

# Growth rates of PIT data from the Delaware River

Ben Letcher

2022-09-23

```
d <- tar_read(target_d)
eh <- tar_read(target_eh)
```

Add growth rate

```
d <- d %>%
  mutate(year = as.numeric(substr(dateYM, 1,4)),
         month = as.numeric(substr(dateYM, 6,7)))

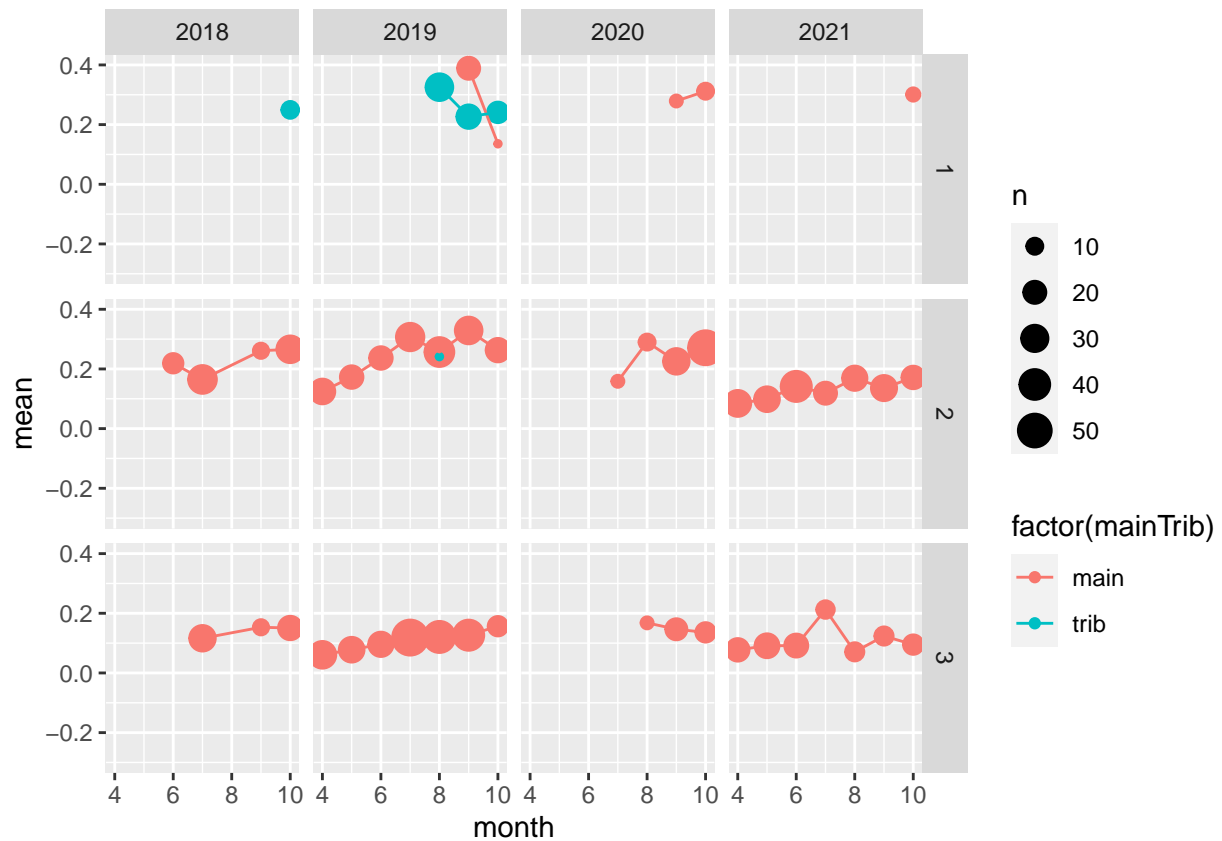
d <- d %>%
  group_by(tag) %>%
  arrange(tag, date) %>%
  mutate(LengthLag = lag(Length),
         dateLag = lag(date),
         dateDiff = as.numeric(date - dateLag),
         growth = (Length - LengthLag) / dateDiff)
```

```
meanGrowth <- d %>%
  filter(dateDiff < 300, growth > 0, growth < 3) %>%
  group_by(mainTrib, year, month, sizeState) %>%
  summarise(mean = mean(growth, na.rm = TRUE),
            n = n()) %>%
  filter(n > 3)

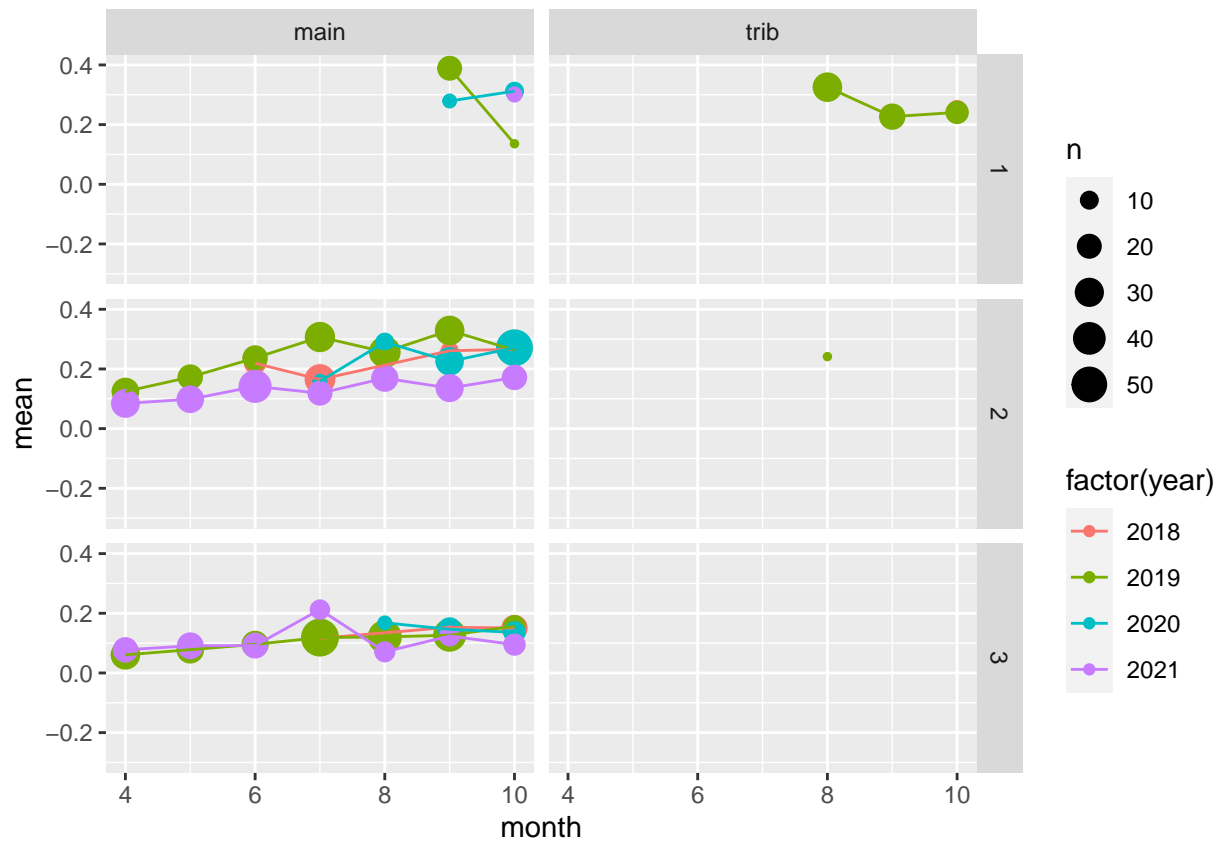
#> `summarise()` has grouped output by 'mainTrib', 'year', 'month'. You can
#> override using the `.groups` argument.

ggplot(meanGrowth, aes(month, mean, color = factor(mainTrib))) +
  geom_point(aes(size = n)) +
  geom_line() +
  ylim(-0.3, 0.4) +
  facet_grid(sizeState ~ year)

#> Warning: Removed 3 rows containing missing values (geom_point).
#> Warning: Removed 3 row(s) containing missing values (geom_path).
#> geom_path: Each group consists of only one observation. Do you need to adjust
#> the group aesthetic?
#> geom_path: Each group consists of only one observation. Do you need to adjust
#> the group aesthetic?
```

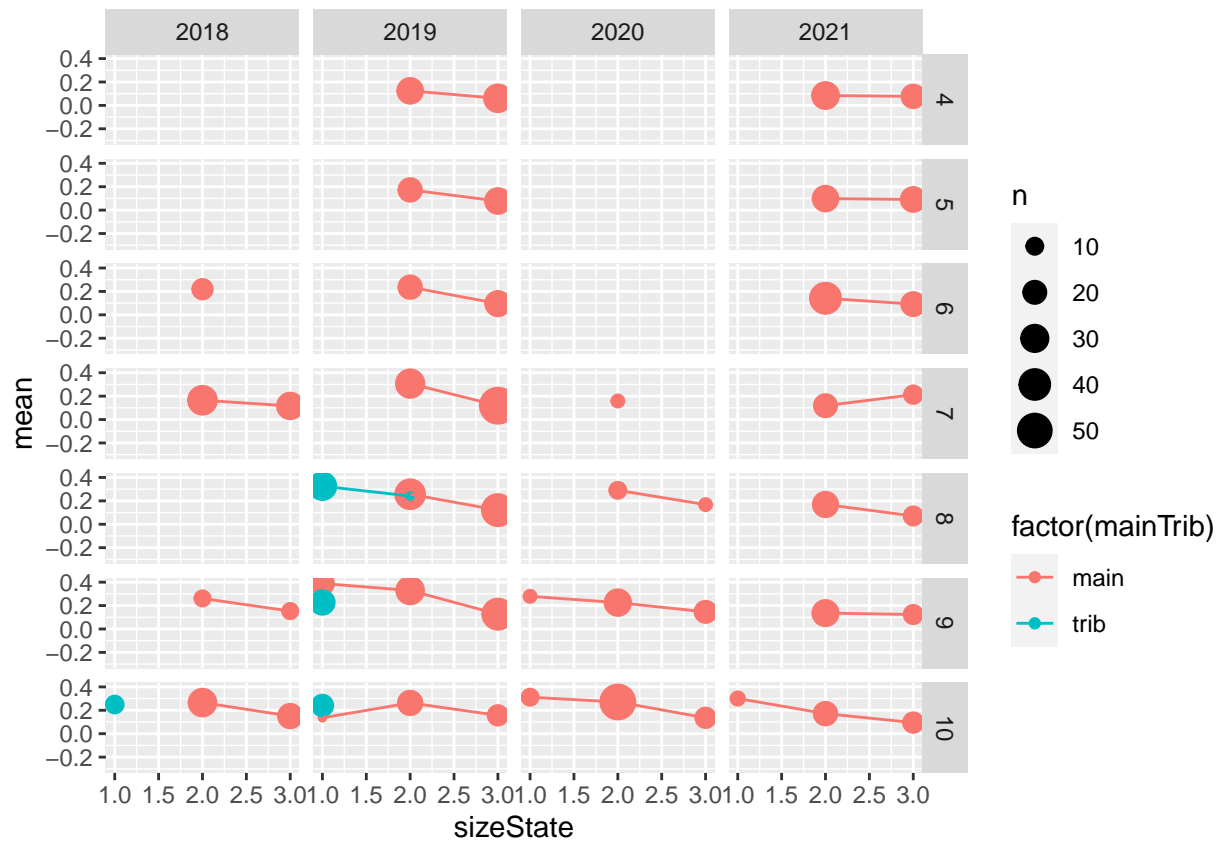


```
ggplot(meanGrowth, aes(month, mean, color = factor(year))) +
  geom_point(aes(size = n)) +
  geom_line() +
  ylim(-0.3, 0.4) +
  facet_grid(sizeState ~ mainTrib)
#> Warning: Removed 3 rows containing missing values (geom_point).
#> Warning: Removed 2 row(s) containing missing values (geom_path).
```



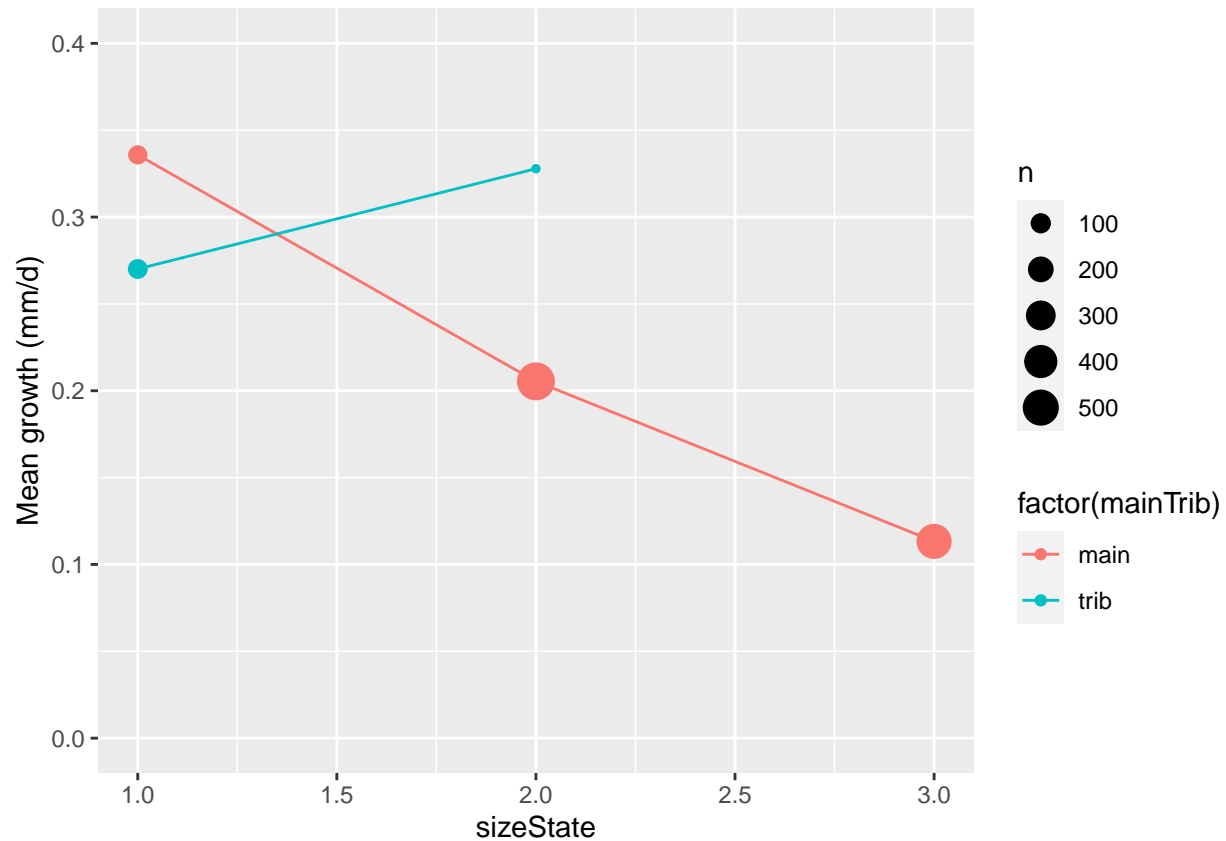
```
ggplot(meanGrowth, aes(sizeState, mean, color = factor(mainTrib))) +
  geom_point(aes(size = n)) +
  geom_line() +
  ylim(-0.3, 0.4) +
  facet_grid(month ~ year)
```

#> Warning: Removed 3 rows containing missing values (geom\_point).  
 #> geom\_path: Each group consists of only one observation. Do you need to adjust  
 #> the group aesthetic?  
 #> geom\_path: Each group consists of only one observation. Do you need to adjust  
 #> the group aesthetic?



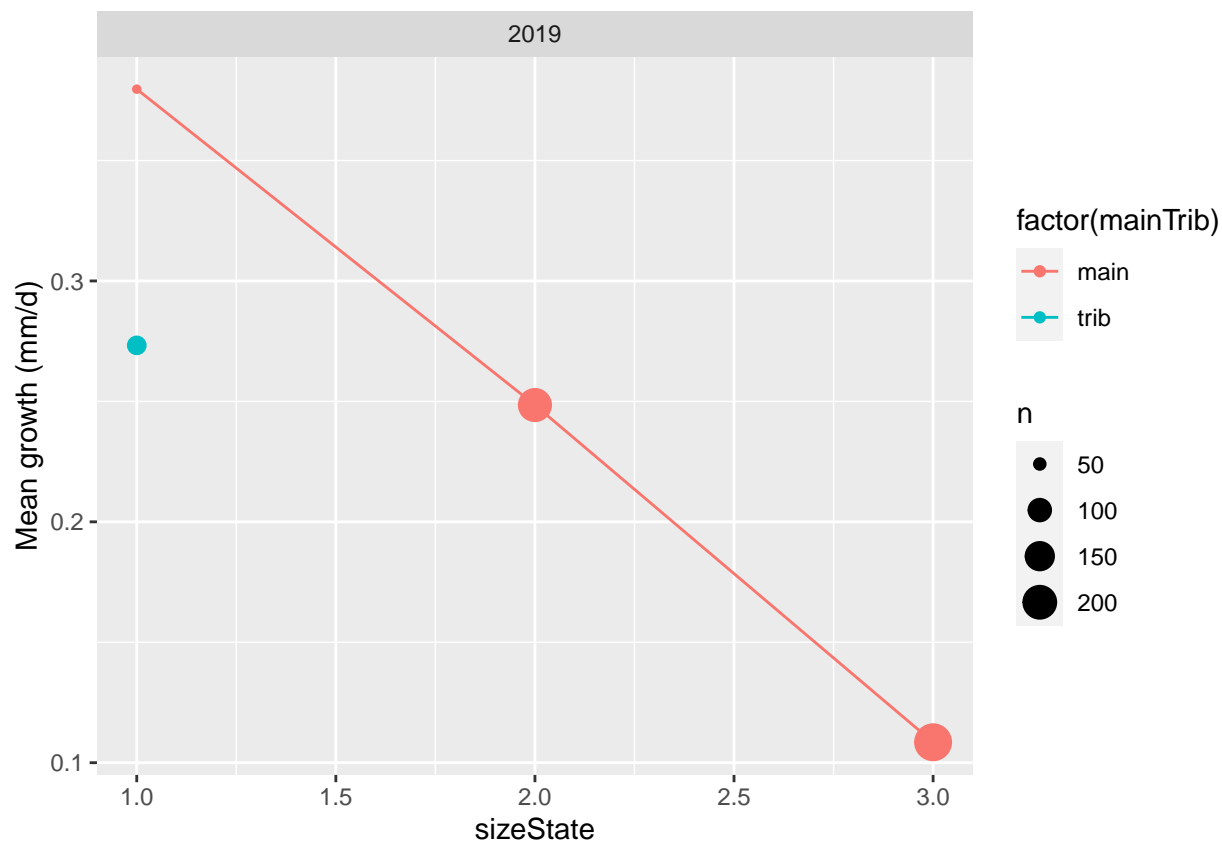
```
meanGrowthSize <- d %>%
  filter(dateDiff < 300, growth > 0, growth < 3) %>%
  group_by(mainTrib, sizeState) %>%
  summarise(mean = mean(growth, na.rm = TRUE),
            sd = sd(growth, na.rm = TRUE),
            n = n()) %>%
  filter(n > 5)
#> `summarise()` has grouped output by 'mainTrib'. You can override using the
#> `.groups` argument.

ggplot(meanGrowthSize, aes(sizeState, mean, color = factor(mainTrib))) +
  geom_point(aes(size = n)) +
  ylab("Mean growth (mm/d)") +
  ylim(0, 0.4) +
  geom_line()
```



```
meanGrowthSizeYear <- d %>%
  filter(dateDiff < 300, growth > 0, growth < 3) %>%
  group_by(mainTrib, sizeState, year) %>%
  summarise(mean = mean(growth, na.rm = TRUE),
            sd = sd(growth, na.rm = TRUE),
            n = n()) %>%
  filter(n > 5)
#> `summarise()` has grouped output by 'mainTrib', 'sizeState'. You can override
#> using the `.groups` argument.
```

```
ggplot(meanGrowthSizeYear %>% filter(year == 2019, n > 20), aes(sizeState, mean, color = factor(mainTrib))) +
  geom_point(aes(size = n)) +
  ylab("Mean growth (mm/d)") +
  #ylim(0, 0.4) +
  geom_line() +
  facet_wrap(~year)
```



```
tmp <- d%>% filter(year == 2020, mainTrib == "main", sizeState == 2)
table(tmp$growth)
```

```
#>
#> -3.37931034482759 -1.24074074074074 -0.551724137931034 -0.333333333333333
#> 1 1 1 1
#> -0.285714285714286 -0.208333333333333 -0.19047619047619 -0.1875
#> 1 1 2 1
#> -0.166666666666667 -0.157894736842105 -0.121212121212121 -0.12
#> 1 1 1 1
#> -0.0952380952380952 -0.0833333333333333 -0.0476190476190476 -0.0357142857142857
#> 1 1 2 1
#> 0 0.00699300699300699 0.0138888888888889 0.0285714285714286
#> 2 1 1 1
#> 0.0287141073657928 0.0303030303030303 0.0337078651685393 0.0350877192982456
#> 1 1 1 1
#> 0.0363128491620112 0.0383480825958702 0.0416666666666667 0.0476190476190476
#> 1 1 1 1
#> 0.0535714285714286 0.0606060606060606 0.0645161290322581 0.0663983903420523
#> 1 1 1 1
#> 0.0689655172413793 0.0695970695970696 0.0701754385964912 0.0705882352941176
#> 3 1 1 1
#> 0.0829145728643216 0.0833333333333333 0.0855263157894737 0.0892857142857143
#> 1 2 1 2
#> 0.0898876404494382 0.0909090909090909 0.0973451327433628 0.107142857142857
#> 1 1 1 1
```

#>	0.112956810631229	0.114942528735632	0.117136659436009	0.121212121212121
#>	1	1	1	1
#>	0.123893805309735	0.124050632911392	0.124309392265193	0.125
#>	1	1	1	2
#>	0.129032258064516	0.131578947368421	0.132307692307692	0.135384615384615
#>	1	1	1	1
#>	0.136456211812627	0.136531365313653	0.137426900584795	0.137931034482759
#>	1	1	1	2
#>	0.140672782874618	0.141263940520446	0.142857142857143	0.143292682926829
#>	1	1	1	1
#>	0.145945945945946	0.146198830409357	0.149122807017544	0.150442477876106
#>	1	1	1	1
#>	0.151515151515152	0.152941176470588	0.154761904761905	0.155555555555556
#>	1	1	1	1
#>	0.157458563535912	0.157894736842105	0.158357771260997	0.16
#>	1	1	1	1
#>	0.160714285714286	0.160756501182033	0.161290322580645	0.163398692810458
#>	2	1	1	1
#>	0.166666666666667	0.167088607594937	0.170575692963753	0.172413793103448
#>	1	1	1	2
#>	0.175438596491228	0.178117048346056	0.178438661710037	0.178571428571429
#>	1	1	1	2
#>	0.18052738336714	0.182242990654206	0.182890855457227	0.18450184501845
#>	1	1	1	1
#>	0.187683284457478	0.191374663072776	0.194373401534527	0.194594594594595
#>	1	1	1	1
#>	0.198473282442748	0.203125	0.204545454545455	0.206896551724138
#>	1	1	1	1
#>	0.210526315789474	0.212121212121212	0.214285714285714	0.215102974828375
#>	1	1	1	1
#>	0.215189873417722	0.218954248366013	0.219941348973607	0.222222222222222
#>	1	1	1	1
#>	0.223529411764706	0.228070175438596	0.228571428571429	0.228668941979522
#>	1	1	1	1
#>	0.229281767955801	0.230088495575221	0.23155737704918	0.232142857142857
#>	1	1	1	1
#>	0.234604105571848	0.235135135135135	0.241379310344828	0.241448692152918
#>	1	1	2	1
#>	0.241635687732342	0.242424242424242	0.25	0.255372945638432
#>	1	2	3	1
#>	0.256130790190736	0.258064516129032	0.258169934640523	0.265625
#>	1	2	1	1
#>	0.267857142857143	0.270516717325228	0.272264631043257	0.272727272727273
#>	2	1	1	1
#>	0.274285714285714	0.275862068965517	0.278	0.280701754385965
#>	1	1	1	1
#>	0.281767955801105	0.283582089552239	0.28997867803838	0.290780141843972
#>	1	1	1	1
#>	0.291666666666667	0.294117647058824	0.298245614035088	0.303030303030303
#>	1	1	1	1
#>	0.310344827586207	0.321663019693654	0.32258064516129	0.328125
#>	2	1	1	1
#>	0.339285714285714	0.342857142857143	0.376470588235294	0.387096774193548

#>	1	1	1	1
#>	0.4	0.40625	0.417366946778712	0.436781609195402
#>	2	1	1	1
#>	0.448275862068966	0.451612903225806	0.482758620689655	0.514285714285714
#>	1	2	1	1
#>	0.517241379310345	0.529411764705882	0.53125	0.586206896551724
#>	1	1	1	1
#>	1.07142857142857	2.20689655172414	52	
#>	1	1	1	