

# Mihin praedorin kannattaa sijoittaa? Stokastinen mallinnus Jaconian liiketoiminnasta

TUOMO SIPOLA

*tuomo.sipola@iki.fi*

**ABSTRACT:** Tämä tutkimus keskittyy praedorien varallisuuden arvon kehittymisen mallintamiseen. Malli perustuu 10000 praedorin varallisuuden kehittymisen stokastiseen simulointiin sääntötaulukon pohjalta. Laskenta-ajoista saadut kuvaajat havainnollistavat sijoitustoiminnan onnistumista. Lopuksi annetaan suosituksia optimaalisten sijoitusten tekemiseksi.

Published: 2017-07-27 Version: 2017-07-28.51

PRAEDORIT OVAT SEIKKAILIJOITA, jotka keräävät aarteita niin Jaconian sivistyneessä maailmassa kuin vaarallisessa Borvariassa (Vuorela ja Hiltunen 2005). Ellei huvittelua ja hyvään elämään sijoittamista lasketa mukaan, varallisuutta kartuttaneella praedorilla, on yleisesti ottaen kolme mahdollisuutta sen säilömiseen (Vuorela ja Hiltunen 2017, s. 88–91):

1. Omaisuuden pitäminen mukana
2. Pankkitalletus
3. Sijoittaminen liiketoimintaan

## OMAIUUUDEN PITÄMINEN MUKANA

Praedorilla on mahdollisuus sijoittaa mukana kulkeviin kalliisiin esineisiin, vaatteisiin ja koruihin. Myös maaomaisuus voi tulla kyseeseen. Tähän toimintamalliin kuitenkin liittyy riskejä, sillä mukana kulkeva omaisuus houkuttaa myös varkaita ja muita epämiellyttäviä tuttavuuksia.

## PANKKITALLETUS

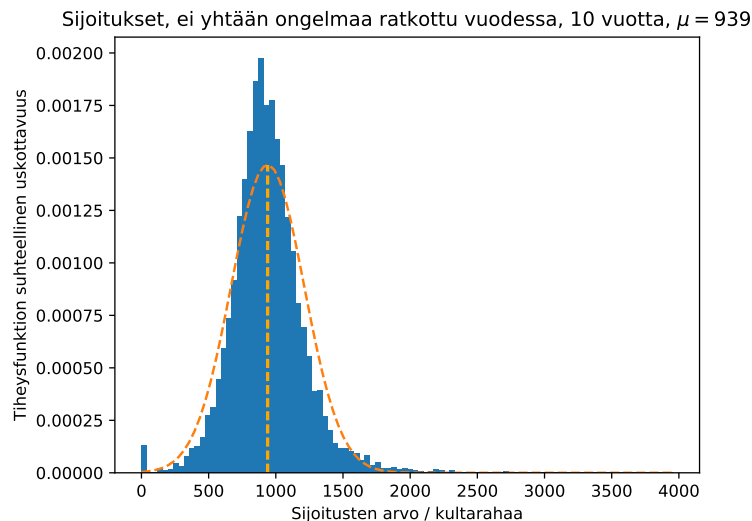
Pankit veloittavat 5% vuotuista talletuspalkkiota (Vuorela ja Hiltunen 2017, s. 91). Laskemalla koron  $p$  vaikutus pääomaan  $c$  ajalta  $t$ ,  $g(c, p, t) = c \left(1 - \frac{p}{100}\right)^t$ , saadaan selville, että  $t = 10$  vuoden kuluttua varallisuuden arvo on enää  $g(1000, 5, 10) = 598$  kultarahaa ja  $t = 50$  vuoden kulutta vain  $g(1000, 5, 50) = 76$  kultarahaa.

## SIJOITTAMINEN LIIKETOIMINTAAN

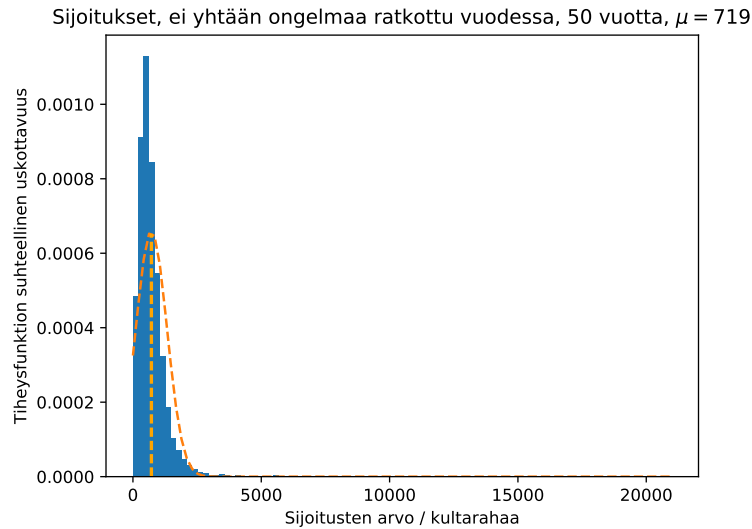
Jacionian taloutta mallinnetaan satunnaistaulukolla. Taulukon tuloksiin voi vaikuttaa myönteisesti, mikäli ratkaisee 1 - 3 liiketoimintaan liittyvää ongelmaa, minkä lisäksi on tärkeää huomioida, että konkurssin uhka on aina läsnä (Vuorela ja Hiltunen 2017, s. 91). Tämä malli voidaan toteuttaa stokastisena tietokoneohjelmalla (Taylor ja Karlin 2014; Nelson 2012), joka mallintaa praedorien sijoitusten arvojen kehittymistä.

Oletetaan, että praedor hankkii alkuhetkellä  $t = 0$  tasajakauman tuottaman määrän  $U(1, 10) = n$  sijoituskohteita. Oletetaan, että hän sijoittaa hallussaan olevista 1000 kultarahasta kuhunkin kohteeseen saman määrän  $1000/n$  kultarahaa. Ratkottavien ongelmien määrä voi olla vuodessa 3 yhdellä omistuksella ja jopa 30, kun omistuksia on 10.

Mallinnetaan  $N = 10000$  praedorin omistuksien arvon kehitys. Oletetaan ensiksi, että praedor ei selvitä yhtään liiketoiminnan ongelmaa. Kuva 1 kuvaa tuloksen  $t = 10$  vuoden päästä ja kuva 2  $t = 50$  vuoden päästä.



Kuva 1: Praedorien sijoitusten arvon kehitys. Mallinnettuja praedoreja  $N = 10000$ ,  $t = 10$ , ei yhtään ongelmaa ratkaistu.



Kuva 2: Praedorien sijoitusten arvon kehitys. Mallinnettuja praedoreja  $N = 10000$ ,  $t = 50$ , ei yhtään ongelmaa ratkaistu.

Oletetaan, että praedor onnistuu selvittämään ongelman 50% tapauksista. Kuva 3 kuvaa tuloksen  $t = 10$  vuoden päästä ja kuva 4  $t = 50$  vuoden päästä.

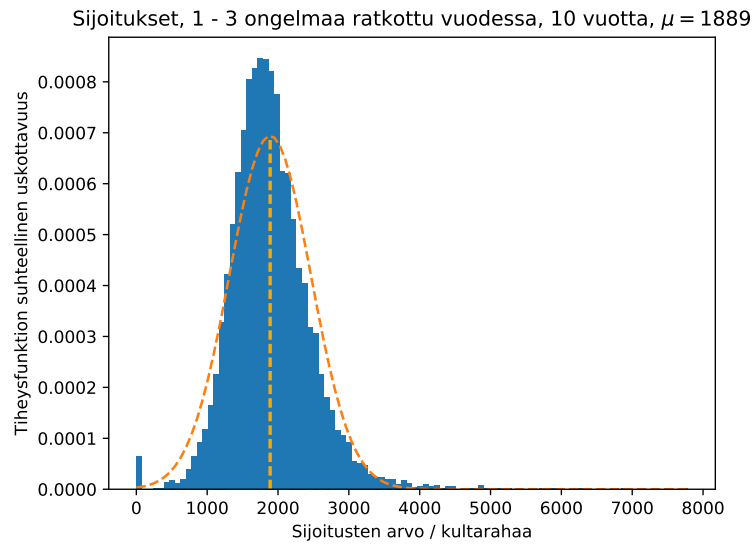
## JOHTOPÄÄTÖKSET

Tulokset voidaan tiivistää näihin ohjeisiin:

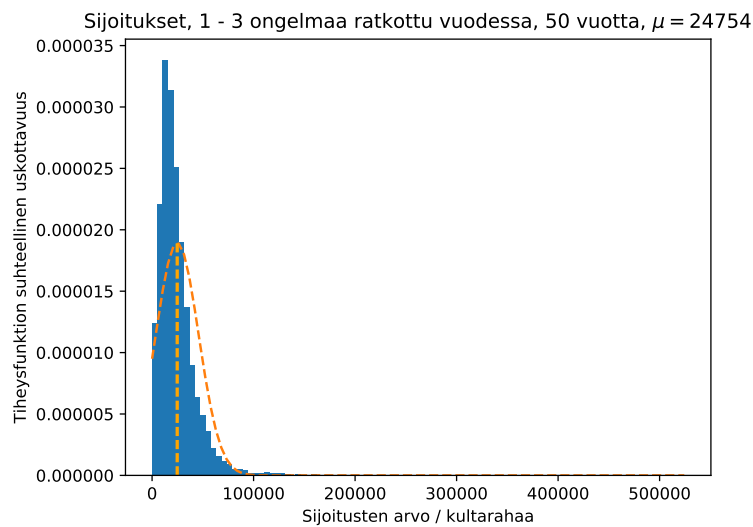
- Omaisuuden kantaminen mukana voi olla hyvä tapa säilöä sitä, kunhan tiedostaa ryöstetyksi tulemisen riskin.
- Pankkiin tallettaminen tuhoaa varallisuutesi arvon.
- Jos sijoitat liiketoimintaan etkä ratkaise yhtään ongelmaa, odotusarvo jää tappion puolelle.
- Jos sijoitat liiketoimintaan, ratkaise kaikki ongelmat tuoton maksimimiseksi.

## LÄHTEET

- Nelson, B.L. (2012). *Stochastic Modeling: Analysis and Simulation*. Dover Books on Mathematics. Dover Publications.
- Taylor, H.M. ja S. Karlin (2014). *An Introduction to Stochastic Modeling*. Elsevier Science.
- Vuorela, Ville ja Petri Hiltunen (2017). *Praedor: Kirottu kirja*. Burger Games.
- Vuorela, Ville ja Petri Hiltunen (2005). *Praedor: Roolipeli Jaconian seikkailijoista*. versio 1.1. Burger Games.



Kuva 3: Praedorien sijoitusten arvon kehitys. Mallinnettuja praedoreja  $N = 10000$ ,  $t = 10$ , 1 - 3 ongelmaa ratkaistu.



Kuva 4: Praedorien sijoitusten arvon kehitys. Mallinnettuja praedoreja  $N = 10000$ ,  $t = 50$ , 1 - 3 ongelmaa ratkaistu.

## LIITE A: LÄHDEKOODI

```
#!/bin/env python
# -*- coding: utf-8 -*-

import random
import numpy
import matplotlib.mlab as mlab
import matplotlib.pyplot as plt

# Which 3N roll results gives which value change in asset.
value_change_chart = {
    0: 1.50,
    1: 1.50,
    2: 1.50,
    3: 1.50,
    4: 1.30,
    5: 1.25,
    6: 1.20,
    7: 1.15,
    8: 1.10,
    9: 1.05,
    10: 1.00,
    11: 1.00,
    12: 0.95,
    13: 0.90,
    14: 0.85,
    15: 0.80,
    16: 0.70,
    17: 0.50,
    18: 0.00,
    19: 0.00,
    20: 0.00,
    21: 0.00
}

def roll_dice(n, k):
    """Roll n k-sided dice and return the sum of them."""
    result = 0
    for i in range(n):
        result += random.randint(1, k)
    return result
```

```

def fail_roll(k):
    """Rolls 3N6 with modifier k but 18 should always return 18
    without the modifier"""
    r = roll_dice(3, 6)
    if r == 18:
        return 18 # Bankrupt if 3N6 gives 3*6 = 18
    return r - k

def update_assets(assets, problems):
    if problems == 'min': # no problems solved
        k = 0
    elif problems == 'max': # all problems solved
        k = 3
    elif problems == 'med': # 50% chance of solving the problems
        k = roll_dice(3, 2) - 3
    assets[:] = [a * value_change_chart[fail_roll(k)] for a in assets]

def create_assets(n_min, n_max, value):
    """Give n_min to n_max assets, value divided equally among them."""
    n = random.randint(n_min, n_max)
    return [value / n] * n

def asset_development(t_max, problems):
    """Let the assets gain or lose value until year t_max is reached."""
    value = 1000
    n_min = 1
    n_max = 10
    assets = create_assets(1, 10, value)

    for t in range(t_max):
        update_assets(assets, problems)

    return sum(assets)

def model(n_praedors, t, problems):
    """Model the asset portfolios of n_praedors during t years.
    Problems can be 'min', 'max' or 'med', being no solved problems,
    3 solved problems every time, or 1 - 3 problems solved.
    Save a figure about it."""

```

```

hist = []
for i in range(n_praedors):
    hist.append(asset_development(t, problems))

# Data statistics.
mu = numpy.mean(hist)
sigma = numpy.std(hist)

# Add histogram.
num_bins = 100
fig, ax = plt.subplots()
n, bins, patches = ax.hist(hist, num_bins, normed=1)

# Best fit line.
y = mlab.normpdf(bins, mu, sigma)
ax.plot(bins, y, '--')

# Mean line.
plt.plot([mu, mu], [0, max(y)],
         color='orange', linestyle='--', linewidth=2)

# Titles.
ax.set_xlabel(u'Sijoitusten arvo / kultarahaa')
ax.set_ylabel(u'Tiheysfunktion suhteellinen uskottavuus')
prob_text = {'min': u'ei yhtään', 'max': u'3', 'med': u'1 - 3'}
ax.set_title(
    u'Sijoitukset, ' + prob_text[
        problems] + ' ongelmaa ratkottu vuodessa, ' + str(
            t) + ' vuotta,  $\mu =$ ' + '{:4.0f}'.format(
                mu) + '$')

# Save figure.
plt.savefig(
    'praedor-' + str(
        n_praedors) + '-' + str(
            t) + '-' + str(
                problems) + '.pdf',
        dpi=None,
        facecolor='w',
        edgecolor='w',
        orientation='portrait',
        papertype=None,
        format='pdf',
        transparent=False,

```

```
        bbox_inches='tight',
        pad_inches=0.2,
        frameon=None)

# Show figure.
plt.show()

if __name__ == "__main__":
    """Run these simulations with these parameters."""
    model(10000, 10, 'min')
    model(10000, 10, 'max')
    model(10000, 10, 'med')
    model(10000, 50, 'min')
    model(10000, 50, 'max')
    model(10000, 50, 'med')
```