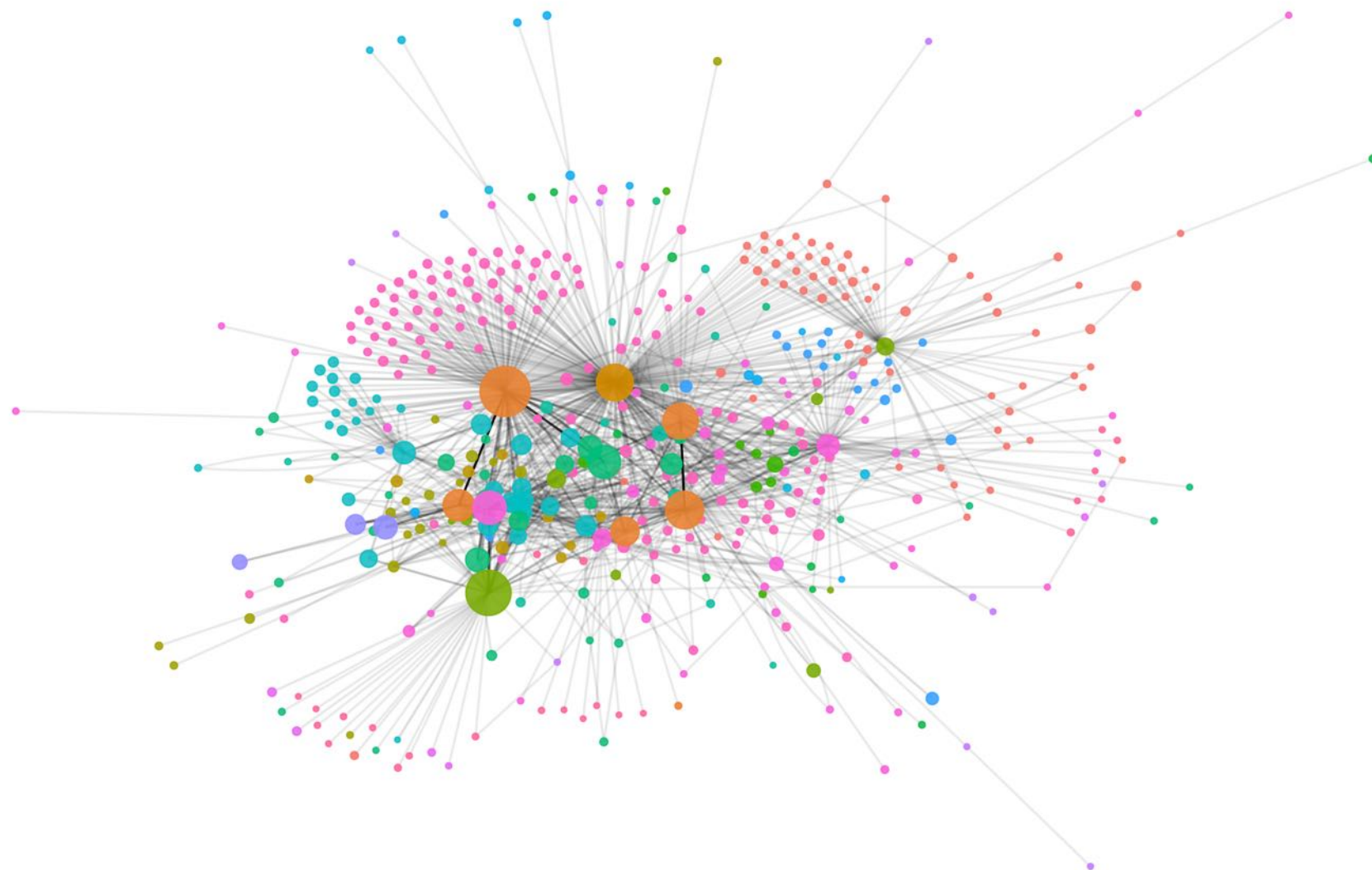


Subgraph of the inheritance from London

Estate of Jan della Faille de Oude, 1582–1594



Estate of Jan della Faille de Oude, 1582–1594



Account types

Bequest Branch Cassa Company Erffgoed Estate Factor Giovane Goods Heir
Inheritance Kin Law Loan London Inheritance Miscellaneous Political Trade Wissel Written off

Thank you

Jesse Sadler

Twitter: @vivalosburros

website: jessesadler.com

GitHub: github.com/jessesadler

Slides: jessesadler.com/slides/chico2021.pdf