
sphinx-example Documentation

Erik Kjellgren

May 21, 2022

GENERAL KODNING

1	Development Pipe-Line	1
1.1	Coding-Style	1
1.2	Pre-Commit	1
1.3	Continuous Integration and Code Coverage	1
1.4	Documentation	1
2	depotmodel.py	3
3	formler.py	5
4	kurtage.py	7
5	laanmodel.py	9
6	skat.py	11
7	valuta.py	13
8	vaerdipapirer.py	15
9	webscrape.py	17
	Python Module Index	19
	Index	21

DEVELOPMENT PIPE-LINE

This section is a brief description of the development pipe-line of this project.

1.1 Coding-Style

- Test-Driven-Development
- Type-hinting is required
- Documentation is required

1.2 Pre-Commit

- Black
- flake8
- mypy
- isort

1.3 Continuous Integration and Code Coverage

- pytest
- travis
- codecov

1.4 Documentation

- Compiled with Sphinx
- Hosted on Read-the-Docs

The documentation uses the [sphinx rtd theme](#).

DEPOTMODEL.PY

FORMLER.PY

`dkfinance_modeller.utility.formler.CAGR(start_kapital: float, slut_kapital: float, antal_år: float) → float`

Beregn annualiseret flerårig vækstrate, “Compound annual growth rate” (CAGR) på engelsk.

$$CAGR = \left(\frac{k_{\text{slut}}}{k_{\text{start}}} \right)^{\frac{1}{n}} - 1$$

k_{slut} slut kapital.

k_{start} start kapital.

n antal år.

$CAGR$ CARG.

Fra, https://en.wikipedia.org/wiki/Compound_annual_growth_rate, 21-11-2020

Parameters

- **start_kapital** – start kapital.
- **slut_kapital** – slut kapital.
- **antal_år** – tid i antal år, behøver ikke at være et heltal.

Returns CAGR

`dkfinance_modeller.utility.formler.afbetalling(klån: float, r: float, n: int) → float`

Beregn størrelse af afbetalings størrelse for at afbetale et lån over n gange.

$$k_{\text{afbetaling}} = k_{\text{ln}} \cdot \frac{r}{1 - (1 + r)^{-n}}$$

r rente på lånet.

n antal gange der skal afbetales.

k_{ln} start kapital der skal afbetales.

$k_{\text{afbetaling}}$ månedlig ydelse.

Parameters

- **klån** – Kapital der skal tilbage betales.
- **r** – renten på lånet.
- **n** – antal afbetalinger.

Returns Ydelse for at afbetale lån over n gange.

`dkfinance_modeller.utility.formler.opsparing(kstart: float, kmåned: float, r: float, n: int) → float`

Opsparings formel.

$$k_n = k_{\text{start}} (1 + r)^n + k_{\text{mned}} \left(\left(\frac{1 - (1 + r)^n}{-r} \right) - 1 + (1 + r)^n \right)$$

r rente.

n antal gange renten bliver beregnet.

k_n slut kapital.

k_{start} start kapital.

k_{mned} månedlig ydelse.

Parameters

- **kstart** – start kapital.
- **kmåned** – månedlig indskud.
- **r** – månedlig rente.
- **n** – antal måneder.

Returns Slut kapital af opsparingen.

`dkfinance_modeller.utility.formler.årlig_til_n_rente(r: float, n: int) → float`

Omregn en årlig rente til en rente over n gange.

$$r_n = (1 + r)^{\frac{1}{n}} - 1$$

r årlig rente.

n antal gange renten bliver beregnet per år.

r_n rente for n periode.

Parameters

- **r** – årlig rente.
- **n** – antal gange renten beregnes.

Returns Rente for n periode.

KURTAGE.PY

```
dkfinance_modeller.aktieskat.kurtage.lunar_kurtage(dkk: float, kurs: float, kurtageprocent: float,
                                                    minimums_kurtage: float, valutakurtage: float)
                                                    → float
```

Kurtage for Lunar invest.

Kurtagen er givet ved,

$$kurtage = m + H(k - 50000)kp + kv$$

k er investeringskapital. m er minimumskurtage. n p er kurtageprocent. n v er valutakurtageprocent. n $H(x)$ er Heaviside step funktionen.

```
dkfinance_modeller.aktieskat.kurtage.lunar_kurtage_bygger(valuta: str = 'DKK') → Callable[[float,
                                                                                          float], float]
```

Bygger af Lunar invest kurtage funktion.

<https://static-assets.prod.lunarway.com/da/docs/prisliste-privat/>, 21-10-2020

Parameters **valuta** – valuta værdipapir handles i.

Returns Kurtagefunktion.

```
dkfinance_modeller.aktieskat.kurtage.nordnet_kurtage_bygger(valuta: str = 'DKK', valutakurs: float
                                                            = 1.0, valutakonto: bool = False) →
                                                            Callable[[float, float], float]
```

Bygger af Nordnet kurtage funktion.

<https://www.nordnet.dk/dk/kundeservice/prisliste/priser-aktiedepot>, 21-10-2020

Medregner ikke GDR-gebyr.

Parameters

- **valuta** – valuta værdipapir handles i. Euro vil give XETRA børsen.
- **valutakurs** – omregningsfaktor til DKK.
- **valutakonto** – handler foretages i underkonto i given valuta. Valutakurtagen vil ikke være inkluderet i kurtagefunktionen.

Returns Kurtagefunktion.

```
dkfinance_modeller.aktieskat.kurtage.nulskurtage(dkk: float, kurs) → float
```

Ingen kurtage.

Parameters

- **dkk** – kapital der skal betales kurtage af.

- **kurs** – kurs på værdipapir.

Returns kurtage = 0 DKK

`dkfinance_modeller.aktieskat.kurtage.saxo_kurtage_bygger(valuta: str = 'DKK', valutakurs: float = 1.0, underkonto: bool = False) → Callable[[float, float], float]`

Bygger af Saxo kurtage funktion.

<https://www.home.saxo/da-dk/rates-and-conditions/stocks/commissions>, 21-10-2020

Parameters

- **valuta** – valuta værdipapir handles i. Euro vil give XETRA børsen.
- **valutakurs** – omregningsfaktor til DKK.
- **underkonto** – handler foretages i underkonto i given valuta. Valutakurtagen vil ikke være inkluderet i kurtagefunktionen.

Returns Kurtagefunktion.

`dkfinance_modeller.aktieskat.kurtage.saxo_nordnet_kurtage(dkk: float, kurs: float, valutakurs: float, kurtageprocent: float, minimums_kurtage: float, valutakurtage: float, saxo_usd_type: bool) → float`

Kurtage for Nordnet og Saxo bank.

Kurtagen er givet ved,

$$kurtage(k) = \max(m, kp) + kv$$

k er investeringskapital.

m er minimumskurtage.

p er kurtageprocent.

v er valutakurtageprocent.

Med mindre det er Amerikanske aktier til under 10 USD ved Saxo, så er kurtagen givet ved,

$$kurtage(k) = \max(m, 0.02N) + kv$$

N er antal købte værdipapirer, faktoren foran er i USD.

Parameters

- **dkk** – kapital der skal betales kurtage af.
- **kurs** – kurs på værdipapir i DKK.
- **kurtageprocent** – kurtageprocent.
- **minimums_kurtage** – minimums kurtage givet i værdipapirets valuta.
- **valutakurtage** – valutakurtage.
- **saxo_usd_type** – Aktiver speciel kurtage for USD værdipapirer hos Saxo bank, hvis kurs under 10 USD.

Returns kurtage

LAANMODEL.PY

```
class dkfinance_modeller.laan.laanmodel.SUlån(uddannelse_måneder: int, afdragsfrie_månder: int,  
                                             rente: float)
```

Klasse for SU-lån.

propager_måned() → Generator[Tuple[float, float], None, None]

Propagere SU lån måned for måned.

Positivt “afdrag” er lånte penge udbetalt.

Returns afdrag og fradrag.

SKAT.PY

```
class dkfinance_modeller.aktieskat.skat.Skat(beskatningstype: str)
```

Skat.

Variables

- **progressionsgrænse** (*float*) – Progressionsgrænse.
- **skatteprocenter** (*List[float]*) – Skatteprocenter.
- **skattefunktion** (*Callable[[float], float]*) – Skattefunktion.

```
__init__(beskatningstype: str) → None
```

Setup skat.

Parameters **beskatningstype** – Hvilken beskatning der skal bruges, ['aktie', 'ask', 'pension', 'nul'].

```
beregn_skat(dkk: float) → float
```

Beregn skat.

Parameters **dkk** – Kapital til beskatning.

Returns Skat i DKK.

VALUTA.PY

`dkfinance_modeller.aktieskat.valuta.nordnet_valutakonto_kurtage(dkk: float) → float`

Valutakurtage for Nordnet valutakonto.

<https://www.nordnet.dk/faq/2334-hvad-koster-veksling-hos-nordnet-vs-min-bank>, 31-10-2020

Parameters **dkk** – kapital der skal betales valutakurtage af.

Returns kurtage

`dkfinance_modeller.aktieskat.valuta.nulvalutakurtage(dkk: float) → float`

Ingen kurtage.

Parameters **dkk** – kapital der skal betales valutakurtage af.

Returns kurtage = 0 DKK

`dkfinance_modeller.aktieskat.valuta.saxo_underkonto_kurtage(dkk: float) → float`

Valutakurtage for Saxo underkonto.

<https://www.home.saxo/da-dk/rates-and-conditions/commissions-charges-and-margin-schedule>, 31-10-2020.

Parameters **dkk** – kapital der skal betales valutakurtage af.

Returns kurtage

VAERDIPAPIRER.PY

class dkfinance_modeller.aktieskat.vaerdipapirer.ETF(*kurs: float, åop: float, beskatningstype: str*)

ETF og investeringsforening.

property antal_værdipapirer: float

Getter for antal_værdipapirer.

Returns Antal værdipapirer

lagerrealisering(*ændre_kurs: bool = True*) → float

Beregn overskud via lagerbeskatning.

Parameters *ændre_kurs* – Sætter beskattet_kurs til kurs. False hvis skattepligtig beholdning skal have, uden at skatten antages betalt.

Returns Kapital der skal beskattes.

modregn_åop() → None

Trækker ÅOP fra kursen, for en måned.

$$a_n|_{n=12} = 1 - \left(\sqrt[n]{1-a}\right)|_{n=12}$$

n antal gange ÅOP betales over per år.

a_n er ÅOP splittet op i n dele.

a er ÅOP.

opdater_kurs(*kursændring: float*) → None

Opdaterer kursen.

Parameters *kursændring* – Ændring af kurs.

tilføj_enheder(*antal: int*) → None

Tilføj antal enheder af ETFen.

ETF'er bliver beskattet via. gennemsnitsmetoden. Den nye gennemsnitskurs er derfor:

$$k_{\text{avg,new}} = \frac{n \cdot k + n_{\text{old}} \cdot k_{\text{avg,old}}}{n + n_{\text{old}}}$$

k kurs.

$k_{\text{avg,new}}$ nye gennemsnitskurs.

$k_{\text{avg,old}}$ gammel gennemsnitskurs.

n antal værdipapirer der bliver tilføjet.

n_{old} antal værdipapirer allerede i beholdningen.

Parameters **antal** – antal enheder at tilføje.

total_værdi() → float

Få total værdi af beholdning.

Returns Total værdi af beholdning.

WEBSRAPE.PY

`dkfinance_modeller.utility.webscrape.få_etf_info`(*ISINer: List[str], threads: int*) → List[Dict[str, object]]

Få information omkring givne ETFer fra justetf.com/en/.

Parameters **ISIN** – Liste af ISINer for ETFer.

Returns En liste af dicts med fundne information om ETFen.

`dkfinance_modeller.utility.webscrape.justetf_info`(*ISIN: str*) → Dict[str, object]

Få information omkring given ETF fra justetf.com/en/.

Parameters **ISIN** – ISIN for ETFen.

Returns En dict med fundne information om ETFen.

PYTHON MODULE INDEX

d

`dkfinance_modeller.aktieskat.kurtage`, 7
`dkfinance_modeller.aktieskat.skat`, 11
`dkfinance_modeller.aktieskat.vaerdipapirer`,
15
`dkfinance_modeller.aktieskat.valuta`, 13
`dkfinance_modeller.laan.laanmodel`, 9
`dkfinance_modeller.utility.formler`, 5
`dkfinance_modeller.utility.webscrape`, 17

Symbols

`årlig_til_n_rente()` (in module `dkfinance_modeller.utility.formler`), 6
`__init__()` (`dkfinance_modeller.aktieskat.skat.Skat` method), 11

A

`afbetalling()` (in module `dkfinance_modeller.utility.formler`), 5
`antal_værdipapirer` (`dkfinance_modeller.aktieskat.vaerdipapirer.ETF` property), 15

B

`beregn_skat()` (`dkfinance_modeller.aktieskat.skat.Skat` method), 11

C

`CAGR()` (in module `dkfinance_modeller.utility.formler`), 5

D

`dkfinance_modeller.aktieskat.kurtage` module, 7
`dkfinance_modeller.aktieskat.skat` module, 11
`dkfinance_modeller.aktieskat.vaerdipapirer` module, 15
`dkfinance_modeller.aktieskat.valuta` module, 13
`dkfinance_modeller.laan.laanmodel` module, 9
`dkfinance_modeller.utility.formler` module, 5
`dkfinance_modeller.utility.webscrape` module, 17

E

`ETF` (class in `dkfinance_modeller.aktieskat.vaerdipapirer`), 15

F

`få_etf_info()` (in module `dkfinance_modeller.utility.webscrape`), 17

J

`justetf_info()` (in module `dkfinance_modeller.utility.webscrape`), 17

L

`lagerrealisering()` (`dkfinance_modeller.aktieskat.vaerdipapirer.ETF` method), 15
`lunar_kurtage()` (in module `dkfinance_modeller.aktieskat.kurtage`), 7
`lunar_kurtage_bygger()` (in module `dkfinance_modeller.aktieskat.kurtage`), 7

M

`modregn_åop()` (`dkfinance_modeller.aktieskat.vaerdipapirer.ETF` method), 15
module
`dkfinance_modeller.aktieskat.kurtage`, 7
`dkfinance_modeller.aktieskat.skat`, 11
`dkfinance_modeller.aktieskat.vaerdipapirer`, 15
`dkfinance_modeller.aktieskat.valuta`, 13
`dkfinance_modeller.laan.laanmodel`, 9
`dkfinance_modeller.utility.formler`, 5
`dkfinance_modeller.utility.webscrape`, 17

N

`nordnet_kurtage_bygger()` (in module `dkfinance_modeller.aktieskat.kurtage`), 7
`nordnet_valutakonto_kurtage()` (in module `dkfinance_modeller.aktieskat.valuta`), 13
`nul_kurtage()` (in module `dkfinance_modeller.aktieskat.kurtage`), 7
`nulvalutakurtage()` (in module `dkfinance_modeller.aktieskat.valuta`), 13

O

`opdater_kurs()` (*dkfinance_modeller.aktieskat.vaerdipapirer.ETF*
method), 15

`opsparing()` (*in module dkfinance_modeller.utility.formler*), 5

P

`propager_måned()` (*dkfinance_modeller.laan.laanmodel.SULån*
method), 9

S

`saxo_kurtage_bygger()` (*in module dkfinance_modeller.aktieskat.kurtage*), 8

`saxo_nordnet_kurtage()` (*in module dkfinance_modeller.aktieskat.kurtage*), 8

`saxo_underkonto_kurtage()` (*in module dkfinance_modeller.aktieskat.valuta*), 13

`Skat` (*class in dkfinance_modeller.aktieskat.skat*), 11

`SULån` (*class in dkfinance_modeller.laan.laanmodel*), 9

T

`tilføj_enheder()` (*dkfinance_modeller.aktieskat.vaerdipapirer.ETF*
method), 15

`total_værdi()` (*dkfinance_modeller.aktieskat.vaerdipapirer.ETF*
method), 16