Analysing the Structure of John Finnemore's Souvenir Programme, Series 9

William Denton

17 June 2021

INTRODUCTION

Series 9 of *John Finnemore's Souvenir Programme* is one of the finest things I've ever heard done on radio.

Structurally, it's six half-hour episodes about an extended family, focusing on five members: young parent Russ, his mother Deborah, her father Jerry, his mother Vanessa, and also Uncle Newt, who's not really an uncle but has been attached to the family for decades. Each episode has scenes that go in reverse chronological order, and the first five episodes move back through the family from Russ to Newt. The sixth is different.

There is much to be said about the series, but this is just about the structure and chronology of the episodes, which fascinated me and I wanted to understand more thoroughly. This analysis is based on a spreadsheet with details of every scene, available at:

<https://www.miskatonic.org/files/jfsp-series-9.ods>.

I posted about this at:

<https://www.miskatonic.org/2021/06/17/jfsp/>

SETUP

(I wrote this in Org with R code blocks embedded in it. I'll show the workings in the spirit of openness and reproducability, but don't let that put you off—just look at the charts.)

First, set up the R session. I use three packages, which will need to be installed if they aren't already.

```
## install.packages(c("readODS", "tidyverse", "scales"))
```

Now load them in.

```
library(readODS)
library(tidyverse)
library(scales)
```

Load the data from the spreadsheet, clean it up, and set up some groups of people.

```
j <- read_ods("series-9-data.ods", sheet = 1) %>%
    as_tibble() %>%
    mutate_if(is.numeric, replace_na, 0) %>%
    mutate(Scene = str_pad(Scene, 2, pad = "0")) %>%
    mutate(ep = paste0(Episode, "." , Scene)) %>%
    mutate(ep = as.factor(ep))

five_people <- c("R", "D", "J", "V", "N")
## ten_people <- c("R", "A", "T", "D", "J", "H", "V", "N", "G", "S")
all_people <- c("R", "A", "T", "D", "J", "H", "V", "W", "N", "G", "S", "P")
```

PEOPLE THROUGH TIME

The data in the spreadsheet is in wide form, and we want it long and tidy, so pull out Year and Order plus all the people columns and transform that into a long table.

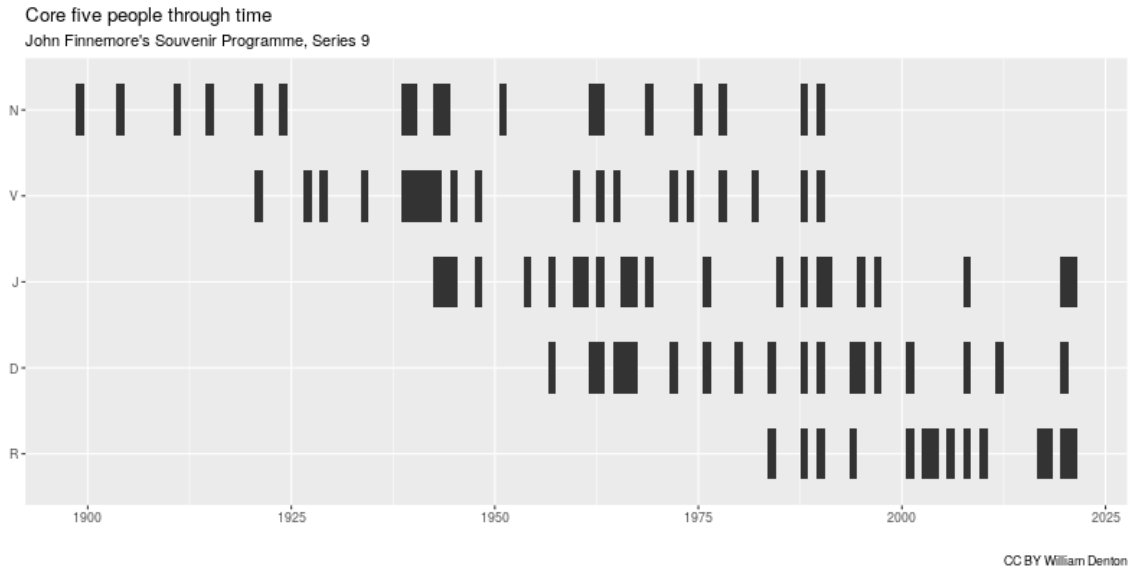
```
people_and_years <- j %>%
    select(Year, Order, R:W) %>%
    pivot_longer(!c(Year, Order), names_to = "person", values_to = "x") %>%
    filter(! x == 0)
people_and_years$person <- factor(people_and_years$person, levels = all_people)
```

It looks like this (the x column is unused):

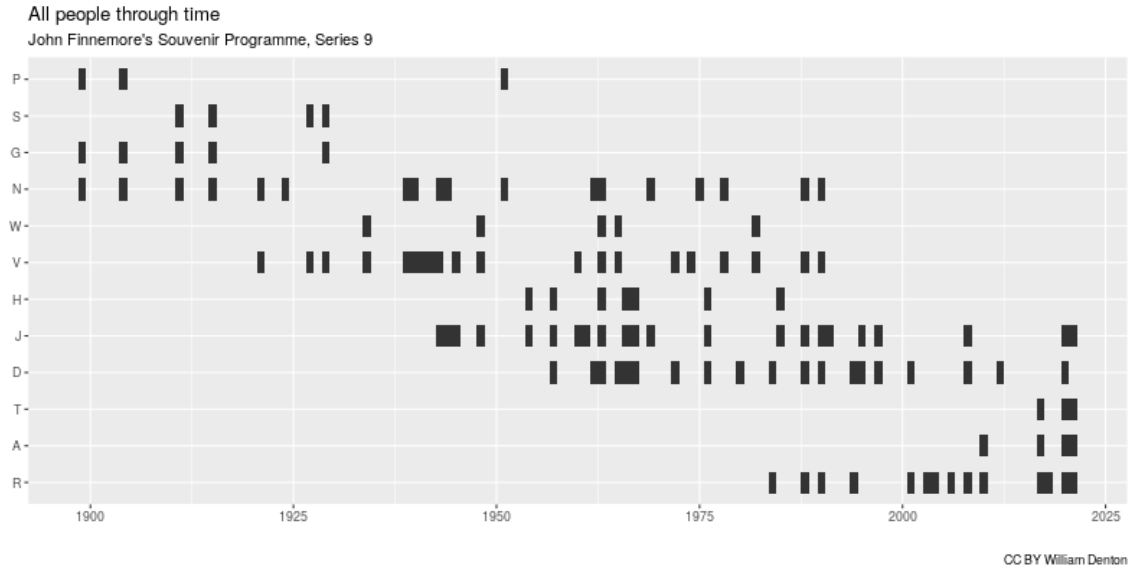
```
head(people_and_years)
```

Year	Order	person	x
2020	2	R	1
2020	2	D	1
2020	2	J	1
2020	2	A	1
2020	2	T	1
2018	0	R	1

This is ready for charting. The core five go in order up the y-axis: Russ, Deborah, Jerry, Vanessa, Newt.



We can also chart everyone through time. Here the order is: Russ, Alex, Toby, Deborah, Jerry, Hilla, Vanessa, Walter, Newt, Gally, Susanna, Patrick. (Luckily everyone's name starts with a different letter—perhaps Finnemore did this to help with his planning?)



This nicely shows Newt, Gally and Patrick (and of course Lettie, who I left out, as I did Cliff) at the start, then Susanna entering, then Vanessa, and so on, with Alex and Toby appearing at the end.

PEOPLE THROUGH SCENES

Another way of looking at the data is tracking people through the scenes in the episodes. Here again we begin by transforming the data into a long format.

```
people_and_eps <- j %>%
    select(ep, R:W) %>%
    pivot_longer(! ep, names_to = "person", values_to = "x") %>%
    filter(! x == 0)
people_and_eps$person <- factor(people_and_eps$person, levels = all_people)
```

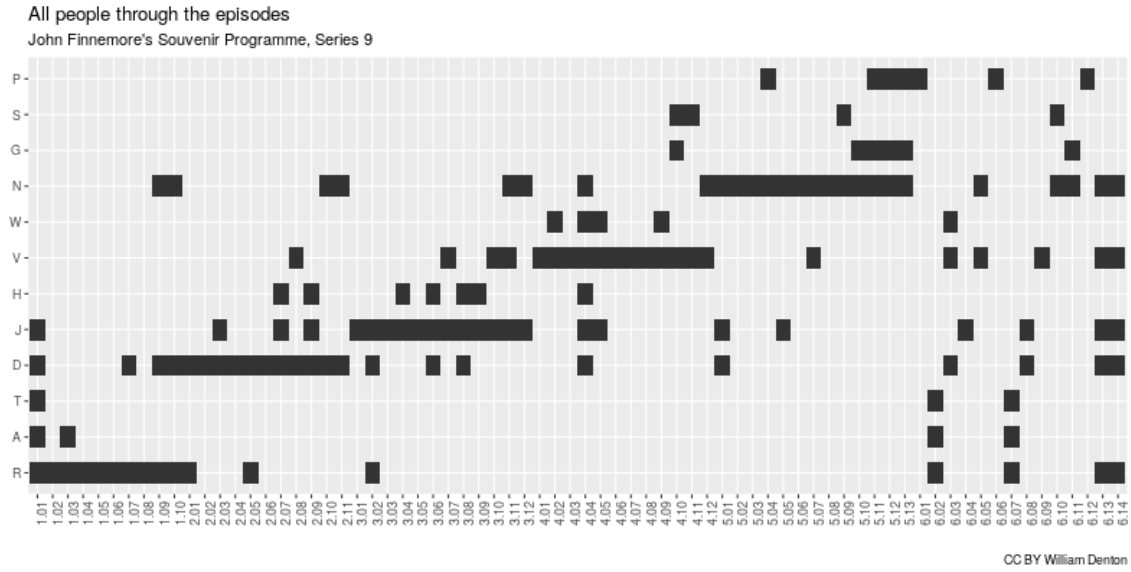
It looks like this (the x column is unused):

```
head(people_and_eps)
```

ep	person	x
1.01	R	1
1.01	D	1
1.01	J	1
1.01	A	1
1.01	T	1
1.02	R	1

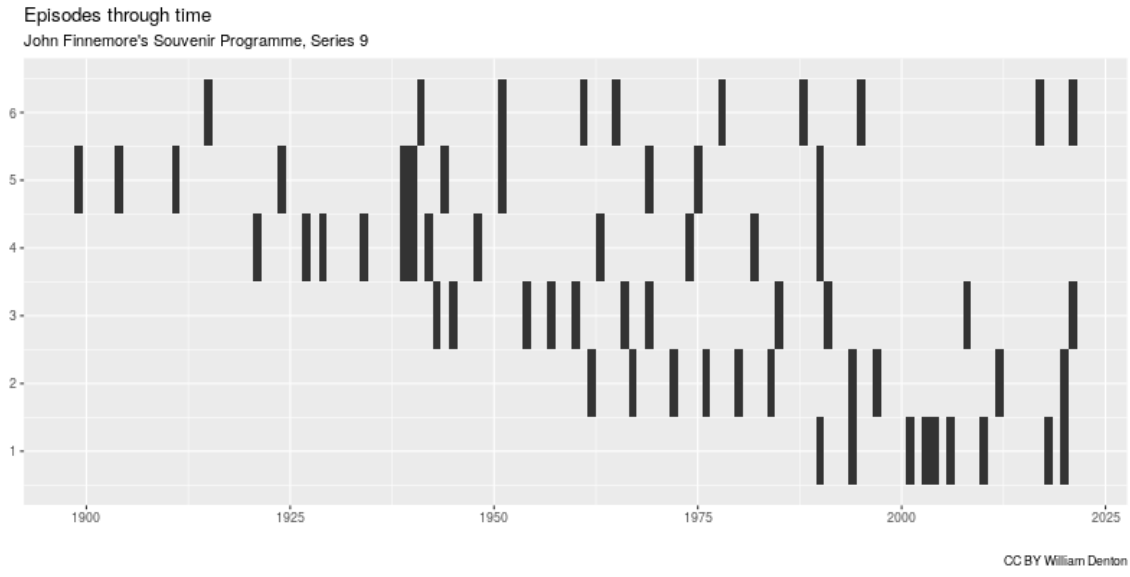
And now it's ready for charting. This shows clearly how the first episode is about Russ, the second about Deborah, etc. Note how Russ, Deborah and Jerry are there at the start in the first scene, and then together with Vanessa and Newt in the very last.

And we can track everyone through all the scenes, too. The opening five and closing five are easily seen, as is the mix of events in episode six.



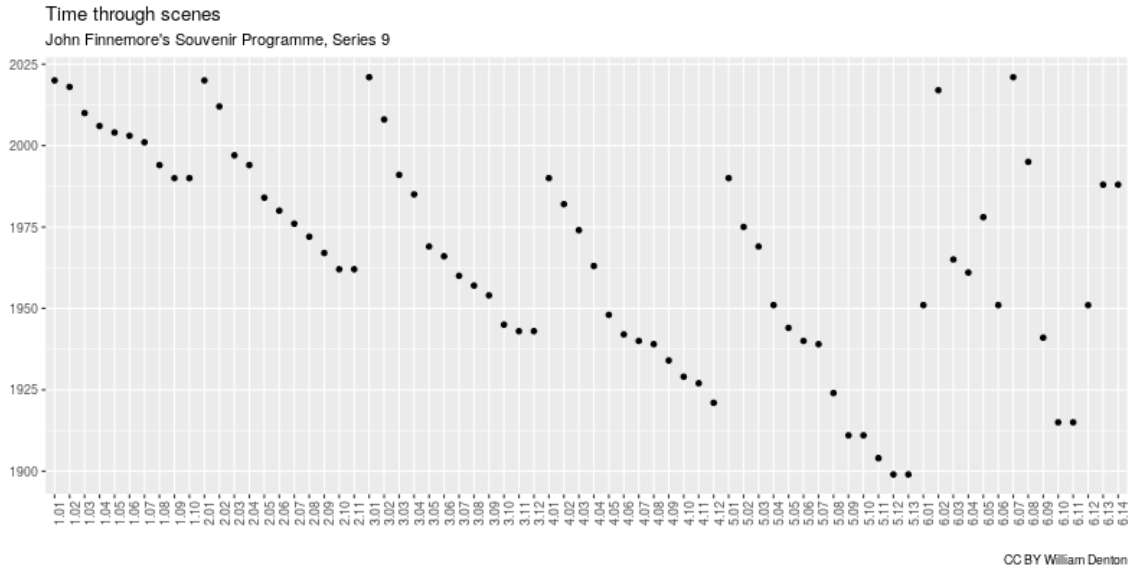
EPISODES AND SCENES IN TIME

Another way of looking at what happens is showing the years in which events happen in the different episodes.



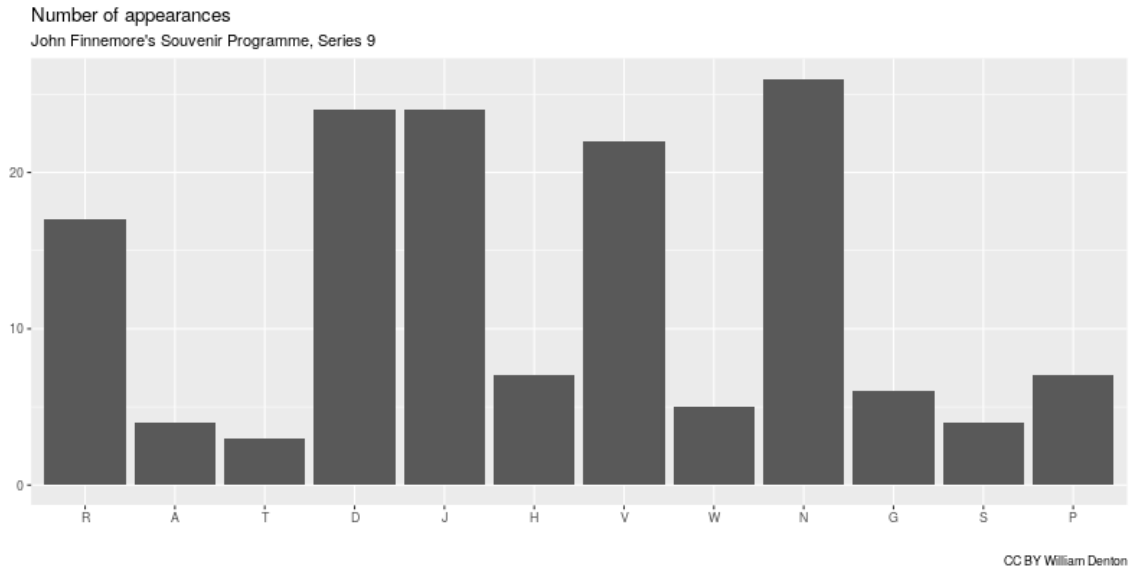
It's more interesting to see how time moves through the scenes within the episodes, though. This

shows how the scenes in each episode move back in time, and the episodes move back in time, until the last episode, which jumps all over.



HOW OFTEN PEOPLE APPEAR

This is just a straightforward count of how many scenes everyone is in.



HOW PEOPLE OCCUR TOGETHER

We can also look at how people are together in scenes.

We can make a raw count of how often people appear together. Using `crossprod` it's easy enough to make a symmetric count.

```
crossprod_all <- people_and_eps %>% select(-x) %>% table() %>% crossprod()
diag(crossprod_all) <- 0
crossprod_all
```

```
      person
person R A T  D  J H V W N G S P
     R 0 4 3  9  4 0 2 0 4 0 0 0
     A 4 0 3  1  1 0 0 0 0 0 0 0
     T 3 3 0  1  1 0 0 0 0 0 0 0
     D 9 1 1  0 12 5 5 2 8 0 0 0
     J 4 1 1 12  0 7 7 2 7 0 0 0
     H 0 0 0  5  7 0 1 1 1 0 0 0
     V 2 0 0  5  7 1 0 5 7 1 2 0
     W 0 0 0  2  2 1 5 0 1 0 0 0
     N 4 0 0  8  7 1 7 1 0 5 2 4
     G 0 0 0  0  0 0 1 0 5 0 1 3
     S 0 0 0  0  0 0 2 0 2 1 0 0
     P 0 0 0  0  0 0 0 0 4 3 0 0
```

But it's more helpful to make a relative one-way measure of how connected each person is to every other person. Take Hilla, for example. We can look at every scene she's in and count up how many times other people appear in scenes with her.

```
hilla_eps <- people_and_eps %>% filter(person == "H") %>% pull(ep)
people_and_eps %>%
    filter(ep %in% hilla_eps, person != "H") %>%
    count(person)
```

person	n
D	5
J	7
V	1
W	1
N	1

So of all her appearances, she's with Jerry 7 times, Deborah 5 times, and Vanessa, Walter and Newt once each (at the Christmas scene). We can turn this into a relative measure of how connected Hilla is to all these people, by dividing these numbers by their total.

```
people_and_eps %>%
    filter(ep %in% hilla_eps, person != "H") %>%
    count(person) %>%
    mutate(ratio = 100 * round(n / sum(n), 2))
```

person	n	ratio
D	5	33
J	7	47
V	1	7
W	1	7
N	1	7

Hilla is 47% connected to Jerry, 33% connected to Deborah, and 7% to the others. How does this compare to Jerry, who is in scenes before and after his marriage to Hilla?

```
jerry_eps <- people_and_eps %>% filter(person == "J") %>% pull(ep)
people_and_eps %>%
    filter(ep %in% jerry_eps, person != "J") %>%
    count(person) %>%
    mutate(ratio = 100 * round(n / sum(n), 2))
```

person	n	ratio
R	4	10
A	1	2
T	1	2
D	12	29
H	7	17
V	7	17
W	2	5
N	7	17

Jerry is 29% connected to Deborah and 17% to Hilla, Vanessa and Uncle Newt.

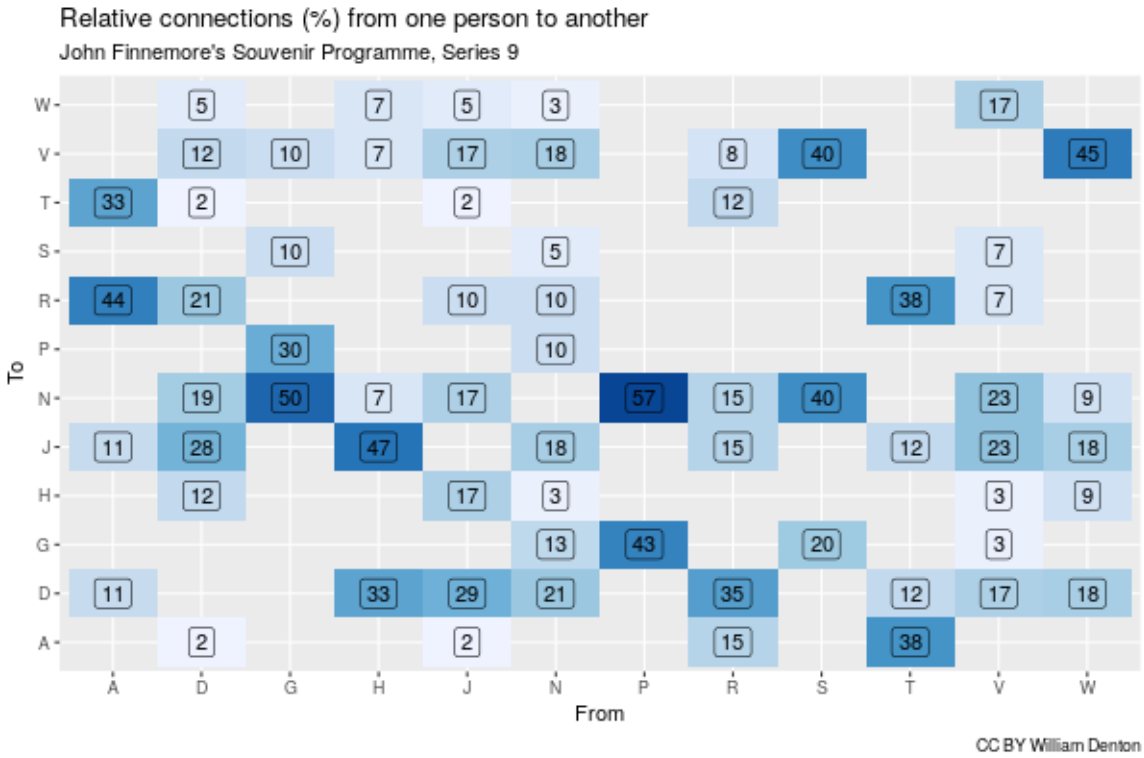
We can produce this table for everyone, like so:

```
rel_connections <- crossprod_all %>%
    as_tibble() %>%
    mutate(total = rowSums(.)) %>%
    mutate(across(R:P, ~ 100 * round(.x / total, 2))) %>%
    bind_cols(from = all_people) %>%
    select(from, R:P)
rel_connections
```

from	R	A	T	D	J	H	V	W	N	G	S	P
R	0	15	12	35	15	0	8	0	15	0	0	0
A	44	0	33	11	11	0	0	0	0	0	0	0
T	38	38	0	12	12	0	0	0	0	0	0	0
D	21	2	2	0	28	12	12	5	19	0	0	0
J	10	2	2	29	0	17	17	5	17	0	0	0
H	0	0	0	33	47	0	7	7	7	0	0	0
V	7	0	0	17	23	3	0	17	23	3	7	0
W	0	0	0	18	18	9	45	0	9	0	0	0
N	10	0	0	21	18	3	18	3	0	13	5	10
G	0	0	0	0	0	0	10	0	50	0	10	30
S	0	0	0	0	0	0	40	0	40	20	0	0
P	0	0	0	0	0	0	0	0	57	43	0	0

And we can chart this (after turning it from wide data into long data):

```
rel_connections %>%
    pivot_longer(!from, names_to = "to", values_to = "ratio") %>%
    filter(ratio > 0) %>%
    ggplot(aes(x = from, y = to, fill = ratio, label = ratio)) +
    geom_tile(show.legend = FALSE) +
    geom_label(show.legend = FALSE) +
    scale_fill_distiller(palette = "Blues", direction = 1) +
    labs(title = "Relative connections (%) from one person to another",
         subtitle = "John Finnemore's Souvenir Programme, Series 9",
         x = "From", y = "To",
         caption = "CC BY William Denton")
```



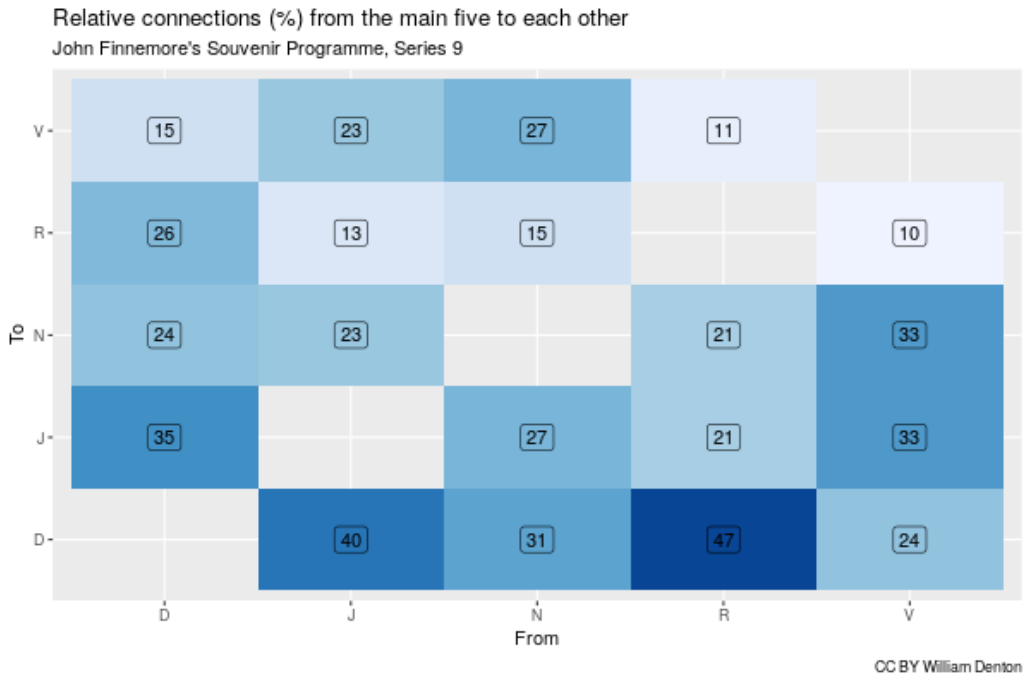
Read this by starting on the x axis and finding a person then going up the y axis to find out how much that person is connected to another person. Looking up in the P (Patrick) column we see he's 43% connected to Gally and 57% to Newt (which reflects the one scene where he's with Newt but not Gally). Jerry and Vanessa are connected to many people, but Newt the most, since he's met everyone except Toby and Alex.

We can do this just for the core five, too.

```
crossprod_five <- people_and_eps %>%
    select(-x) %>%
    filter(person %in% five_people) %>%
    mutate(person = as.character(person)) %>%
    table() %>%
    crossprod()
diag(crossprod_five) <- 0
rel_connections_five <- crossprod_five %>%
    as_tibble() %>%
    mutate(total = rowSums(.)) %>%
    mutate(across(c(five_people), ~ 100 * round(.x / total, 2))) %>%
    bind_cols(from = colnames(crossprod_five)[1:5]) %>%
    select(-total)
```

Here the ordering of the names in the chart is different, and I can't get it back into the order I want, but so be it.

```
rel_connections_five %>%
    pivot_longer(!from, names_to = "to", values_to = "ratio") %>%
    filter(ratio > 0) %>%
    ggplot(aes(x = from, y = to, fill = ratio, label = ratio)) +
    geom_tile(show.legend = FALSE) +
    geom_label(show.legend = FALSE) +
    scale_fill_distiller(palette = "Blues", direction = 1) +
    labs(title = "Relative connections (%) from the main five to each other",
         subtitle = "John Finnemore's Souvenir Programme, Series 9",
         x = "From", y = "To",
         caption = "CC BY William Denton")
```



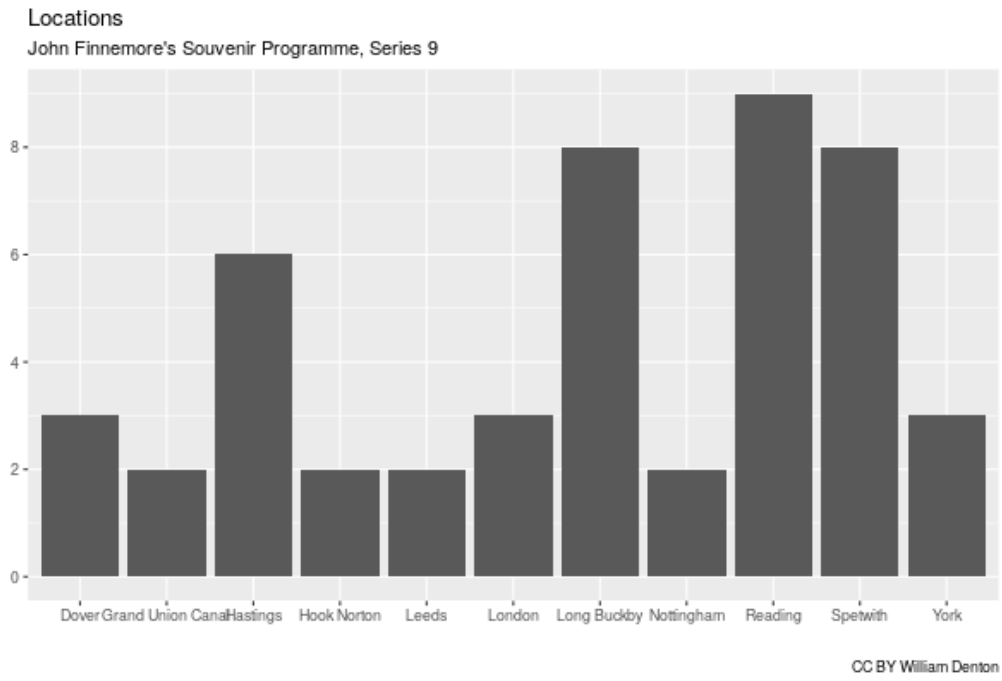
This makes clearer Russ's close connection to Deborah, for example.

LOCATIONS

Finally, it's straightforward to tally all of the locations used. Some are only used once, so this chart just shows the ones used at least twice.

```
locations <- j %>% count(Location) %>% filter(n > 1) %>% pull(Location)

j %>% count(Location) %>% filter(Location %in% locations) %>%
    ggplot(aes(x = Location, y = n)) +
    geom_col() +
    labs(title = "Locations",
         subtitle = "John Finnemore's Souvenir Programme, Series 9",
         x = "", y = "",
         caption = "CC BY William Denton") +
    scale_y_continuous(breaks = pretty_breaks())
```


Locations
John Finnemore's Souvenir Programme, Series 9
8
6
4
2
0
Dover
Grand Union Canal
Hastings
Hook Norton
Leeds
London
Long Buckby
Nottingham
Reading
Spetwith
York
CC BY William Denton