Class Notes

Tim Riffe

July 5, 2019

First merge HMD and HFD

This is a left join, and I figured out which merge function to use by referring to the `dplyr` cheat cheet tiny.cc/37798y and looking at the diagrams on that cheat sheet. Then the final filtering happened basically using demographic logic and not particularly using merge functionality.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.2.1 --

## v ggplot2 3.1.0       v purrr   0.3.2
## v tibble  2.1.1       v dplyr   0.8.0.1
## v tidyr   0.8.3       v stringr 1.4.0
## v readr   1.3.1       v forcats 0.4.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(here)
```

```
## here() starts at /home/tim/workspace/BSSD/BSSD2019
```

```
HMD <- readRDS(here("03_Wednesday", "HMD.rds")) %>%
    filter(Sex == "f")

# now let's get the HFD
HFD <- readRDS("HFD.rds")

# now merge
Stable <- left_join(HMD,
                    HFD,
                    by = c("Country","Year","Age")) %>%           group_by(Country, Year) %>%
          mutate(ASFR = replace_na(ASFR, 0),
                 TFR = sum(ASFR)) %>%
          ungroup() %>%
          filter(TFR > 0)
```

Now we want to merge on the SRB. Guess what, if you want info on Male and Female births you look to HMD not HFD. Weird huh? First calculate:

```
# Get info we need about SRB
B <- readRDS("B.rds") %>%
    mutate(SRB = Male / Female,
           PF = Female / Total) %>%
    select(Country, Year, SRB, PF)
```

Now merge:

```
Stable <- left_join(Stable,
                    B,
                    by = c("Country", "Year"))
```

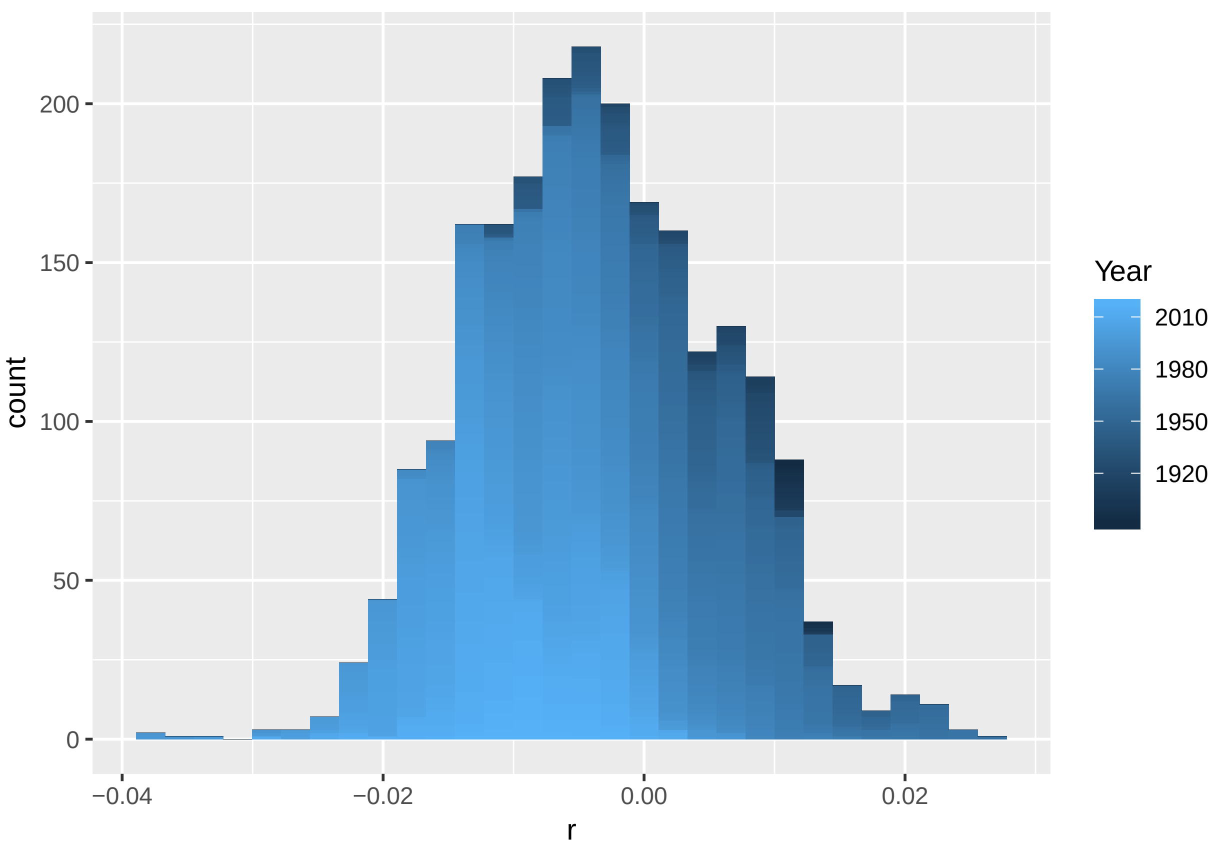
Now for the sake of getting Lx, we're going to add on all the lifetable columns, using our Wednesday functions, sitting in the file `LifeTableFunctions.R`

```
source(here("03_Wednesday","LifeTableFunctions.R"))

Stable <- Stable %>%
          group_by(Country, Year) %>%
          my_lt3()
```

Now we're ready to calculate r, but how do we want to do this?

```
Stable %>%
    group_by(Country, Year) %>%
    mutate(fxf = ASFR * PF) %>%
    summarize(r = Coales_r(Lx, fxf, Age)) %>%
    ggplot(mapping = aes(x = r,
                         group = Year,
                         fill = Year)) +
    geom_histogram()
```

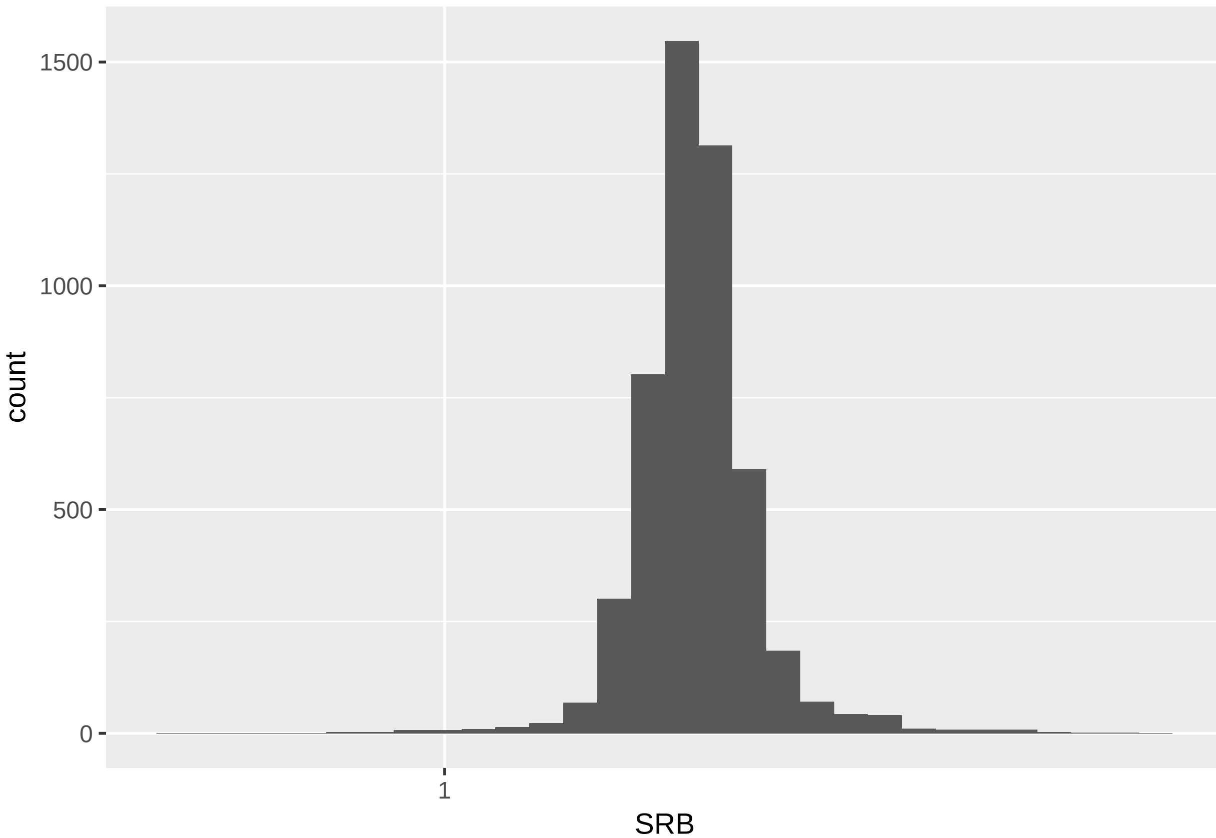
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Aside: yes, SRB varies quite a bit:

```r
readRDS("B.rds") %>% mutate(SRB = Male / Female) %>%
    ggplot(mapping = aes(x = SRB)) +
    geom_histogram() + scale_x_log10()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## Warning: Removed 179 rows containing non-finite values (stat_bin).
```



Free exercises:

Here are some variables, some with definitions, possibly new to you, some devised on the spot. Errors fixed along the way, in part along with calibration of calculations.

1. TFR
$$TFR = \sum ASFR(x)$$
2. MAB
$$MAB = \frac{\sum xASFR(x)}{TFR}$$
3. T (mean generation length in the stable population)
$$T = \frac{log(R(0))}{r}$$
4. $R(0)$ net reproductive ratio (number of daughters per mother, discounted for mortality and SRB), assumes $L(x)$ has radix $= 1$, i.e. that $l(0) = 1$
$$R(0) = \sum ASFR(x)^f L(x)$$

5. $e(0)^\dagger$ average years of life lost in the stationary population. Assumes $1 = \sum d(x)$

$$e(0)^\dagger = \sum d(x)e(x)$$

6. net ASFR, often call $\phi(x)$, i.e. ASFR discounted for mortality:

$$\phi(x) = ASFR(x)^f L(x)$$

7. net TFR

$$netTFR = \sum L(x)ASFR(x)$$

8. Replacement TFR, $TFR^\star$, i.e. the quantity that everyone thinks equals 2.1. What is it really? A couple steps. Here derived logically: in the stationary population, $l(0)$ is the number of births. Since, in our version of a stable population the lifetable pertains only to women, we need to produce $l(0) * (SRB + 1)$ births per year in order to acheive stationarity. The number of births we're actually getting is $netTFR$, so we can get a correction factor, k:

$$k = \frac{SRB + 1}{netTFR}$$

Then replacement fertility, $TFR^\star$ must be equal to:

$$TFR^\star = k * TFR$$

I wonder what it looks like really? Is it higher than 2.1, or lower? Is 2.1 old fashioned?

Some possible calcs + plots

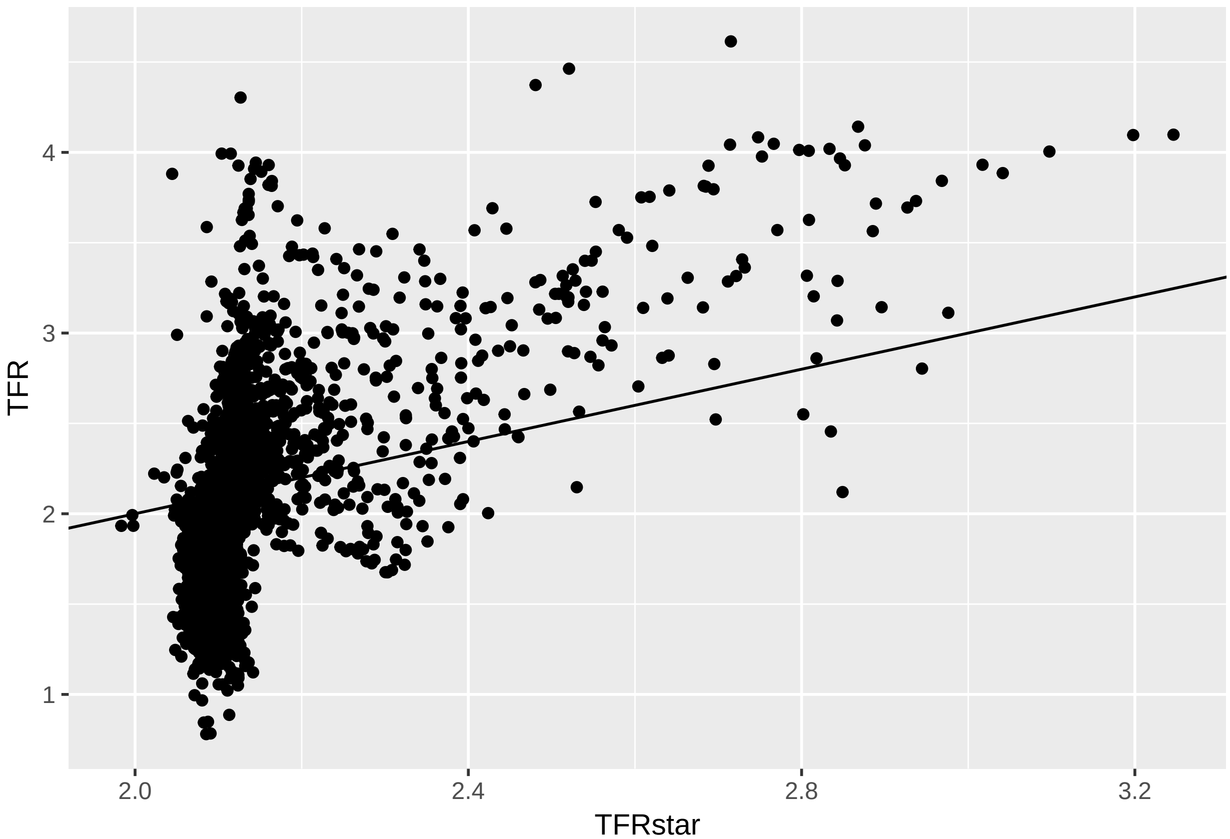
How about $TFR^\star$ by ...?

```
# this could be cleaner
Stable2 <- Stable %>%
    group_by(Country, Year) %>%
    mutate(
        fxf = ASFR * PF,    # asfr only females
        phi = fxf * Lx,     # net maternity rates
        R0 = sum(phi),      # NRR net reproductive ratio
        r = Coales_r(Lx, fxf, Age), # intrinsic
        TFR = sum(ASFR),
        netTFR = sum(ASFR * Lx)
        ) %>%
    filter(Age == 0) %>%
    mutate(
        k = (1 + SRB) / netTFR,
        TFRstar = TFR * k,
        TT = log(R0) / r)
```

Below the line is below replacement, above it is above replacement.

```
ggplot(Stable2, mapping = aes(x = TFR, y = TFRstar)) +
    geom_point() + geom_abline(slope=1,intercept=0) +
    coord_fixed() + coord_flip()
```

```
## Coordinate system already present. Adding new coordinate system, which will replace the existing one
```



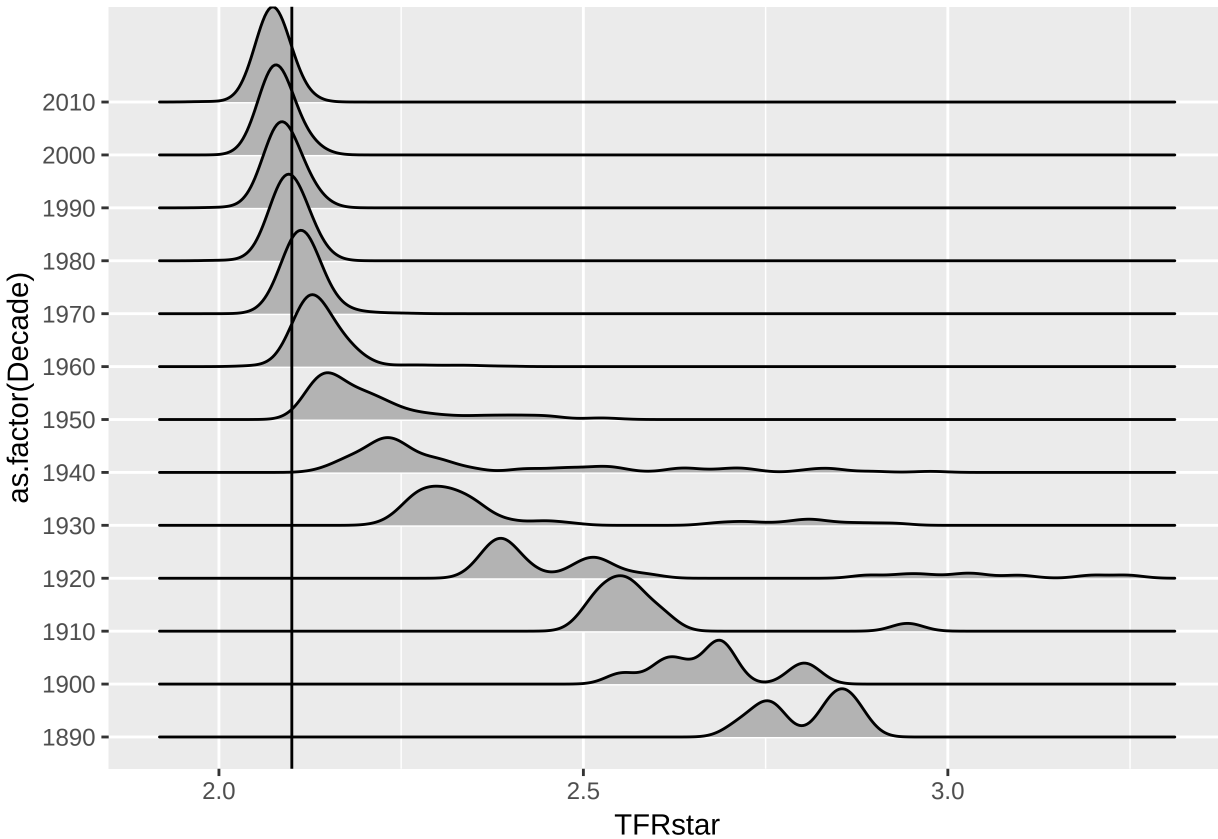
And a final plot in a bit of a hurry, this one was particularly cooperative in its construction. Replacement fertility by decades. Moral of the story holds today though, since the movement is due to mortality: high mortality implies higher TFR needed.

```
library(ggridges)
```

```
##
## Attaching package: 'ggridges'

## The following object is masked from 'package:ggplot2':
##
##     scale_discrete_manual
```

```
Stable2 %>% mutate(Decade = Year - Year %% 10) %>%
ggplot(mapping = aes(x = TFRstar,
                     y = as.factor(Decade))) +
    geom_density_ridges() +
    geom_vline(xintercept=2.1)
```

```
## Picking joint bandwidth of 0.0217
```



And right about here I talked a bunch about motivating work with a clear picture, for purposes of science communication, and gave examples: Oeppen-Vaupel, Mikko, Sander, Kashnitsky :-) And I mentioned some discipline journals that have data viz manuscript types, as well as the above high impact journals where it has worked before for demographers.

Some quantities by age:

(we didn't do anything here together) 1. The stationary age strucure is just $L(x)$ 2. The stable age structure, $c(x)$ is:

$$c(x) = \frac{L(x)e^{-rx}}{\sum L(x)e^{-rx}}$$

3. so the mean age of the stationary and stable populations are:

$$A = \frac{\sum xL(x)}{e(0)}$$

$$A^{\star} = \frac{\sum xc(x)}{\sum c(x)}$$

And of course, wouldn't we like to know how $A^{\star}$ changes with r ? Maybe turn it into an elasticity problem?

4. Some people (not me) would take this as an opportunity to introduce dependency ratios, like OADR, the old-age dependency ratio. But I think it's too ill-formed to be useful anymore, so I won't give formulas for that. But there's this not-so-bad alternative from Scherbov & Sanderson, the so-called *prospective* old-age dependency ratio, which is at least benchmarked to mortality conditions.

$$POARD = \frac{\sum P(x|x > x\star)}{P}$$

where:

$$x^\star = x \sim e(x) = 15$$

Which is maybe not the best notation, but the idea is to find x where $e(x) = 15$, and then just sum the population up from there. At least is moves with mortality.

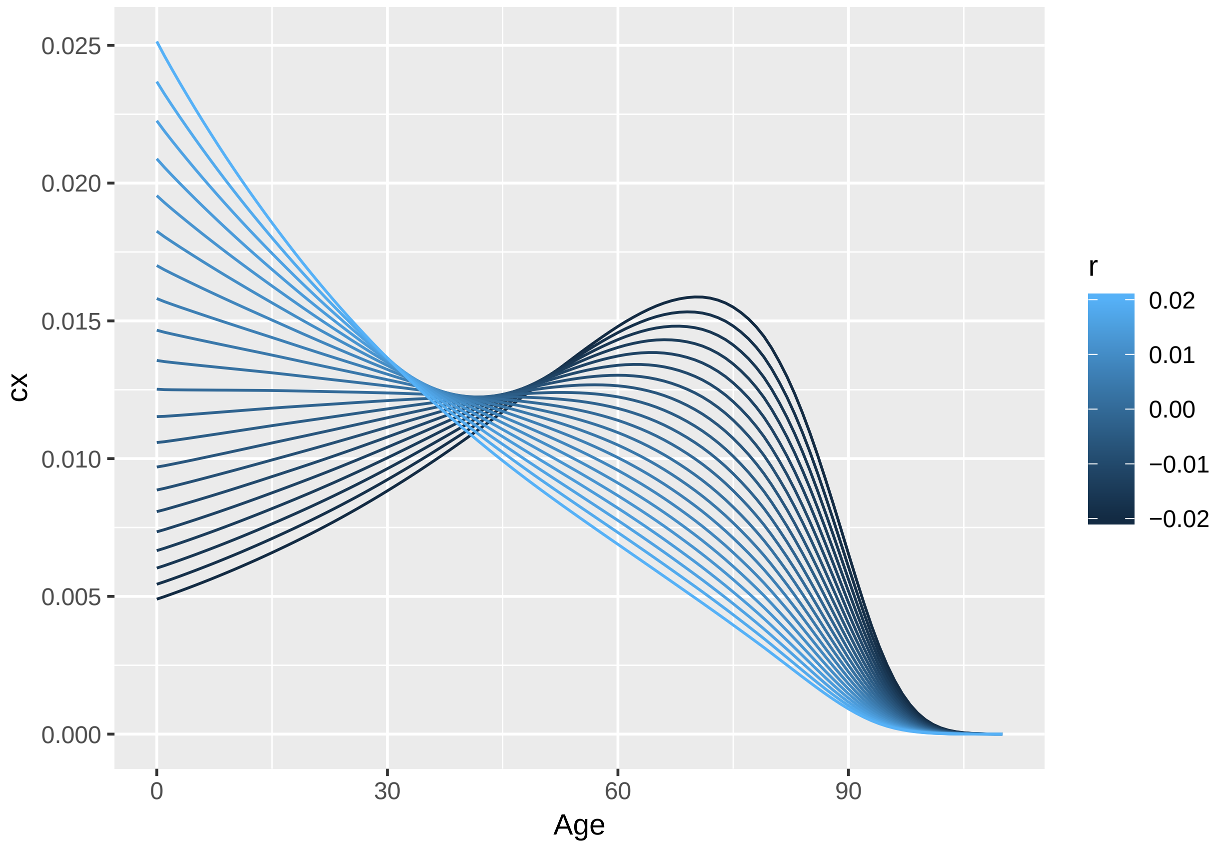
Intuition break: stable age structure

This was just for the sake of the picture: we didn't talk about this code really

```
Lx <- Stable %>%
    filter(Country == "USA" &
            Year == 2000) %>%
    pull(Lx)

r <- seq(-.02, .02, length = 21)
Lxr_to_cx <- function(Lx, r, x){
    CX <- Lx * exp(-r * x)
    CX / sum(CX)
}

StablePop <- matrix(nrow = 111,
                    ncol = 21,
                    dimnames = list(Age = 0:110, r = r))
for (i in 1:21){
    StablePop[,i] <- Lxr_to_cx(Lx, r[i], x = 0:110)
}

library(reshape2)
```

```
##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
melt(StablePop, varnames = c("Age","r"),
          value.name = "cx") %>%
    ggplot(mapping = aes(x = Age,
                         y = cx,
                         group = r,
                         color = r)) +
    geom_line()
```


0.025
0.020
0.015
0.010
0.005
0.000
cx
0
30
60
90
Age
r
0.02
0.01
0.00
−0.01
−0.02