Plant Tillering - Fibonacci Numbers

Gunther Krauss

January 11, 2022

This handout is a supplement to the presentation of the first modeling exercise. You can use it if you want to type and run the R code that was presented during the exercise.

1 Calculating number of tillers

1.1 Calculation - first approach

Remind the formula

$$t_1 = 1, \quad t_2 = 1, \quad t_n = t_{n-1} + t_{n-2}$$

We use R like a simple calculator

```
t1 <- 1
t2 <- 1
t3 <- t2 + t1
t3
```

```
# [1] 2
```

```
t4 <- t3 + t2
t4
```

```
# [1] 3
```

```
t5 <- t4 + t3
t6 <- t5 + t4
t7 <- t6 + t5
t8 <- t7 + t6
t8
```

```
# [1] 21
```

What is the value of t12? And t50???

1.2 Using the for-loop

We use the for-loop to calculate the numbers of new tillers up to stage 50.

```
m <- 50
t <- numeric(m)
t[1] <- 1
t[2] <- 1
for(n in 3:m)
{
  t[n] <- t[n-1] + t[n-2]
}
```

1.3 Displaying the number of new tillers

```
t
```

```
#  [1]          1           1          2          3          5          8
#  [7]         13          21         34         55         89        144
# [13]        233         377        610        987       1597       2584
# [19]       4181        6765      10946      17711      28657      46368
# [25]      75025      121393     196418     317811     514229     832040
# [31]    1346269     2178309    3524578    5702887    9227465   14930352
# [37]   24157817    39088169   63245986  102334155  165580141  267914296
# [43]  433494437   701408733 1134903170 1836311903 2971215073 4807526976
# [49] 7778742049 12586269025
```

1.4 Total number of tillers

Total number of tillers of a stage is the sum of the actual stage tillers and previous stages tillers.

```
tt2 <- t[1] + t[2]
tt3 <- t[1] + t[2] +t[3]
```

Better to use the sum function

```
tt4 <- sum(t[1:4])
tt5 <- sum(t[1:5])
```

1.5 Total number of tillers

How can we do this more efficient? For-loop?

```
tt<- numeric(m)
for(n in 1:m)
{
  tt[n] <- sum(t[1:n])
}
```

We can do it even easier, as R already has the `cumsum` function:

```
tt <- cumsum(t)
```

1.6 Displaying the number of total tillers

```
tt
```

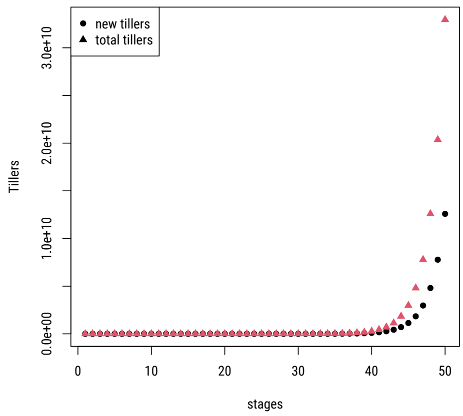
```
#  [1]          1           2          4          7         12         20
#  [7]         33          54         88        143        232        376
# [13]        609         986       1596       2583       4180       6764
# [19]      10945       17710      28656      46367      75024     121392
```

```
# [25]      196417      317810      514228      832039     1346268     2178308
# [31]     3524577     5702886     9227464    14930351    24157816    39088168
# [37]    63245985   102334154   165580140   267914295   433494436   701408732
# [43]  1134903169  1836311902  2971215072  4807526975  7778742048 12586269024
# [49] 20365011073 32951280098
```

1.7 Plot new and total tillers

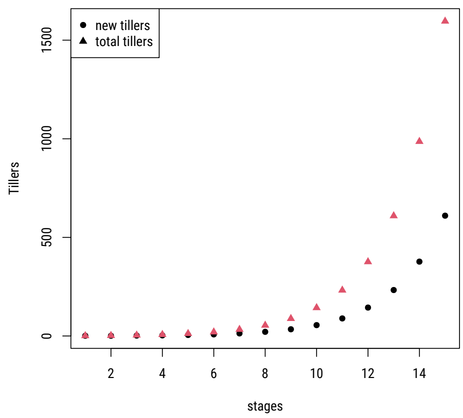
We use the `matplot` function to plot the new and total tillers in a common plot.

```
matplot(cbind(t,tt), pch=c(16,17),
       xlab="stages", ylab="Tillers")
legend("topleft",
       legend=c("new tillers","total tillers"),
       pch=c(16,17))
```



We “zoom in” by plotting only the first 15 stages

```
matplot(cbind(t[1:15],tt[1:15]), pch=c(16,17),
        xlab="stages", ylab="Tillers")
legend("topleft",
       legend=c("new tillers","total tillers"),
       pch=c(16,17))
```



2 Tillers and temperature calculations

2.1 Phyllochron

Let’s take a value of *120 °C day* for the phyllochron. How many days does it take between two stages, if the daily mean temperature is *8 °C*? And when it is *10 °C*? *11 °C*? We calculate it manually:

```
Tphyl <- 120
Tphyl / 8
```

```
# [1] 15
```

```
Tphyl / 10
```

```
# [1] 12
```

```
Tphyl / 11
```

```
# [1] 10.90909
```

As R can do arithmetic calculations with vectors, we can perform it in one calculation step:

```
Temp <- c(8,10,11)
Tphyl/Temp
```

```
# [1] 15.00000 12.00000 10.90909
```

2.2 Calculation of stages - Result

```
Tdmean <- 8
Tphyl <- 120
time <- 100
Tsum <- Tdmean * time
stages <- Tsum/Tphyl
stages
```

```
# [1] 6.666667
```

We need to round the stages to integer numbers. To get completed stages, we always have to take the lower integer (round down).

2.3 Rounding

```
nm <- c(11.7, 14.3)
round(nm)
```

```
# [1] 12 14
```

```
floor(nm)
```

```
# [1] 11 14
```

```
ceiling(nm)
```

```
# [1] 12 15
```

2.4 Calculating number of completed stages

Calculate the stages for all Tdmeans from *6 °C* to *18 °C*.

```
Tphyl <- 120
Tdmean <- 6:18
time <- 100
Tsum  <- Tdmean * time
stages <- floor(Tsum/Tphyl)

data.frame(Tdmean, stages)
```

```
#    Tdmean stages
# 1       6      5
# 2       7      5
# 3       8      6
# 4       9      7
# 5      10      8
# 6      11      9
# 7      12     10
# 8      13     10
# 9      14     11
# 10     15     12
# 11     16     13
# 12     17     14
# 13     18     15
```

2.5 Finally the number of tillers.

Assume the daily mean temperature is *8 °C* and a time period of *100 days*, then with a phyllochron of *120 °C* day, *6* stages are completed. To get the number of new tillers, we take `t[6]`, and the total number of tillers is `tt[6]`.

```
t[6]
```

```
# [1] 8
```

```
tt[6]
```

```
# [1] 20
```

2.6 Number of tillers for all Tdmean

```
Tphyl <- 120
Tdmean <- 6:18
time <- 100
Tsum <- Tdmean * time

stages <- floor(Tsum/Tphyl)

newtil <- t[stages]
newtil
```

```
#  [1]   5   5   8  13  21  34  55  55  89 144 233 377 610
```

```
tottil <- tt[stages]
tottil
```

```
#  [1]   12   12   20   33   54   88  143  143  232  376  609  986 1596
```

2.7 Plot total tillers vs daily means

```
plot(Tdmean, tottil,
     xlab="Daily mean temperature °C",
     ylab="Number of tillers",
     main="Tillers 100 days after emergence")
```