In [1]:

```python
# import
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import os
import openpyxl
```

THIN Guidelines

! since we have no initial stocking data, assume all stands will produce the optimum scenario

Harvest-based Rules: ¶

- if no previous operation and age >= 14, first thin in current year
- if previous first thin and age >= 23, second thin in current year
- if previous first or second thin and age >= 30, bypass second thin
- if previous first thin is later than expected, and gap between 2nd thin at 23 and first thin is less than 5 years, extend second thin an additional 5-9 years beyond 23.
- if previous second thin and gap between final harvest year and year of last thin is >= 13, then 3rd thin 7 years following last thin
- if age >= 35, final harvest in current year

In [2]:

```python
# load dataframe
raw = pd.read_csv('HISTORIC_HARVEST_DATA.csv')
raw.fillna(0, inplace=True)
raw.head()
```

Out[2]:

	OBJECTID	StandID	EstablishYear	Age	StandAcres	TractID	StandClass	StandNum	Stan
0	2546	0001-01	2018	2	76.268562	Sarah White	2.0	1	
1	2445	0002-01	2015	5	24.309170	William Wray	2.0	1	
2	2463	0002-03	2020	0	17.694790	William Wray	2.0	3	
3	2584	0003-01	2019	1	68.508957	Zanco	2.0	1	
4	2588	0004-01	2008	12	72.351051	Mary Nail	2.0	1	

In [7]:

```python
def schedule_first_thin(row, base_yr):
    '''determine year for first thin, will be used with apply function over series
    base_yr is the current year of the harvest schedule
    '''
    thin1 = 14
    last_op = row['LastOperation']
    age = row['Age']

    if last_op == 0:
        if age >= 14:
            'no prior thin, scheudule 1st thin now'
            return int(base_yr)
        else:
            'ingrowth scenario'
            return int(base_yr + (thin1 - age))
    else:
        'stand already first thinned'
        return int(0)


def schedule_second_thin(row, base_yr):
    '''determine year for second thin, will be used with apply functio
n over series
    base_yr is the current year of the harvest schedule
    '''
    thin2 = 23
    last_op = row['LastOperation']
    last_thin = row['YrOfLastThin']
    age = row['Age']

    if last_op == 0 and age >=23:
        'first thin only case'
        return int(0)
    elif last_op == 1 and age >= 23:
        'okay to 2nd thin now'
        return int(base_yr)
    elif last_op > 1 and age >= 30:
        'at or beyond rotation age, dont 2nd thin'
        return int(0)
    elif last_op == 2:
        'already 2nd thin, bypass'
        return int(0)
    else:
        if thin2 - age <= 5:
            'late first thin scenario, +5 pushes thin out to between 5
and 9 years'
            return int(base_yr + (thin2 - age) + 5)
        'base scenario applies, thin at age 23'
        return int(base_yr + (thin2 - age))


def schedule_third_thin(row, base_yr):
    '''determine year for third thin, will be used with apply function
over series
    base_yr is the current year of the harvest schedule
    '''
    harvest = 35
    last_op = row['LastOperation']
    last_thin = row['YrOfLastThin']
    estab_yr = row['EstablishYear']

    if last_op == 2:
        'stand has a second thin, see how much time until harvest'
        if harvest + estab_yr - last_thin >= 13:
            return int(last_thin + 7)
    return 0


def schedule_final_harvest(row, base_yr):
    '''determine year for final harvest, will be used with apply funct
ion over series
    base_yr is the current year of the harvest schedule
    '''
    harvest = 35
    age = row['Age']

    if age >= 35:
        'the previous operation does not matter, stand is at rotation
-- harvest now'
        return int(base_yr)
    else:
        'base scenario applies, harvest at 35'
        return int(base_yr + (harvest - age))


sched_base_yr = 2020
raw['1stThin'] = raw.apply(schedule_first_thin, args=(sched_base_yr,),
axis=1)
raw['2ndThin'] = raw.apply(schedule_second_thin, args=(sched_base_yr,)
, axis=1)
raw['3rdThin'] = raw.apply(schedule_third_thin, args=(sched_base_yr,),
axis=1)
raw['Harvest'] = raw.apply(schedule_final_harvest, args=(sched_base_yr
,), axis=1)
```

In [8]:

```
# export dataframe to excel
raw.to_excel('HGT_Thin_Schedule_2.xlsx')
```

In [ ]: