Barn Mass Balance

Table of contents

WHYSOR

Everlasting IT

```python
import os, json
```

Achtergrond

In Nederland is men momenteel vooral geïnteresseerd in de emissie uit stallen met dieren. We kunnen onderscheid maken tussen 2 type stallen: de natuurlijk geventileerde stallen (open stallen) en de mechanisch geventileerde stallen (gesloten stallen). Over het algemeen zitten melkkoeien (en -geiten) in open stallen en zitten de intensievere dieren (pluimvee, varkens, kalveren) in gesloten stallen.

Emissieberekening

Emissie wordt berekend door de gemeten concentratie van een bepaalde stof te vermenigvuldigen met het ventilatiedebiet. De eenheid van emissie wordt uitgedrukt in massa/tijdseenheid" (zoals kg/uur).

Concentratiemeting

De concentratie van de stof wordt gemeten met sensoren. Deze worden op bepaalde plekken bij de luchtinlaat en luchtuitlaat van de stal gehangen. Er is een consensus dat de gemeten concentraties dan betrouwbaar en representatief zijn.

Ventilatiedebiet

Mechanisch geventileerde stallen

In mechanisch geventileerde stallen wordt het ventilatiedebiet bepaald met behulp van meetwaaiers. Deze meetwaaiers worden op ventilatoren geplaatst en kunnen meten hoeveel lucht er per tijdseenheid door de ventilator wordt geblazen. Omdat mechanisch geventileerde stallen maar één (of enkele) uitstroomopeningen hebben, kan op deze manier worden bepaald wat het totale ventilatiedebiet van een stal is. Er is een consensus dat het bepalen van ventilatiedebiet met behulp van meetwaaiers betrouwbaar en representatief is.

Natuurlijk geventileerde stallen

Natuurlijk geventileerde stallen zijn voornamelijk open, waardoor het niet mogelijk is om met meetwaaiers te bepalen wat het debiet van de stallen is. Daarom wordt bij deze stallen gebruik gemaakt van de CO_2 massabalans. Dit is een theoretische benadering gebaseerd op verschillende parameters die door de tijd heen kunnen veranderen.

Emissieberekening met CO_2 massabalans

De ratiomethode is een veelgebruikte methode om de emissie van ammoniak (NH_3) in stallen te schatten. Deze methode maakt gebruik van de concentratie van CO_2 als tracergas, omdat CO_2 een relatief constante productie heeft in de stal en goed te berekenen en meten is. De basis van de ratiomethode is het idee dat de verhouding tussen de concentraties van CO_2 binnen en buiten de stal een indicatie geeft van het ventilatiedebiet en daarmee de totale emissie van NH_3.

Een randvoorwaarde van de ratiomethode is dat de concentraties van NH_3 en het tracergas - in dit geval CO_2 - op dezelfde meetpunten en met dezelfde meetfrequentie gemeten moeten worden. Om een goede schatting van de emissie te verkrijgen is het van belang dat de concentratieratio's per meetpunt worden geschat en daarna een gemiddeld van deze waarden wordt genomen, in plaats dan eerst een gemiddelde concentratie van al die punten te bepalen en daarna de ratiomethode te gebruiken.

Support calculations

Gas density

The density of a gas can be calculated using the ideal gas law:

$$\rho = \frac{Mp}{R \cdot T}$$

where: * M is the molar mass of the gas in kg/mol

- p is the pressure in Pascals
- T is the temperature in Kelvin
- R is the universal gas constant

---

source

gas_density

```
gas_density (P:float, T:float, ppm:float, molweight:float)
```

Calculates mass density in grams per cubic metre P : pressure in Pa T : temperature in degrees Kelvin ppm : measured parts per million molweight: molecular weight in grams per mole

	Type	Details
P	float	pressure in Pascal
T	float	temperature in Kelvin
ppm	float	measured parts per million
molweight	float	molecular weight in grams per mole

Gasconcentraties in de lucht

Concentraties van chemicaliën in de lucht worden meestal gemeten als de massa van chemicaliën (milligram, microgram, nanogram of picogram) per volume lucht (kubieke meter of kubieke voet). Concentraties kunnen ook worden uitgedrukt als delen per miljoen (ppm) of delen per miljard (ppb) door gebruik te maken van een conversiefactor. Deze conversiefactor is gebaseerd op het moleculair gewicht van de chemische stof en is voor elke chemische stof verschillend. Typisch worden conversies voor chemicaliën in de lucht gemaakt met een veronderstelling van een druk van 1 atmosfeer en een temperatuur van 25 graden Celsius. Voor deze omstandigheden is de vergelijking om te converteren van concentratie in delen per miljoen naar concentratie in milligram per kubieke meter (mg/m3) als volgt:

$$\text{Concentratie (mg/m3)} = 0.0409 \times \text{concentratie (ppm)} \times \text{moleculair gewicht}$$

Concentrations of chemicals

Concentrations of chemicals in the air are usually measured as the mass of chemicals (milligrams, micrograms, nanograms or picograms) per volume of air (cubic meters or cubic feet). Concentrations can also be expressed as parts per million (ppm) or parts per billion (ppb) by using a conversion factor. This conversion factor is based on the molecular weight of the chemical and it is different for every chemical. The temperature of the atmosphere also has an influence on the calculation.

Typically, conversions for chemicals in air are made as suming a pressure of 1 atmosphere and a temperature of 25 degrees Celsius. For these conditions, the equation to convert from concentration in parts per million to con centration in milligrams per cubic meter (mg/m3) is as follows:

$$\text{Concentration (mg/m3)} = 0.0409 \times \text{concentration (ppm)} \times \text{molecular weight}$$

Functie implementatie

---

source

gas_density_from_sensor_measurment

```
gas_density_from_sensor_measurment (ppm:float, molweight:float)
```

Calculates mass density in milligrams per cubic metre

	Type	Details
ppm	float	measured parts per million
molweight	float	molecular weight in grams per mole

CO_2 productie

De CO_2 productie in een stal (in m^3 / uur) kan worden berekend met behulp van de volgende formules voor melkvee en pinken

Melkvee

$$PCO_2 = 0.2\frac{5.6m^{0.75} + 22Y_1 + 1.6 \times 10^{-5}p^3}{1000}$$

Where:

- m is the live weight in kg
- Y_1 is the daily milk production in kg per dier per dag
- p number of dracht dagen

Functie implementatie

---

source

PCO2_melkvee

```
PCO2_melkvee (aantal, melkproductie, drachtdagen, gewicht)
```

CO2 productie van melkvee per dier per dag gewicht: (gemiddelde) gewicht van de dieren melkproductie: melkproductie in kg per dier per dag drachtdagen: gemiddelde drachttijd (in dagen) De defaults zijn voor droogstaande koeien

	Details
aantal	number of animals
melkproductie	milk production in kg per animal per day
drachtdagen	days carrying (average)
gewicht	average weight of the animals in kg

```
test_args_melkvee = dict(
    aantal=130,
    melkproductie=28,
    drachtdagen=160,
    gewicht=650
)

PCO2_melkvee(**test_args_melkvee)
```

```
np.float64(36.46325050213528)
```

```
test_args_droog = dict(
    aantal=6,
    melkproductie=0,
    drachtdagen=220,
    gewicht=650
)

PCO2_melkvee(**test_args_droog)
```

```
np.float64(1.0695176539447053)
```

Pinken

$$PCO_2 = 0.2\frac{7.64m^{0.69} + Y_2\left(\frac{23}{M} - 1\right)\left(\frac{57.27+0.302m}{1-0.171Y_2}\right) + 1.6\times 10^{-5}p^3}{1000}$$

Where:

- m is the live weight in kg
- M is the energy content of their food in MJ per kg
- Y_2 is the daily weight gain in kg per dier per dag
- p number of dracht dagen

Functie implementatie

source

PCO2_pinken

```
PCO2_pinken (aantal, energievoeding, drachtdagen, gewicht,
             gewichtstoename)
```

CO2 productie van pinken

	Details
aantal	number of animals
energievoeding	energy feed
drachtdagen	days carrying (average)
gewicht	average weight of the animals in kg
gewichtstoename	average weight gain of the animals in kg per day

```
test_args_pinken = dict(
    aantal=0,
    energievoeding=10,
    drachtdagen=140,
    gewicht=400,
    gewichtstoename=0.6
)
PCO2_pinken(**test_args_pinken)
```

```
np.float64(0.0)
```

```
test_args_pinken_niet_drachtig = dict(
    aantal=0,
```

```
    energievoeding=10,
    drachtdagen=0,
    gewicht=250,
    gewichtstoename=0.6
)
PCO2_pinken(**test_args_pinken_niet_drachtig)
```

```
np.float64(0.0)
```

Temperatuur correctie

Temperatuur heeft invloed op spijsveetering en gedrag en daarmee op de CO_2 productie, correctie kan worden toegepast met de volgende formule:

$$PCO_2(T) = PCO_2 \times (1000 + 4 \times (20 - T_{stal}))/1000$$

---

source

PCO2_temperatuurcorrectie

```
PCO2_temperatuurcorrectie (pco2, temperatuur)
```

Bereken temperatuur correctie voor de CO2 productie

	Details
pco2	calculated CO2 production in cubic meters per hour
temperatuur	temperature in the barn in degrees Celsius

---

source

calculate_temperatuur_correctie

```
calculate_temperatuur_correctie (temperatuur)
```

Calculate temperature correction factor for CO2 production

PcO_2 functie categorie mapping

```
dict(inspect.signature(PCO2_melkvee).parameters)
```

```
{'aantal': <Parameter "aantal">,
 'melkproductie': <Parameter "melkproductie">,
 'drachtdagen': <Parameter "drachtdagen">,
 'gewicht': <Parameter "gewicht">}
```

```
pco2_category_functions_parameters
```

```
{'melkvee': {'aantal': <Parameter "aantal">,
  'melkproductie': <Parameter "melkproductie">,
  'drachtdagen': <Parameter "drachtdagen">,
  'gewicht': <Parameter "gewicht">},
 'droogstaande koeien': {'aantal': <Parameter "aantal">,
  'melkproductie': <Parameter "melkproductie">,
  'drachtdagen': <Parameter "drachtdagen">,
  'gewicht': <Parameter "gewicht">},
 'drachtig jongvee': {'aantal': <Parameter "aantal">,
  'energievoeding': <Parameter "energievoeding">,
  'drachtdagen': <Parameter "drachtdagen">,
  'gewicht': <Parameter "gewicht">,
  'gewichtstoename': <Parameter "gewichtstoename">},
 'niet drachtig jongvee': {'aantal': <Parameter "aantal">,
  'energievoeding': <Parameter "energievoeding">,
  'drachtdagen': <Parameter "drachtdagen">,
  'gewicht': <Parameter "gewicht">,
  'gewichtstoename': <Parameter "gewichtstoename">}}
```

PCO_2 Parameters

Voor het CO2-productiemodel zijn een aantal productiegegevens nodig. Melkproductie en –samenstelling worden altijd gemeten en gerapporteerd. De andere benodigde parameters (diergewicht, dagen in dracht, en voor jongvee de energiewaarde van het voer en gewichtstoename), worden bij voorkeur op basis van metingen op de bedrijfslocaties vastgesteld. Wanneer deze niet beschikbaar zijn dienen de volgende standaardwaarden voor te worden gebruikt.

```
pd.DataFrame(_default_parameters).set_index('categorie')
```

	gewicht	drachtdagen	melkproductie	energievoeding	gewichtstoename
categorie					
melkvee	650	160	NaN	NaN	NaN
droogstaande koeien	650	220	0.0	NaN	NaN
drachtig jongvee	400	140	NaN	10.0	0.6
niet drachtig jongvee	250	0	NaN	10.0	0.6

```
print(json.dumps(default_pco2_parameters, indent=4))
```

```
{
    "melkvee": {
        "gewicht": 650,
        "drachtdagen": 160
    },
    "droogstaande koeien": {
        "gewicht": 650,
        "drachtdagen": 220,
        "melkproductie": 0
    },
    "drachtig jongvee": {
        "gewicht": 400,
        "drachtdagen": 140,
        "energievoeding": 10.0,
        "gewichtstoename": 0.6
    },
    "niet drachtig jongvee": {
        "gewicht": 250,
        "drachtdagen": 0,
        "energievoeding": 10.0,
        "gewichtstoename": 0.6
    }
}
```

---

source

create_pco2_function_mapping_from_parameters

```
create_pco2_function_mapping_from_parameters (pco2_parameters)
```

Create a mapping of category to PCO2 calculation functions

---

source

PCO2_calculation_from_mapping

```
PCO2_calculation_from_mapping (mapping, category, aantal, **kwargs)
```

Test berekeningen

Emissie berekeningen

Ratiomethode

De ammoniakemissies (E_i; in kg/jaar per dierplaats) worden per meetdag i bepaald op basis van de geschatte CO_2 - productie in de stal (PCO_{2i}; in m^3 CO_2 /uur), en de gemiddelde concentratieratio van CO_2 en NH_3 als CR_i over alle meetpunten m waar CO_2 - en NH_3 concentraties tegelijkertijd in de stal gemeten zijn:

$$E_i = PCO_{2i} \cdot CR_i$$

Voor CR_i

$$CR_i = \frac{1}{m}\sum_m \frac{(NH_3)_{im}^{stal} - \left(\overline{NH_3}\right)_i^{buiten}}{(CO_2)_{im}^{stal} - \left(\overline{CO_2}\right)_i^{buiten}}$$

$$\overline{X_i^{buiten}} = \sum_m X_i^{buiten}$$

Waarin

$\boldsymbol{X_{im}^{stal}}$
: het 24-uurs gemiddelde van de concentratie van stof X in stal op meetdag i en op meetpunt m

$\boldsymbol{X_{im}^{buiten}}$
: het 24-uurs gemiddelde van de concentratie van stof X in de ingaande lucht op meetdag i en op meetpunt m

$\overline{\boldsymbol{X_{i}^{buiten}}}$
: het 24-uurs gemiddelde van de concentratie van stof X in de ingaande lucht op meetdag i over alle meetpunten

$$\overline{(NH_3)_i}^{buiten} = \frac{1}{m}\sum_m (NH_3)_i^{buiten}$$

and

$$\overline{CO_{2i}}^{buiten} = \frac{1}{n}\sum_{k=1}^{n} (CO_2)_{ik}^{buiten}$$

Ratiomethode met twee meetpunten

Wanneer er slechts twee meetpunten zijn, een binnen en een buiten, dan vervalt de berekening van de gemiddelden over meetpunten en kan de emissie worden berekend met de vereenvoudiging van CR_i:

$$CR_i = \frac{(NH_3)_i^{stal} - (NH_3)_i^{buiten}}{(CO_2)_i^{stal} - CO_{2i}^{buiten}}$$

en

$$E_i = PCO_{2i} \cdot CR_i$$

wordt

$$E_i = PCO_{2i} \cdot \frac{(NH_3)_i^{stal} - (NH_3)_i^{buiten}}{(CO_2)_i^{stal} - CO_{2i}^{buiten}}$$

Implementatie ratiomethode

We verwachten dat de gebruiker de volgende data als timeseries dataframe aanlevert:

Kolomnaam	Omschrijving	Eenheid
CO2_stal	CO2 concentratie in de stal in ppm	ppm
CO2_buiten	CO2 concentratie buiten de stal in ppm	ppm
NH3_stal	NH3 concentratie in de stal in ppm	ppm
NH3_buiten	NH3 concentratie buiten de stal in ppm	ppm
temperatuur	Temperatuur in de stal in Celcius	°C

Daarnaast verwachten we dat de gebruiker de volgende gegevens meegeeft

bezetting
: aantal dieren in de stal per categorie als dictionary met categorie als key en aantal als value

parameters
: dictionary met de parameters voor de verschillende categorieën. Missende waardes voor parameters worden aangevult uit de volgende standaard parameters:

```
pd.DataFrame(default_pco2_parameters).transpose()
```

	gewicht	drachtdagen	melkproductie	energievoeding	gewichtstoename
melkvee	650.0	160.0	NaN	NaN	NaN
droogstaande koeien	650.0	220.0	0.0	NaN	NaN
drachtig jongvee	400.0	140.0	NaN	10.0	0.6
niet drachtig jongvee	250.0	0.0	NaN	10.0	0.6

Externe data

Data voor verificatie van de implementatie wordt veelal aangeleverd in excel werkboeken. Deze data kan worden ingelezen en aangepast aan onze behoeften.

VERA data

```python
vera_data_filename = os.path.join(os.getcwd(), '..', 'data', 'massabalans', 'Rekenbestand
emissie VERA.xlsx')
vera_dataframe = pd.read_excel(
    vera_data_filename,
    sheet_name='Emissions (daily means)',
    header=3,
    index_col=7,
    parse_dates=True
).drop([
    ' C1:                  cows >= 70%',
    'C2:                          Occupation rate >= 90%',
    'C3:                                                             milk production > 25',
    ' C1:                  heifers < 30%',
    'C2:                          Occupation rate >= 80%',
    'C3:                                                             milk production >
25.1',
    'C4:                                      urea content in milk > 15',
    'C5:                            dry cows < 25%'], axis=1
).dropna(axis=1, how='all')
```

```python
vera_dataframe.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 60 entries, 2011-04-04 to 2012-02-16
Data columns (total 55 columns):
 #   Column                                                  Non-Null Count  Dtype
---  ------                                                  --------------  -----
 0   Measurement institute                                   60 non-null     object
 1   Animal Category                                         60 non-null     object
 2   Housing system                                          60 non-null     object
 3   Measurement location                                    60 non-null     object
 4   Measurement period                                      60 non-null     int64
 5   Measurement day (in period)                             60 non-null     int64
 6   Day in year                                             60 non-null     int64
 7   Outside temperature [oC]                                24 non-null     float64
 8   Outside RH [%]                                          23 non-null     float64
 9   Inside temperature [oC]                                 59 non-null     float64
 10  Inside RH [%]                                           44 non-null     float64
 11  Animal places                                           60 non-null     int64
 12  Milking cows                                            60 non-null     int64
 13  Dry cows                                                60 non-null     int64
 14  Heifers (pregnant)                                      60 non-null     int64
 15  Heifers (not pregnant)                                  60 non-null     int64
 16  Floor type (0: slatted floor; 1: closed floor)          60 non-null     int64
 17  Walking area per animal (m2)                            60 non-null     float64
 18  Grazing (hours per day)                                 60 non-null     int64
 19  Closed cubicles                                         60 non-null     int64
 20  Milk production [kg/animal/day]                         56 non-null     float64
 21  Milk [% protein]                                        60 non-null     float64
 22  Milk [% fat]                                            60 non-null     float64
 23  Urea content in milk [mg/100g]                          56 non-null     float64
 24  Weight milking cows [kg]                                60 non-null     int64
 25  Weight dry cows [kg]                                    60 non-null     int64
 26  Weight heifers (pregnant) [kg]                          60 non-null     int64
 27  Weight heifers (not pregnant) [kg]                      60 non-null     int64
 28  Days in pregnancy (milking cows)                        60 non-null     int64
 29  Days in pregnancy (dry cows)                            60 non-null     int64
 30  Days in pregnancy (heifers)                             60 non-null     int64
 31  Energy value of feed (heifers; MJ/kg dry matter)        60 non-null     int64
 32  Weight gain heifers [kg/day]                            60 non-null     float64
 33  CO2 inside [ppm]                                        60 non-null     int64
```

```
 34  CO2 outside [ppm]                                     52 non-null     float64
 35  NH3 inside [mg/m3]                                    33 non-null     float64
 36  NH3 outside [mg/m3]                                   60 non-null     int64
 37  Number of animals                                     60 non-null     int64
 38  Dairy cows (milking + dry)                            60 non-null     int64
 39  % closed cubicles                                     60 non-null     float64
 40  Occupation rate (%)                                   60 non-null     float64
 41  Dairy cows (%)                                        60 non-null     float64
 42  Heifers vs. dairy cows (%)                            60 non-null     float64
 43  Dry cows vs. dairy cows (%)                           60 non-null     float64
 44  Heat production milking cows (hpu)                    56 non-null     float64
 45  Heat production dry cows (hpu)                        56 non-null     float64
 46  Heat production heifers (pregnant) (hpu)              60 non-null     float64
 47  Heat production heifers (not pregnant) (hpu)          60 non-null     float64
 48  Total heat production (hpu)                           60 non-null     float64
 49  Total heat production corrected for temperature (hpu) 59 non-null     float64
 50  Ventilation rate [m3/h]                               51 non-null     float64
 51  Ventilation rate [m3/h per animal]                    51 non-null     float64
 52  NH3 Emission [kg/year per animal place]               32 non-null     float64
 53  Summary                                               32 non-null     float64
 54  Summary.1                                             32 non-null     float64
dtypes: float64(28), int64(23), object(4)
memory usage: 26.2+ KB
```

Exctractie van werkbare data uit gegeven werkboeken

Warmte & CO_2 data

```
data = vera_dataframe.copy()
datacolumns = set(data.columns)
```

[source](#)

find_production_column_names

```
find_production_column_names (data:pandas.core.frame.DataFrame)
```

Find the column names for the co2 production columns in the VERA data

```
print(json.dumps(find_production_column_names(vera_dataframe), indent=3))
```


```
{
   "drachtdagen": [
      "Days in pregnancy (heifers)",
      "Days in pregnancy (dry cows)",
      "Days in pregnancy (milking cows)"
   ],
   "energievoeding": [
      "Energy value of feed (heifers; MJ/kg dry matter)"
   ],
   "melkproductie": [
      "Milk production [kg/animal/day]"
   ],
   "gewichtstoename": [
```

```
        "Weight gain heifers [kg/day]"
    ],
    "gewicht": [
        "Weight heifers (pregnant) [kg]",
        "Weight heifers (not pregnant) [kg]",
        "Weight milking cows [kg]",
        "Weight dry cows [kg]"
    ],
    "remaining_columns": [
        "Dry cows vs. dairy cows (%)",
        "Closed cubicles",
        "NH3 outside [mg/m3]",
        "Summary.1",
        "Day in year",
        "% closed cubicles",
        "Housing system",
        "Animal Category",
        "CO2 outside [ppm]",
        "NH3 Emission [kg/year per animal place]",
        "Animal places",
        "NH3 inside [mg/m3]",
        "Ventilation rate [m3/h]",
        "CO2 inside [ppm]",
        "Dairy cows (milking + dry)",
        "Number of animals",
        "Total heat production corrected for temperature (hpu)",
        "Grazing (hours per day)",
        "Heat production heifers (pregnant) (hpu)",
        "Measurement period",
        "Total heat production (hpu)",
        "Measurement institute",
        "Milk [% protein]",
        "Urea content in milk [mg/100g]",
        "Heifers (not pregnant)",
        "Measurement day (in period)",
        "Occupation rate (%)",
        "Heifers vs. dairy cows (%)",
        "Heat production milking cows (hpu)",
        "Walking area per animal (m2)",
        "Dairy cows (%)",
        "Measurement location",
        "Ventilation rate [m3/h per animal]",
        "Floor type (0: slatted floor; 1: closed floor)",
        "Outside RH [%]",
        "Outside temperature [oC]",
        "Summary",
        "Heat production heifers (not pregnant) (hpu)",
        "Heifers (pregnant)",
        "Milk [% fat]",
        "Dry cows",
        "Inside temperature [oC]",
        "Heat production dry cows (hpu)",
        "Milking cows",
        "Inside RH [%]"
    ]
}
```


```python
print(json.dumps(default_pco2_parameters, indent=4))
```


```
{
    "melkvee": {
```

```
        "gewicht": 650,
        "drachtdagen": 160
    },
    "droogstaande koeien": {
        "gewicht": 650,
        "drachtdagen": 220,
        "melkproductie": 0
    },
    "drachtig jongvee": {
        "gewicht": 400,
        "drachtdagen": 140,
        "energievoeding": 10.0,
        "gewichtstoename": 0.6
    },
    "niet drachtig jongvee": {
        "gewicht": 250,
        "drachtdagen": 0,
        "energievoeding": 10.0,
        "gewichtstoename": 0.6
    }
}
```

---

source

extract_production_column_names

```
extract_production_column_names (data:pandas.core.frame.DataFrame)
```

Extract column names for the co2 production columns from the DataFrame

	Type	Details
data	DataFrame	DataFrame with measurement data
Returns	dict	

```
print(json.dumps(extract_production_column_names(vera_dataframe), indent=4))
#extract_production_column_names(vera_dataframe)
```

```
{
    "melkvee": {
        "gewicht": [
            "Weight milking cows [kg]"
        ],
        "drachtdagen": [
            "Days in pregnancy (milking cows)"
        ],
        "melkproductie": [
            "Milk production [kg/animal/day]"
        ],
        "aantal": [
            "Milking cows"
        ]
    },
```

```
    "droogstaande koeien": {
        "gewicht": [
            "Weight dry cows [kg]"
        ],
        "drachtdagen": [
            "Days in pregnancy (dry cows)"
        ],
        "aantal": [
            "Dry cows"
        ]
    },
    "drachtig jongvee": {
        "gewicht": [
            "Weight heifers (pregnant) [kg]"
        ],
        "drachtdagen": [
            "Days in pregnancy (heifers)"
        ],
        "energievoeding": [
            "Energy value of feed (heifers; MJ/kg dry matter)"
        ],
        "gewichtstoename": [
            "Weight gain heifers [kg/day]"
        ],
        "aantal": [
            "Heifers (pregnant)"
        ]
    },
    "niet drachtig jongvee": {
        "gewicht": [
            "Weight heifers (not pregnant) [kg]"
        ],
        "energievoeding": [
            "Energy value of feed (heifers; MJ/kg dry matter)"
        ],
        "gewichtstoename": [
            "Weight gain heifers [kg/day]"
        ],
        "aantal": [
            "Heifers (not pregnant)"
        ]
    }
}
```

```
print(json.dumps(flatten_column_mapping(extract_production_column_names(vera_dataframe)),
indent=4))
```

```
[
    "Weight milking cows [kg]",
    "Days in pregnancy (milking cows)",
    "Milk production [kg/animal/day]",
    "Milking cows",
    "Weight dry cows [kg]",
    "Days in pregnancy (dry cows)",
    "Dry cows",
    "Weight heifers (pregnant) [kg]",
    "Days in pregnancy (heifers)",
    "Energy value of feed (heifers; MJ/kg dry matter)",
    "Weight gain heifers [kg/day]",
    "Heifers (pregnant)",
    "Weight heifers (not pregnant) [kg]",
```

```
    "Energy value of feed (heifers; MJ/kg dry matter)",
    "Weight gain heifers [kg/day]",
    "Heifers (not pregnant)"
]
```

```
vera_dataframe[flatten_column_mapping(extract_production_column_names(vera_dataframe))]
```

Date	Weight milking cows [kg]	Days in pregnancy (milking cows)	Milk production [kg/animal/day]	Milking cows	Weight dry cows [kg]	Days in pregnancy (dry cows)	Dry cows	Weight heifers (pregnant) [kg]	Days in pregnancy (heifers)	Energy value of feed (heifers; MJ/kg dry matter)	Weight gain heifers [kg/day]	Heifers (pregnant)	Weight heifers (not pregnant) [kg]	Energy value of feed (heifers; MJ/kg dry matter)	Weight gain heifers [kg/day]	Heifers (not pregnant)
2011-04-14	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-04-15	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-04-16	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-06-16	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-06-17	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-06-18	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-08-12	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-08-13	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-08-14	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-10-16	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-10-17	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-10-18	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-11-24	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-11-25	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-11-26	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2012-01-24	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2012-01-25	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2012-01-26	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-05-11	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15

Date	Weight milking cows [kg]	Days in pregnancy (milking cows)	Milk production [kg/animal/day]	Milking cows	Weight dry cows [kg]	Days in pregnancy (dry cows)	Dry cows	Weight heifers (pregnant) [kg]	Days in pregnancy (heifers)	Energy value of feed (heifers; MJ/kg dry matter)	Weight gain heifers [kg/day]	Heifers (pregnant)	Weight heifers (not pregnant) [kg]	Energy value of feed (heifers; MJ/kg dry matter)	Weight gain heifers [kg/day]	Heifers (not pregnant)
2011-07-06	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-09-07	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-10-26	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-12-08	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2012-02-16	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-05-03	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-06-27	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-08-31	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-11-02	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-12-15	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2012-02-14	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-04-12	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-06-15	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-08-17	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-10-12	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-12-01	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2012-01-31	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-04-04	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-04-05	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-04-06	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-06-06	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-06-07	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-06-08	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-08-02	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15

Date	Weight milking cows [kg]	Days in pregnancy (milking cows)	Milk production [kg/animal/day]	Milking cows	Weight dry cows [kg]	Days in pregnancy (dry cows)	Dry cows	Weight heifers (pregnant) [kg]	Days in pregnancy (heifers)	Energy value of feed (heifers; MJ/kg dry matter)	Weight gain heifers [kg/day]	Heifers (pregnant)	Weight heifers (not pregnant) [kg]	Energy value of feed (heifers; MJ/kg dry matter)	Weight gain heifers [kg/day]	Heifers (not pregnant)
2011-08-03	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-08-04	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-10-06	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-10-07	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-10-08	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-11-24	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-11-25	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-11-26	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2012-01-24	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2012-01-25	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2012-01-26	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-05-11	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-07-06	650	160	30.0	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-09-07	650	160	NaN	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-10-26	650	160	NaN	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2011-12-08	650	160	NaN	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15
2012-02-11	650	160	NaN	110	650	220	13	400	140	10	0.6	14	250	10	0.6	15

Emissie data

source

find_emission_column_names

```
find_emission_column_names (data:pandas.core.frame.DataFrame)
```

Find column names for NH3, CO2 and temperature from the DataFrame

```python
print(json.dumps(find_emission_column_names(vera_dataframe), indent=3))
```


```
{
   "nh3": [
      "NH3 inside [mg/m3]",
      "NH3 outside [mg/m3]",
      "NH3 Emission [kg/year per animal place]"
   ],
   "co2": [
      "CO2 inside [ppm]",
      "CO2 outside [ppm]"
   ],
   "temp": [
      "Outside temperature [oC]",
      "Inside temperature [oC]",
      "Total heat production corrected for temperature (hpu)"
   ],
   "rh": [
      "Outside RH [%]",
      "Inside RH [%]"
   ],
   "wind": []
}
```


---

source

extract_emission_column_names

```
extract_emission_column_names (data:pandas.core.frame.DataFrame)
```

Extract column names for NH3, CO2 and temperature from the DataFrame

	Type	Details
data	DataFrame	DataFrame with measurement data
Returns	dict	

```python
column_mapping = extract_emission_column_names(vera_dataframe)
```

```python
print(json.dumps(column_mapping, indent=2)) #extract_column_names(vera_dataframe)
```


```
{
  "binnen": {
    "nh3": [
      "NH3 inside [mg/m3]"
    ],
    "co2": [
      "CO2 inside [ppm]"
    ],
    "temp": [
      "Inside temperature [oC]"
```

```
    ],
    "rh": [
      "Inside RH [%]"
    ],
    "wind": []
  },
  "buiten": {
    "nh3": [
      "NH3 outside [mg/m3]"
    ],
    "co2": [
      "CO2 outside [ppm]"
    ],
    "temp": [
      "Outside temperature [oC]"
    ],
    "rh": [
      "Outside RH [%]"
    ],
    "wind": []
  }
}
```

```
[col for loc, measures in column_mapping.items() for measure, cols in measures.items() for
col in cols]
```

```
['NH3 inside [mg/m3]',
 'CO2 inside [ppm]',
 'Inside temperature [oC]',
 'Inside RH [%]',
 'NH3 outside [mg/m3]',
 'CO2 outside [ppm]',
 'Outside temperature [oC]',
 'Outside RH [%]']
```

```
flatten_column_mapping(column_mapping)
```

```
['NH3 inside [mg/m3]',
 'CO2 inside [ppm]',
 'Inside temperature [oC]',
 'Inside RH [%]',
 'NH3 outside [mg/m3]',
 'CO2 outside [ppm]',
 'Outside temperature [oC]',
 'Outside RH [%]']
```

```
vera_dataframe[flatten_column_mapping(column_mapping)]
```

Date	NH3 inside [mg/m3]	CO2 inside [ppm]	Inside temperature [oC]	Inside RH [%]	NH3 outside [mg/m3]	CO2 outside [ppm]	Outside temperature [oC]	Outside RH [%]
2011-04-04	3.970204	1063	17.9	72.0	0	578.0	13.5	76.0
2011-04-05	3.970204	1062	17.8	71.0	0	576.0	13.5	76.0
2011-04-06	4.039857	1061	17.8	71.0	0	571.0	13.5	76.0
2011-06-06	3.970204	1060	17.8	70.0	0	569.0	13.5	76.0
2011-06-07	4.039857	1059	17.8	71.0	0	570.0	13.5	76.0
2011-06-08	3.970204	1056	NaN	NaN	0	568.0	13.4	77.0
2011-08-02	3.970204	1056	17.7	72.0	0	568.0	13.4	77.0
2011-08-03	3.970204	1056	17.7	72.0	0	568.0	13.4	77.0
2011-08-04	3.970204	1057	17.7	72.0	0	565.0	13.4	77.0
2011-10-06	4.039857	1057	17.7	72.0	0	566.0	13.4	77.0
2011-10-07	3.970204	1054	17.7	72.0	0	566.0	13.4	77.0
2011-10-08	3.970204	1058	17.7	72.0	0	565.0	13.3	77.0
2011-11-24	3.970204	1058	17.7	73.0	0	566.0	13.3	77.0
2011-11-25	3.970204	1051	17.8	73.0	0	565.0	13.3	77.0
2011-11-26	3.970204	1051	17.7	72.0	0	561.0	13.3	77.0
2012-01-24	3.970204	1054	17.7	72.0	0	563.0	13.3	77.0
2012-01-25	3.970204	1059	17.7	71.0	0	563.0	13.3	77.0
2012-01-26	3.970204	1055	17.7	72.0	0	565.0	13.3	77.0
2011-05-11	3.970204	1058	17.7	72.0	0	568.0	13.3	77.0
2011-07-06	3.970204	1058	17.7	72.0	0	567.0	13.3	77.0
2011-09-07	3.970204	1058	17.7	72.0	0	569.0	13.3	77.0
2011-10-26	3.970204	1061	17.7	73.0	0	568.0	13.2	77.0
2011-12-08	3.970204	1058	17.7	73.0	0	570.0	13.2	78.0
2012-02-16	3.970204	1059	17.7	73.0	0	574.0	13.2	NaN
2011-05-03	3.970204	1059	17.7	72.0	0	575.0	NaN	NaN
2011-06-27	3.970204	1061	17.8	72.0	0	573.0	NaN	NaN
2011-08-31	3.970204	1062	17.8	72.0	0	574.0	NaN	NaN

Date	NH3 inside [mg/m3]	CO2 inside [ppm]	Inside temperature [oC]	Inside RH [%]	NH3 outside [mg/m3]	CO2 outside [ppm]	Outside temperature [oC]	Outside RH [%]
2011-11-02	3.970204	1063	17.8	71.0	0	574.0	NaN	NaN
2011-12-15	3.970204	1063	17.7	71.0	0	571.0	NaN	NaN
2012-02-14	3.970204	1068	17.6	71.0	0	570.0	NaN	NaN
2011-04-12	2.830898	1070	17.6	71.0	0	568.0	NaN	NaN
2011-06-15	5.761246	1069	17.5	71.0	0	566.0	NaN	NaN
2011-08-17	7.691593	1070	17.4	70.0	0	565.0	NaN	NaN
2011-10-12	NaN	1069	17.3	71.0	0	563.0	NaN	NaN
2011-12-01	NaN	1067	17.3	71.0	0	563.0	NaN	NaN
2012-01-31	NaN	1062	17.3	71.0	0	561.0	NaN	NaN
2011-04-04	NaN	1059	17.3	71.0	0	560.0	NaN	NaN
2011-04-05	NaN	1047	17.2	72.0	0	561.0	NaN	NaN
2011-04-06	NaN	1034	17.3	72.0	0	558.0	NaN	NaN
2011-06-06	NaN	1027	17.3	73.0	0	555.0	NaN	NaN
2011-06-07	NaN	1016	17.4	73.0	0	553.0	NaN	NaN
2011-06-08	NaN	1017	17.5	73.0	0	555.0	NaN	NaN
2011-08-02	NaN	1017	17.5	73.0	0	556.0	NaN	NaN
2011-08-03	NaN	1018	17.6	73.0	0	555.0	NaN	NaN
2011-08-04	NaN	1019	17.6	73.0	0	554.0	NaN	NaN
2011-10-06	NaN	1026	17.7	NaN	0	551.0	NaN	NaN
2011-10-07	NaN	1027	17.7	NaN	0	554.0	NaN	NaN
2011-10-08	NaN	1028	17.7	NaN	0	553.0	NaN	NaN
2011-11-24	NaN	1030	17.8	NaN	0	554.0	NaN	NaN
2011-11-25	NaN	1031	17.8	NaN	0	558.0	NaN	NaN
2011-11-26	NaN	1035	17.8	NaN	0	559.0	NaN	NaN
2012-01-24	NaN	1045	17.8	NaN	0	557.0	NaN	NaN
2012-01-25	NaN	1048	17.7	NaN	0	NaN	NaN	NaN
2012-01-26	NaN	1060	17.7	NaN	0	NaN	NaN	NaN

Date	NH3 inside [mg/m3]	CO2 inside [ppm]	Inside temperature [oC]	Inside RH [%]	NH3 outside [mg/m3]	CO2 outside [ppm]	Outside temperature [oC]	Outside RH [%]
2011-05-11	NaN	1066	17.7	NaN	0	NaN	NaN	NaN
2011-07-06	NaN	1076	17.7	NaN	0	NaN	NaN	NaN
2011-09-07	NaN	1073	17.7	NaN	0	NaN	NaN	NaN
2011-10-26	NaN	1079	17.8	NaN	0	NaN	NaN	NaN
2011-12-08	NaN	1077	17.7	NaN	0	NaN	NaN	NaN
2012-02-16	NaN	1079	17.7	NaN	0	NaN	NaN	NaN

Opschonen van data

Een randvoorwaarde van de ratiomethode is dat de concentraties van NH_3 en het tracergas - in dit geval CO_2 - op dezelfde meetpunten en met dezelfde meetfrequentie gemeten moeten worden. Om een goede schatting van de emissie te verkrijgen is het van belang dat de concentratieratio's per meetpunt worden geschat en daarna een gemiddeld van deze waarden wordt genomen, in plaats dan eerst een gemiddelde concentratie van al die punten te bepalen en daarna de ratiomethode te gebruiken.

We verwachten dat de metingen van veschillende sensoren komen en op verschillende tijdstippen zijn gedaan. Om te kunnen rekenen moeten rijen volledig gevult zijn. We kunnen dit doen door de data te resamplen op een vast tijdsinterval (bijv. 10 minuten).

---

source

resample_data

```
resample_data (data:pandas.core.frame.DataFrame, interval:str,
               method:str)
```

Resample data to a specified interval and interpolate missing values with the givien method

	Type	Details
data	DataFrame	DataFrame with measurement data

	Type	Details
interval	str	resampling interval (e.g. '10min' for 10 minutes)
method	str	resampling method (e.g. 'linear', 'cubic')
Returns	DataFrame	

CO_2 productie

```
extract_production_column_names(vera_dataframe)
```

```
{'melkvee': {'gewicht': ['Weight milking cows [kg]'],
  'drachtdagen': ['Days in pregnancy (milking cows)'],
  'melkproductie': ['Milk production [kg/animal/day]'],
  'aantal': ['Milking cows']},
 'droogstaande koeien': {'gewicht': ['Weight dry cows [kg]'],
  'drachtdagen': ['Days in pregnancy (dry cows)'],
  'aantal': ['Dry cows']},
 'drachtig jongvee': {'gewicht': ['Weight heifers (pregnant) [kg]'],
  'drachtdagen': ['Days in pregnancy (heifers)'],
  'energievoeding': ['Energy value of feed (heifers; MJ/kg dry matter)'],
  'gewichtstoename': ['Weight gain heifers [kg/day]'],
  'aantal': ['Heifers (pregnant)']},
 'niet drachtig jongvee': {'gewicht': ['Weight heifers (not pregnant) [kg]'],
  'energievoeding': ['Energy value of feed (heifers; MJ/kg dry matter)'],
  'gewichtstoename': ['Weight gain heifers [kg/day]'],
  'aantal': ['Heifers (not pregnant)']}}
```

```
for category, params in pco2_category_functions_parameters.items():
    print(f"Category: {category}")
    for param, param_info in params.items():
        print(f"  Parameter: {param}, Type: {param_info.annotation}, Default:
{param_info.default}")
```

```
Category: melkvee
  Parameter: aantal, Type: <class 'inspect._empty'>, Default: <class 'inspect._empty'>
  Parameter: melkproductie, Type: <class 'inspect._empty'>, Default: <class
'inspect._empty'>
  Parameter: drachtdagen, Type: <class 'inspect._empty'>, Default: <class 'inspect._empty'>
  Parameter: gewicht, Type: <class 'inspect._empty'>, Default: <class 'inspect._empty'>
Category: droogstaande koeien
  Parameter: aantal, Type: <class 'inspect._empty'>, Default: <class 'inspect._empty'>
  Parameter: melkproductie, Type: <class 'inspect._empty'>, Default: <class
'inspect._empty'>
  Parameter: drachtdagen, Type: <class 'inspect._empty'>, Default: <class 'inspect._empty'>
  Parameter: gewicht, Type: <class 'inspect._empty'>, Default: <class 'inspect._empty'>
Category: drachtig jongvee
  Parameter: aantal, Type: <class 'inspect._empty'>, Default: <class 'inspect._empty'>
  Parameter: energievoeding, Type: <class 'inspect._empty'>, Default: <class
'inspect._empty'>
  Parameter: drachtdagen, Type: <class 'inspect._empty'>, Default: <class 'inspect._empty'>
  Parameter: gewicht, Type: <class 'inspect._empty'>, Default: <class 'inspect._empty'>
  Parameter: gewichtstoename, Type: <class 'inspect._empty'>, Default: <class
'inspect._empty'>
Category: niet drachtig jongvee
  Parameter: aantal, Type: <class 'inspect._empty'>, Default: <class 'inspect._empty'>
  Parameter: energievoeding, Type: <class 'inspect._empty'>, Default: <class
'inspect._empty'>
```

```
  Parameter: drachtdagen, Type: <class 'inspect._empty'>, Default: <class 'inspect._empty'>
  Parameter: gewicht, Type: <class 'inspect._empty'>, Default: <class 'inspect._empty'>
  Parameter: gewichtstoename, Type: <class 'inspect._empty'>, Default: <class
'inspect._empty'>
```

```
default_pco2_parameters
```

```
{'melkvee': {'gewicht': 650, 'drachtdagen': 160},
 'droogstaande koeien': {'gewicht': 650,
  'drachtdagen': 220,
  'melkproductie': 0},
 'drachtig jongvee': {'gewicht': 400,
  'drachtdagen': 140,
  'energievoeding': 10.0,
  'gewichtstoename': 0.6},
 'niet drachtig jongvee': {'gewicht': 250,
  'drachtdagen': 0,
  'energievoeding': 10.0,
  'gewichtstoename': 0.6}}
```

---

source

calculate_pco2_production_from_data

```
calculate_pco2_production_from_data (data:pandas.core.frame.DataFrame,
                                     pco2_parameters={'melkvee':
                                     {'gewicht': 650, 'drachtdagen':
                                     160}, 'droogstaande koeien':
                                     {'gewicht': 650, 'drachtdagen': 220,
                                     'melkproductie': 0}, 'drachtig
                                     jongvee': {'gewicht': 400,
                                     'drachtdagen': 140,
                                     'energievoeding': 10.0,
                                     'gewichtstoename': 0.6}, 'niet
                                     drachtig jongvee': {'gewicht': 250,
                                     'drachtdagen': 0, 'energievoeding':
                                     10.0, 'gewichtstoename': 0.6}})
```

```
calculate_pco2_production_from_data(vera_dataframe)
```

	PCO2_melkvee	PCO2_droogstaande koeien	PCO2_drachtig jongvee	PCO2_niet drachtig jongvee
Date				
2011-04-04	31.82152	2.317288	1.891889	1.380852
2011-04-05	31.82152	2.317288	1.891889	1.380852
2011-04-06	31.82152	2.317288	1.891889	1.380852
2011-06-06	31.82152	2.317288	1.891889	1.380852

Date	PCO2_melkvee	PCO2_droogstaande koeien	PCO2_drachtig jongvee	PCO2_niet drachtig jongvee
2011-06-07	31.82152	2.317288	1.891889	1.380852
2011-06-08	31.82152	2.317288	1.891889	1.380852
2011-08-02	31.82152	2.317288	1.891889	1.380852
2011-08-03	31.82152	2.317288	1.891889	1.380852
2011-08-04	31.82152	2.317288	1.891889	1.380852
2011-10-06	31.82152	2.317288	1.891889	1.380852
2011-10-07	31.82152	2.317288	1.891889	1.380852
2011-10-08	31.82152	2.317288	1.891889	1.380852
2011-11-24	31.82152	2.317288	1.891889	1.380852
2011-11-25	31.82152	2.317288	1.891889	1.380852
2011-11-26	31.82152	2.317288	1.891889	1.380852
2012-01-24	31.82152	2.317288	1.891889	1.380852
2012-01-25	31.82152	2.317288	1.891889	1.380852
2012-01-26	31.82152	2.317288	1.891889	1.380852
2011-05-11	31.82152	2.317288	1.891889	1.380852
2011-07-06	31.82152	2.317288	1.891889	1.380852
2011-09-07	31.82152	2.317288	1.891889	1.380852
2011-10-26	31.82152	2.317288	1.891889	1.380852
2011-12-08	31.82152	2.317288	1.891889	1.380852
2012-02-16	31.82152	2.317288	1.891889	1.380852
2011-05-03	31.82152	2.317288	1.891889	1.380852
2011-06-27	31.82152	2.317288	1.891889	1.380852
2011-08-31	31.82152	2.317288	1.891889	1.380852
2011-11-02	31.82152	2.317288	1.891889	1.380852
2011-12-15	31.82152	2.317288	1.891889	1.380852
2012-02-14	31.82152	2.317288	1.891889	1.380852
2011-04-12	31.82152	2.317288	1.891889	1.380852
2011-06-15	31.82152	2.317288	1.891889	1.380852
2011-08-17	31.82152	2.317288	1.891889	1.380852

Date	PCO2_melkvee	PCO2_droogstaande koeien	PCO2_drachtig jongvee	PCO2_niet drachtig jongvee
2011-10-12	31.82152	2.317288	1.891889	1.380852
2011-12-01	31.82152	2.317288	1.891889	1.380852
2012-01-31	31.82152	2.317288	1.891889	1.380852
2011-04-04	31.82152	2.317288	1.891889	1.380852
2011-04-05	31.82152	2.317288	1.891889	1.380852
2011-04-06	31.82152	2.317288	1.891889	1.380852
2011-06-06	31.82152	2.317288	1.891889	1.380852
2011-06-07	31.82152	2.317288	1.891889	1.380852
2011-06-08	31.82152	2.317288	1.891889	1.380852
2011-08-02	31.82152	2.317288	1.891889	1.380852
2011-08-03	31.82152	2.317288	1.891889	1.380852
2011-08-04	31.82152	2.317288	1.891889	1.380852
2011-10-06	31.82152	2.317288	1.891889	1.380852
2011-10-07	31.82152	2.317288	1.891889	1.380852
2011-10-08	31.82152	2.317288	1.891889	1.380852
2011-11-24	31.82152	2.317288	1.891889	1.380852
2011-11-25	31.82152	2.317288	1.891889	1.380852
2011-11-26	31.82152	2.317288	1.891889	1.380852
2012-01-24	31.82152	2.317288	1.891889	1.380852
2012-01-25	31.82152	2.317288	1.891889	1.380852
2012-01-26	31.82152	2.317288	1.891889	1.380852
2011-05-11	31.82152	2.317288	1.891889	1.380852
2011-07-06	31.82152	2.317288	1.891889	1.380852
2011-09-07	NaN	2.317288	1.891889	1.380852
2011-10-26	NaN	2.317288	1.891889	1.380852
2011-12-08	NaN	2.317288	1.891889	1.380852
2012-02-16	NaN	2.317288	1.891889	1.380852

```
vera_dataframe.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 60 entries, 2011-04-04 to 2012-02-16
Data columns (total 55 columns):
 #   Column                                                 Non-Null Count  Dtype  
---  ------                                                 --------------  -----  
 0   Measurement institute                                  60 non-null     object 
 1   Animal Category                                        60 non-null     object 
 2   Housing system                                         60 non-null     object 
 3   Measurement location                                   60 non-null     object 
 4   Measurement period                                     60 non-null     int64  
 5   Measurement day (in period)                            60 non-null     int64  
 6   Day in year                                            60 non-null     int64  
 7   Outside temperature [oC]                               24 non-null     float64
 8   Outside RH [%]                                         23 non-null     float64
 9   Inside temperature [oC]                                59 non-null     float64
 10  Inside RH [%]                                          44 non-null     float64
 11  Animal places                                          60 non-null     int64  
 12  Milking cows                                           60 non-null     int64  
 13  Dry cows                                               60 non-null     int64  
 14  Heifers (pregnant)                                     60 non-null     int64  
 15  Heifers (not pregnant)                                 60 non-null     int64  
 16  Floor type (0: slatted floor; 1: closed floor)         60 non-null     int64  
 17  Walking area per animal (m2)                           60 non-null     float64
 18  Grazing (hours per day)                                60 non-null     int64  
 19  Closed cubicles                                        60 non-null     int64  
 20  Milk production [kg/animal/day]                        56 non-null     float64
 21  Milk [% protein]                                       60 non-null     float64
 22  Milk [% fat]                                           60 non-null     float64
 23  Urea content in milk [mg/100g]                         56 non-null     float64
 24  Weight milking cows [kg]                               60 non-null     int64  
 25  Weight dry cows [kg]                                   60 non-null     int64  
 26  Weight heifers (pregnant) [kg]                         60 non-null     int64  
 27  Weight heifers (not pregnant) [kg]                     60 non-null     int64  
 28  Days in pregnancy (milking cows)                       60 non-null     int64  
 29  Days in pregnancy (dry cows)                           60 non-null     int64  
 30  Days in pregnancy (heifers)                            60 non-null     int64  
 31  Energy value of feed (heifers; MJ/kg dry matter)       60 non-null     int64  
 32  Weight gain heifers [kg/day]                           60 non-null     float64
 33  CO2 inside [ppm]                                       60 non-null     int64  
 34  CO2 outside [ppm]                                      52 non-null     float64
 35  NH3 inside [mg/m3]                                     33 non-null     float64
 36  NH3 outside [mg/m3]                                    60 non-null     int64  
 37  Number of animals                                      60 non-null     int64  
 38  Dairy cows (milking + dry)                             60 non-null     int64  
 39  % closed cubicles                                      60 non-null     float64
 40  Occupation rate (%)                                    60 non-null     float64
 41  Dairy cows (%)                                         60 non-null     float64
 42  Heifers vs. dairy cows (%)                             60 non-null     float64
 43  Dry cows vs. dairy cows (%)                            60 non-null     float64
 44  Heat production milking cows (hpu)                     56 non-null     float64
 45  Heat production dry cows (hpu)                         56 non-null     float64
 46  Heat production heifers (pregnant) (hpu)               60 non-null     float64
 47  Heat production heifers (not pregnant) (hpu)           60 non-null     float64
 48  Total heat production (hpu)                            60 non-null     float64
 49  Total heat production corrected for temperature (hpu)  59 non-null     float64
 50  Ventilation rate [m3/h]                                51 non-null     float64
 51  Ventilation rate [m3/h per animal]                     51 non-null     float64
 52  NH3 Emission [kg/year per animal place]                32 non-null     float64
 53  Summary                                                32 non-null     float64
 54  Summary.1                                              32 non-null     float64
dtypes: float64(28), int64(23), object(4)
memory usage: 26.2+ KB
```

Emissie ratio

source

calculate_emission_ratio

```
calculate_emission_ratio (NH3_stal, NH3_buiten, CO2_stal, CO2_buiten)
```

Calculate the emission ratio

	Details
NH3_stal	NH3 concentration in the barn in mg/m3
NH3_buiten	NH3 concentration outside in mg/m3
CO2_stal	CO2 concentration in the barn in ppm
CO2_buiten	CO2 concentration outside in ppm

With ratios calculated we can calculate the NH_3 emission.

Uiteindelijke berekenening module

```
columnmapping = extract_emission_column_names(vera_dataframe)
temperatuur = data[columnmapping['binnen']['temp']].mean(axis=1)
temperatuur
```

```
Date
2011-04-04    17.9
2011-04-05    17.8
2011-04-06    17.8
2011-06-06    17.8
2011-06-07    17.8
2011-06-08     NaN
2011-08-02    17.7
2011-08-03    17.7
2011-08-04    17.7
2011-10-06    17.7
2011-10-07    17.7
2011-10-08    17.7
2011-11-24    17.7
2011-11-25    17.8
2011-11-26    17.7
2012-01-24    17.7
2012-01-25    17.7
2012-01-26    17.7
2011-05-11    17.7
2011-07-06    17.7
2011-09-07    17.7
2011-10-26    17.7
2011-12-08    17.7
2012-02-16    17.7
2011-05-03    17.7
2011-06-27    17.8
2011-08-31    17.8
2011-11-02    17.8
```

```
2011-12-15    17.7
2012-02-14    17.6
2011-04-12    17.6
2011-06-15    17.5
2011-08-17    17.4
2011-10-12    17.3
2011-12-01    17.3
2012-01-31    17.3
2011-04-04    17.3
2011-04-05    17.2
2011-04-06    17.3
2011-06-06    17.3
2011-06-07    17.4
2011-06-08    17.5
2011-08-02    17.5
2011-08-03    17.6
2011-08-04    17.6
2011-10-06    17.7
2011-10-07    17.7
2011-10-08    17.7
2011-11-24    17.8
2011-11-25    17.8
2011-11-26    17.8
2012-01-24    17.8
2012-01-25    17.7
2012-01-26    17.7
2011-05-11    17.7
2011-07-06    17.7
2011-09-07    17.7
2011-10-26    17.8
2011-12-08    17.7
2012-02-16    17.7
dtype: float64
```

```
pco2_calculated =     calculate_pco2_production_from_data(data)
pco2_calculated.sum(axis=1).rename('PCO2 totaal')
```

```
Date
2011-04-04    37.411548
2011-04-05    37.411548
2011-04-06    37.411548
2011-06-06    37.411548
2011-06-07    37.411548
2011-06-08    37.411548
2011-08-02    37.411548
2011-08-03    37.411548
2011-08-04    37.411548
2011-10-06    37.411548
2011-10-07    37.411548
2011-10-08    37.411548
2011-11-24    37.411548
2011-11-25    37.411548
2011-11-26    37.411548
2012-01-24    37.411548
2012-01-25    37.411548
2012-01-26    37.411548
2011-05-11    37.411548
2011-07-06    37.411548
2011-09-07    37.411548
2011-10-26    37.411548
2011-12-08    37.411548
2012-02-16    37.411548
```

```
2011-05-03    37.411548
2011-06-27    37.411548
2011-08-31    37.411548
2011-11-02    37.411548
2011-12-15    37.411548
2012-02-14    37.411548
2011-04-12    37.411548
2011-06-15    37.411548
2011-08-17    37.411548
2011-10-12    37.411548
2011-12-01    37.411548
2012-01-31    37.411548
2011-04-04    37.411548
2011-04-05    37.411548
2011-04-06    37.411548
2011-06-06    37.411548
2011-06-07    37.411548
2011-06-08    37.411548
2011-08-02    37.411548
2011-08-03    37.411548
2011-08-04    37.411548
2011-10-06    37.411548
2011-10-07    37.411548
2011-10-08    37.411548
2011-11-24    37.411548
2011-11-25    37.411548
2011-11-26    37.411548
2012-01-24    37.411548
2012-01-25    37.411548
2012-01-26    37.411548
2011-05-11    37.411548
2011-07-06    37.411548
2011-09-07     5.590028
2011-10-26     5.590028
2011-12-08     5.590028
2012-02-16     5.590028
Name: PCO2 totaal, dtype: float64
```

```
pco2_calculated.sum(axis=1).rename('PCO2 totaal') * 5
```

```
Date
2011-04-04    187.057740
2011-04-05    187.057740
2011-04-06    187.057740
2011-06-06    187.057740
2011-06-07    187.057740
2011-06-08    187.057740
2011-08-02    187.057740
2011-08-03    187.057740
2011-08-04    187.057740
2011-10-06    187.057740
2011-10-07    187.057740
2011-10-08    187.057740
2011-11-24    187.057740
2011-11-25    187.057740
2011-11-26    187.057740
2012-01-24    187.057740
2012-01-25    187.057740
2012-01-26    187.057740
2011-05-11    187.057740
2011-07-06    187.057740
2011-09-07    187.057740
```

```
2011-10-26    187.057740
2011-12-08    187.057740
2012-02-16    187.057740
2011-05-03    187.057740
2011-06-27    187.057740
2011-08-31    187.057740
2011-11-02    187.057740
2011-12-15    187.057740
2012-02-14    187.057740
2011-04-12    187.057740
2011-06-15    187.057740
2011-08-17    187.057740
2011-10-12    187.057740
2011-12-01    187.057740
2012-01-31    187.057740
2011-04-04    187.057740
2011-04-05    187.057740
2011-04-06    187.057740
2011-06-06    187.057740
2011-06-07    187.057740
2011-06-08    187.057740
2011-08-02    187.057740
2011-08-03    187.057740
2011-08-04    187.057740
2011-10-06    187.057740
2011-10-07    187.057740
2011-10-08    187.057740
2011-11-24    187.057740
2011-11-25    187.057740
2011-11-26    187.057740
2012-01-24    187.057740
2012-01-25    187.057740
2012-01-26    187.057740
2011-05-11    187.057740
2011-07-06    187.057740
2011-09-07     27.950142
2011-10-26     27.950142
2011-12-08     27.950142
2012-02-16     27.950142
Name: PCO2 totaal, dtype: float64
```

```
pco2_corrected =
pd.concat([calculate_temperatuur_correctie(temperatuur).rename('temperatuur_correctie') ,
pco2_calculated.sum(axis=1).rename('PCO2_totaal')], axis=1)
```

```
pco2_corrected
```

	temperatuur_correctie	PCO2_totaal
Date		
2011-04-04	1.0084	37.411548
2011-04-05	1.0088	37.411548
2011-04-06	1.0088	37.411548
2011-06-06	1.0088	37.411548
2011-06-07	1.0088	37.411548

Date	temperatuur_correctie	PCO2_totaal
2011-06-08	NaN	37.411548
2011-08-02	1.0092	37.411548
2011-08-03	1.0092	37.411548
2011-08-04	1.0092	37.411548
2011-10-06	1.0092	37.411548
2011-10-07	1.0092	37.411548
2011-10-08	1.0092	37.411548
2011-11-24	1.0092	37.411548
2011-11-25	1.0088	37.411548
2011-11-26	1.0092	37.411548
2012-01-24	1.0092	37.411548
2012-01-25	1.0092	37.411548
2012-01-26	1.0092	37.411548
2011-05-11	1.0092	37.411548
2011-07-06	1.0092	37.411548
2011-09-07	1.0092	37.411548
2011-10-26	1.0092	37.411548
2011-12-08	1.0092	37.411548
2012-02-16	1.0092	37.411548
2011-05-03	1.0092	37.411548
2011-06-27	1.0088	37.411548
2011-08-31	1.0088	37.411548
2011-11-02	1.0088	37.411548
2011-12-15	1.0092	37.411548
2012-02-14	1.0096	37.411548
2011-04-12	1.0096	37.411548
2011-06-15	1.0100	37.411548
2011-08-17	1.0104	37.411548
2011-10-12	1.0108	37.411548
2011-12-01	1.0108	37.411548
2012-01-31	1.0108	37.411548

	temperatuur_correctie	PCO2_totaal
Date		
2011-04-04	1.0108	37.411548
2011-04-05	1.0112	37.411548
2011-04-06	1.0108	37.411548
2011-06-06	1.0108	37.411548
2011-06-07	1.0104	37.411548
2011-06-08	1.0100	37.411548
2011-08-02	1.0100	37.411548
2011-08-03	1.0096	37.411548
2011-08-04	1.0096	37.411548
2011-10-06	1.0092	37.411548
2011-10-07	1.0092	37.411548
2011-10-08	1.0092	37.411548
2011-11-24	1.0088	37.411548
2011-11-25	1.0088	37.411548
2011-11-26	1.0088	37.411548
2012-01-24	1.0088	37.411548
2012-01-25	1.0092	37.411548
2012-01-26	1.0092	37.411548
2011-05-11	1.0092	37.411548
2011-07-06	1.0092	37.411548
2011-09-07	1.0092	5.590028
2011-10-26	1.0088	5.590028
2011-12-08	1.0092	5.590028
2012-02-16	1.0092	5.590028

```
pco2_corrected['PCO2_corrected'] = pco2_corrected['PCO2_totaal'] *
pco2_corrected['temperatuur_correctie']
```

Let op, bij berekening volgens Wagenngen is er een factor 0.2 in de CO_2 productie die door de werkboeken pas wordt toegepat bij de debiet berekening.

```
pco2_corrected['PCO2_corrected'] * 5
```

```
Date
2011-04-04    188.629025
2011-04-05    188.703848
2011-04-06    188.703848
2011-06-06    188.703848
2011-06-07    188.703848
2011-06-08           NaN
2011-08-02    188.778671
2011-08-03    188.778671
2011-08-04    188.778671
2011-10-06    188.778671
2011-10-07    188.778671
2011-10-08    188.778671
2011-11-24    188.778671
2011-11-25    188.703848
2011-11-26    188.778671
2012-01-24    188.778671
2012-01-25    188.778671
2012-01-26    188.778671
2011-05-11    188.778671
2011-07-06    188.778671
2011-09-07    188.778671
2011-10-26    188.778671
2011-12-08    188.778671
2012-02-16    188.778671
2011-05-03    188.778671
2011-06-27    188.703848
2011-08-31    188.703848
2011-11-02    188.703848
2011-12-15    188.778671
2012-02-14    188.853494
2011-04-12    188.853494
2011-06-15    188.928317
2011-08-17    189.003141
2011-10-12    189.077964
2011-12-01    189.077964
2012-01-31    189.077964
2011-04-04    189.077964
2011-04-05    189.152787
2011-04-06    189.077964
2011-06-06    189.077964
2011-06-07    189.003141
2011-06-08    188.928317
2011-08-02    188.928317
2011-08-03    188.853494
2011-08-04    188.853494
2011-10-06    188.778671
2011-10-07    188.778671
2011-10-08    188.778671
2011-11-24    188.703848
2011-11-25    188.703848
2011-11-26    188.703848
2012-01-24    188.703848
2012-01-25    188.778671
2012-01-26    188.778671
2011-05-11    188.778671
2011-07-06    188.778671
2011-09-07     28.207283
2011-10-26     28.196103
2011-12-08     28.207283
2012-02-16     28.207283
Name: PCO2_corrected, dtype: float64
```

```
nh3_binnen = data[columnmapping['binnen']['nh3']].mean(axis=1).rename('nh3_binnen')
nh3_buiten = data[columnmapping['buiten']['nh3']].min(axis=1).rename('nh3_buiten')
co2_binnen = data[columnmapping['binnen']['co2']].mean(axis=1).rename('co2_binnen')
co2_buiten = data[columnmapping['buiten']['co2']].min(axis=1).rename('co2_buiten')
```

```
nh3_binnen
```

```
Date
2011-04-04    3.970204
2011-04-05    3.970204
2011-04-06    4.039857
2011-06-06    3.970204
2011-06-07    4.039857
2011-06-08    3.970204
2011-08-02    3.970204
2011-08-03    3.970204
2011-08-04    3.970204
2011-10-06    4.039857
2011-10-07    3.970204
2011-10-08    3.970204
2011-11-24    3.970204
2011-11-25    3.970204
2011-11-26    3.970204
2012-01-24    3.970204
2012-01-25    3.970204
2012-01-26    3.970204
2011-05-11    3.970204
2011-07-06    3.970204
2011-09-07    3.970204
2011-10-26    3.970204
2011-12-08    3.970204
2012-02-16    3.970204
2011-05-03    3.970204
2011-06-27    3.970204
2011-08-31    3.970204
2011-11-02    3.970204
2011-12-15    3.970204
2012-02-14    3.970204
2011-04-12    3.830898
2011-06-15    3.761246
2011-08-17    3.691593
2011-10-12         NaN
2011-12-01         NaN
2012-01-31         NaN
2011-04-04         NaN
2011-04-05         NaN
2011-04-06         NaN
2011-06-06         NaN
2011-06-07         NaN
2011-06-08         NaN
2011-08-02         NaN
2011-08-03         NaN
2011-08-04         NaN
2011-10-06         NaN
2011-10-07         NaN
2011-10-08         NaN
2011-11-24         NaN
2011-11-25         NaN
2011-11-26         NaN
2012-01-24         NaN
2012-01-25         NaN
```

```
2012-01-26         NaN
2011-05-11         NaN
2011-07-06         NaN
2011-09-07         NaN
2011-10-26         NaN
2011-12-08         NaN
2012-02-16         NaN
Name: nh3_binnen, dtype: float64
```

We volgen even de werkboeken

ventilatie 1

BC5 = Total_corrected_heat * 0.2 /(1e-6 * (co2_binnen - co2_buiten))

BD5 = BC5 / (totaal_aantal_dieren)

BE5 = BC5 8 (nh3_binnen - nh3_buiten) / 1e6 * 24*365 / (totaal_plaatsen - gesloten_plaatsen)

Berekening volgens Wageningen

```
ratio = calculate_emission_ratio(
    NH3_stal=nh3_binnen,
    NH3_buiten=nh3_buiten,
    CO2_stal=co2_binnen,
    CO2_buiten=co2_buiten
).rename('ratio')
```

```
emission = pd.concat([ratio, pco2_corrected['PCO2_corrected']], axis=1)
```

```
emission['emission'] = (emission['ratio'] * emission['PCO2_corrected']) * 24 * 365
```

```
emission
```

	ratio	PCO2_corrected	emission
Date			
2011-04-04	0.003168	37.725805	1047.076865
2011-04-05	0.003162	37.740770	1045.336883
2011-04-06	0.003190	37.740770	1054.795812
2011-06-06	0.003130	37.740770	1034.691953
2011-06-07	0.003197	37.740770	1056.952849

Date	ratio	PCO2_corrected	emission
2011-06-08	0.003149	NaN	NaN
2011-08-02	0.003149	37.755734	1041.465523
2011-08-03	0.003149	37.755734	1041.465523
2011-08-04	0.003123	37.755734	1032.998362
2011-10-06	0.003184	37.755734	1053.064947
2011-10-07	0.003149	37.755734	1041.465523
2011-10-08	0.003117	37.755734	1030.903040
2011-11-24	0.003123	37.755734	1032.998362
2011-11-25	0.003162	37.740770	1045.336883
2011-11-26	0.003136	37.755734	1037.214663
2012-01-24	0.003130	37.755734	1035.102219
2012-01-25	0.003098	37.755734	1024.667768
2012-01-26	0.003136	37.755734	1037.214663
2011-05-11	0.003136	37.755734	1037.214663
2011-07-06	0.003130	37.755734	1035.102219
2011-09-07	0.003142	37.755734	1039.335747
2011-10-26	0.003117	37.755734	1030.903040
2011-12-08	0.003149	37.755734	1041.465523
2012-02-16	0.003168	37.755734	1047.907549
2011-05-03	0.003175	37.755734	1050.072637
2011-06-27	0.003149	37.740770	1041.052735
2011-08-31	0.003149	37.740770	1041.052735
2011-11-02	0.003142	37.740770	1038.923802
2011-12-15	0.003123	37.755734	1032.998362
2012-02-14	0.003086	37.770699	1020.957154
2011-04-12	0.002953	37.770699	976.936479
2011-06-15	0.002894	37.785663	957.832483
2011-08-17	0.002828	37.800628	936.567163
2011-10-12	NaN	37.815593	NaN
2011-12-01	NaN	37.815593	NaN
2012-01-31	NaN	37.815593	NaN

	ratio	PCO2_corrected	emission
Date			
2011-04-04	NaN	37.815593	NaN
2011-04-05	NaN	37.830557	NaN
2011-04-06	NaN	37.815593	NaN
2011-06-06	NaN	37.815593	NaN
2011-06-07	NaN	37.800628	NaN
2011-06-08	NaN	37.785663	NaN
2011-08-02	NaN	37.785663	NaN
2011-08-03	NaN	37.770699	NaN
2011-08-04	NaN	37.770699	NaN
2011-10-06	NaN	37.755734	NaN
2011-10-07	NaN	37.755734	NaN
2011-10-08	NaN	37.755734	NaN
2011-11-24	NaN	37.740770	NaN
2011-11-25	NaN	37.740770	NaN
2011-11-26	NaN	37.740770	NaN
2012-01-24	NaN	37.740770	NaN
2012-01-25	NaN	37.755734	NaN
2012-01-26	NaN	37.755734	NaN
2011-05-11	NaN	37.755734	NaN
2011-07-06	NaN	37.755734	NaN
2011-09-07	NaN	5.641457	NaN
2011-10-26	NaN	5.639221	NaN
2011-12-08	NaN	5.641457	NaN
2012-02-16	NaN	5.641457	NaN

source

calculate_emission

```
calculate_emission (data:pandas.core.frame.DataFrame,
                    pco2_parameters:dict, bezetting:dict,
                    interpolate:dict={'interval': '7min', 'method':
                    'linear'})
```

Calculate the emission using the ratio method

	Type	Default	Details
data	DataFrame		DataFrame with measurement data
pco2_parameters	dict		parameters for the PCO2 calculation
bezetting	dict		dictionary with the animal categories and their counts
interpolate	dict	{‘interval’: ‘7min’, ‘method’: ‘linear’}	resampling interval and method

```
calculate_emission(vera_dataframe, pco2_parameters=default_pco2_parameters, bezetting={},
interpolate=dict() ).info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 60 entries, 2011-04-04 to 2012-02-16
Data columns (total 60 columns):
 #   Column                                              Non-Null Count  Dtype
---  ------                                              --------------  -----
 0   ratio                                               33 non-null     float64
 1   temperatuur_correctie                               59 non-null     float64
 2   PCO2_totaal                                         60 non-null     float64
 3   PCO2_corrected                                      59 non-null     float64
 4   emission                                            32 non-null     float64
 5   Measurement institute                               60 non-null     object
 6   Animal Category                                     60 non-null     object
 7   Housing system                                      60 non-null     object
 8   Measurement location                                60 non-null     object
 9   Measurement period                                  60 non-null     int64
 10  Measurement day (in period)                         60 non-null     int64
 11  Day in year                                         60 non-null     int64
 12  Outside temperature [oC]                            24 non-null     float64
 13  Outside RH [%]                                      23 non-null     float64
 14  Inside temperature [oC]                             59 non-null     float64
 15  Inside RH [%]                                       44 non-null     float64
 16  Animal places                                       60 non-null     int64
 17  Milking cows                                        60 non-null     int64
 18  Dry cows                                            60 non-null     int64
 19  Heifers (pregnant)                                  60 non-null     int64
 20  Heifers (not pregnant)                              60 non-null     int64
 21  Floor type (0: slatted floor; 1: closed floor)      60 non-null     int64
 22  Walking area per animal (m2)                        60 non-null     float64
 23  Grazing (hours per day)                             60 non-null     int64
 24  Closed cubicles                                     60 non-null     int64
 25  Milk production [kg/animal/day]                     56 non-null     float64
 26  Milk [% protein]                                    60 non-null     float64
 27  Milk [% fat]                                        60 non-null     float64
 28  Urea content in milk [mg/100g]                      56 non-null     float64
 29  Weight milking cows [kg]                            60 non-null     int64
 30  Weight dry cows [kg]                                60 non-null     int64
```

```
 31  Weight heifers (pregnant) [kg]                          60 non-null     int64
 32  Weight heifers (not pregnant) [kg]                      60 non-null     int64
 33  Days in pregnancy (milking cows)                        60 non-null     int64
 34  Days in pregnancy (dry cows)                            60 non-null     int64
 35  Days in pregnancy (heifers)                             60 non-null     int64
 36  Energy value of feed (heifers; MJ/kg dry matter)        60 non-null     int64
 37  Weight gain heifers [kg/day]                            60 non-null     float64
 38  CO2 inside [ppm]                                        60 non-null     int64
 39  CO2 outside [ppm]                                       52 non-null     float64
 40  NH3 inside [mg/m3]                                      33 non-null     float64
 41  NH3 outside [mg/m3]                                     60 non-null     int64
 42  Number of animals                                       60 non-null     int64
 43  Dairy cows (milking + dry)                              60 non-null     int64
 44  % closed cubicles                                       60 non-null     float64
 45  Occupation rate (%)                                     60 non-null     float64
 46  Dairy cows (%)                                          60 non-null     float64
 47  Heifers vs. dairy cows (%)                              60 non-null     float64
 48  Dry cows vs. dairy cows (%)                             60 non-null     float64
 49  Heat production milking cows (hpu)                      56 non-null     float64
 50  Heat production dry cows (hpu)                          56 non-null     float64
 51  Heat production heifers (pregnant) (hpu)                60 non-null     float64
 52  Heat production heifers (not pregnant) (hpu)            60 non-null     float64
 53  Total heat production (hpu)                             60 non-null     float64
 54  Total heat production corrected for temperature (hpu)   59 non-null     float64
 55  Ventilation rate [m3/h]                                 51 non-null     float64
 56  Ventilation rate [m3/h per animal]                      51 non-null     float64
 57  NH3 Emission [kg/year per animal place]                 32 non-null     float64
 58  Summary                                                 32 non-null     float64
 59  Summary.1                                               32 non-null     float64
dtypes: float64(33), int64(23), object(4)
memory usage: 28.6+ KB
```

```
extract_emission_column_names(data)
```

```
{'binnen': {'nh3': ['NH3 inside [mg/m3]'],
  'co2': ['CO2 inside [ppm]'],
  'temp': ['Inside temperature [oC]'],
  'rh': ['Inside RH [%]'],
  'wind': []},
 'buiten': {'nh3': ['NH3 outside [mg/m3]'],
  'co2': ['CO2 outside [ppm]'],
  'temp': ['Outside temperature [oC]'],
  'rh': ['Outside RH [%]'],
  'wind': []}}
```

Airflow from CO2

```
def calculate_airflow_from_co2(
        PCO2,          # CO2 production in kg per uur
        CO2_stal,      # CO2 concentration in the barn in ppm
        CO2_buiten,    # CO2 concentration outside in ppm
    ):
    '''Calculate the airflow from CO2 concentrations and production'''

    return PCO2 * 1e-6 *  CO2_buiten / CO2_stal  # m3 per uur
```

Analyse worksheet berekeningen

NH3 Emissie berekend [kg/dpl/jaar]

```
IF(
    ISNUMBER(BE4),
    BE4*(AO4-AP4)/1000000*24*365/(Q4-Y4),
    ""
)
```

$$BG_i = BE_i \times \frac{(AO_i - AP_i)}{Q_i - Y_i} \times \frac{24 \times 365}{1000000}$$

- BE_i is Debiet berekend [m3/uur]
- AO_i is NH3 concentratie stal [mg/m3]
- AP_i is NH3 concentratie buiten [mg/m3]
- Y_i is Afgedekte ligboxen
- Q_i is Dierplaatsen

Debiet berekend [m3/uur]

```
IF(
    OR(
        BD4="",AM4="*",AM4=""
    ),
    IF(
        ISNUMBER(AQ4),
        AQ4,
        ""
    ),
    BD4*0.2/(0.000001*(AM4-AN4)))
```

$$BE_i = BD_i \times \frac{0.2}{0.000001 \times (AM_i - AN_i)}$$

- BD_i is Warmteproductie (totaal, gecorrigeerd door temperatuur)
- AM_i is CO2 stal [ppm]
- AN_i is CO2 buiten [ppm]
- AQ_i is Debiet gemeten [m3/uur]

From these two equations we can derive:

$$BG_i = BD_i \times \frac{0.2}{0.000001 \times (AM_i - AN_i)} \times \frac{(AO_i - AP_i)}{Q_i - Y_i} \times \frac{24 \times 365}{1000000} \Leftrightarrow$$

$$BG_i = BD_i \times \frac{0.2}{AM_i - AN_i} \times \frac{(AO_i - AP_i)}{Q_i - Y_i} \times 24 \times 365 \Leftrightarrow$$

$$BG_i = 0.2 \times BD_i \times \frac{AO_i - AP_i}{AM_i - AN_i} \times \frac{24 \times 365}{Q_i - Y_i}$$

$$E_i = PCO_{2i} \cdot \frac{(NH_3)_i^{stal} - (NH_3)_i^{buiten}}{(CO_2)_i^{stal} - CO_{2i}^{buiten}}$$

Warmteproductie (totaal, gecorrigeerd door temperatuur)

```
IF(
    BC4="",
    "",
    IF(
        M4="*",
        BC4,
        BC4*(1000+4*(20-M4))/1000
    )
)
```

$$BD_i = BC_i \times \frac{1000 + 4 \times (20 - M_i)}{1000}$$

- BC_i is Warmteproductie (totaal)
- M_i is Temperatuur [°C]

Warmteproductie totaal [W]

```
=IF(SUM(AY4:BB4)=0,"",SUM(AY4:BB4))
```

$$BC_i = \sum_{j=AY}^{BB} P_j$$

* BC_i is Warmteproductie (totaal)

- P_j is Warmteproductie categorie (melkvee, droogstaande koeien, drachtig jongvee, niet drachtig jongvee)

Warmteproductie

Warmteproductie categorie melkvee

```
=IF(
    OR(R4="",Z4=""),
    "",
    (5.6*(IF(AD4="",'Input voor PCO2'!$C$5,AD4))^0.75+22*Z4+1.6*0.00001*(IF(AH4="",'Input voor PCO2'!$D$5,AH4))^3)*R4/1000
)
```

$$P_{melkvee} = \frac{5.6(AD_i)^{0.75} + 22Z_i + 1.6 \times 10^{-5}(AH_i)^3}{1000} \times R_i$$

Where:

- AD_i is Gewicht melkvee [kg]
- AH_i is Drachtdagen melkvee [dagen]
- Z_i is Melkproductie melkvee [kg/dag]

- R_i is Aantal melkvee

Calculatie vergelijking met voorbeeld data

Standaard parameters

```
test_parameters ={
    'melkvee': {
        'drachtdagen': 160,
        'gewicht': 650,
        'melkproductie': 28
    },
    'droogstaande koeien': {
        'drachtdagen': 220,
        'gewicht': 650,
        'melkproductie': 28
    },
    'drachtig jongvee': {
        'drachtdagen': 140,
        'gewicht': 400,
        'energievoeding': 10.0,
        'gewichtstoename': 0.6
    },
    'niet drachtig jongvee': {
        'drachtdagen': 0,
        'gewicht': 250,
        'energievoeding': 10.0,
        'gewichtstoename': 0.6
    }
}
```

```
bezetting = {
    'melkvee': dict(aantal=130),
    'droogstaande koeien': dict(aantal=6),
    'drachtig jongvee': dict(aantal=0),
    'niet drachtig jongvee': dict(aantal=0)
}
```

Parameters importeren

```
test_data_filename = os.path.join(os.getcwd(), '..', 'data', 'massabalans',
'Testdata2.xlsx')
print(test_data_filename)
```

```
/home/fenke/repos/openstal/nbs/../data/massabalans/Testdata2.xlsx
```

```
test_productiegegevens = pd.read_excel(test_data_filename,
sheet_name='Bedrijfsproductiegegevens', header=0, index_col=0, parse_dates=True)
```

```
test_productiegegevens
```

Parameter	Waarde	Naam parameter in WLR rapport	Naam parameter in Slimme Stal	Hoe vaak deze waarde verandert
Aantal dierplaatsen (=aantal ligboxen)	179.000000	-	NaN	zelden
Aantal melkgevende koeien	110.000000	-	NaN	elke 3-7 dagen
Aantal droogstaande koeien	13.000000	-	NaN	elke 3-7 dagen
Aantal drachtige pinken	14.000000	-	NaN	elke 3-7 dagen
Aantal niet-drachtig jongvee	15.000000	-	NaN	elke 3-7 dagen
Melkproductie (kg/koe/dag)	30.000000	Y1	NaN	elke 3 dagen
Ureumgetal (mg/100g)	16.000000	-	NaN	elke 3 dagen
Mest mest besmeurd oppervlakte (m2)	760.000000	-	NaN	zelden
Aantal ligboxen gesloten	21.000000	NaN	NaN	zelden
Gewicht melkkoe (kg)	650.000000	m	NaN	nooit
Gewicht droogstaande koe (kg)	650.000000	m	NaN	nooit

Parameter	Waarde	Naam parameter in WLR rapport	Naam parameter in Slimme Stal	Hoe vaak deze waarde verandert
Gewicht drachtige pink (kg)	400.000000	m	NaN	nooit
Gewicht niet-drachtig jongvee (kg)	250.000000	m	NaN	nooit
Dagen in dracht melkkoe	160.000000	p	NaN	nooit
Dagen in dracht droogstaande koe	220.000000	p	NaN	nooit
Dagen in dracht drachtige pink	140.000000	p	NaN	nooit
Energiewaarde voer drachtige pink (MJ/kg DS)	10.000000	M	NaN	nooit
Energiewaarde voer niet-drachtig jongevee (MJ/kg DS)	10.000000	M	NaN	nooit
Gewichtstoename drachtige pink (kg/dag)	0.600000	Y2	NaN	nooit
Gewichtstoename niet-drachtig	0.600000	Y2	NaN	nooit

Parameter	Waarde	Naam parameter in WLR rapport	Naam parameter in Slimme Stal	Hoe vaak deze waarde verandert
jongvee (kg/dag)				
NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN
PCO2 berekening	NaN	NaN	NaN	NaN
PCO2 melkvee	159.107598	NaN	NaN	NaN
PCO2 droogstaande koe	11.586441	NaN	NaN	NaN
PCO2 drachtige pink	9.459443	NaN	NaN	NaN
PCO2 niet-drachtig jongee	6.904258	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN
PCO2 totaal	187.057740	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN
Totaal aantal vee	152.000000	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN
Aantal ligboxen open	158.000000	NaN	NaN	NaN

Data importeren

```
test_dataframe = pd.read_excel(test_data_filename, sheet_name='Ruwe Data (CARS)', header=1,
index_col=0, parse_dates=True)
```

```
test_dataframe
```

Tijd	NH3 concentratie stal (ppm)	NH3 concentratie buiten1 (ppm)	CO2 concentratie stal (ppm)	CO2 concentratie buiten1 (ppm)	CO2 concentratie buiten2 (ppm)	Temperatuur stal1 ©	Temperatuur stal2 ©	Temperatuur buiten1 ©	Temperatuur buiten2 ©	Vochtigheid stal1 (%)	...	CO2 correctie [illegible]	Ventilatiedebiet (m3/u)	Ventilatiedebiet (m3/dier/u)	NH3 emissie (kg/u)	NH3 emissie (kg/j)	NH3 emissie (kg/dp/j)	Ventilatiedebiet (m3/u).1	NH3 emissie (kg/u).1	NH3 emissie (kg/j).1	NH3 emissie (kg/dp/j).1
2025-07-04 00:00:00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	17.9	20.2	13.5	13.7	72.0	...	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2025-07-04 00:01:00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	17.8	20.2	13.5	13.7	71.0	...	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2025-07-04 00:02:00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	17.8	20.2	13.5	13.7	71.0	...	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2025-07-04 00:03:00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	17.8	20.2	13.5	13.7	70.0	...	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2025-07-04 00:04:00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	17.8	20.1	13.5	13.7	71.0	...	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
2025-08-22 23:55:00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	16.6	21.3	15.0	14.9	74.5	...	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2025-08-22 23:56:00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	16.6	21.3	15.0	14.9	74.5	...	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2025-08-22 23:57:00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	16.6	21.3	15.0	14.9	74.5	...	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2025-08-22 23:58:00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	16.5	21.3	15.0	14.9	75.0	...	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2025-08-22 23:59:00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	16.6	21.3	15.0	14.9	75.0	...	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

```
test_dataframe.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 60 entries, NaT to NaT
Data columns (total 66 columns):
 #   Column
Non-Null Count  Dtype
---  ------
--------------  -----
 0   Measurement institute
```

```
60 non-null     object
 1   Animal Category
60 non-null     object
 2   Housing system
60 non-null     object
 3   Measurement location
60 non-null     object
 4   Measurement period
60 non-null     int64
 5   Measurement day (in period)
60 non-null     int64
 6   Date
60 non-null     datetime64[ns]
 7   Day in year
60 non-null     int64
 8   Outside temperature [oC]
24 non-null     float64
 9   Outside RH [%]
23 non-null     float64
 10  Inside temperature [oC]
59 non-null     float64
 11  Inside RH [%]
44 non-null     float64
 12  Winddirection
0 non-null      float64
 13  Windspeed (10 m height) [m/s]
0 non-null      float64
 14  Animal places
60 non-null     int64
 15  Milking cows
60 non-null     int64
 16  Dry cows
60 non-null     int64
 17  Heifers (pregnant)
60 non-null     int64
 18  Heifers (not pregnant)
60 non-null     int64
 19  Floor type (0: slatted floor; 1: closed floor)
60 non-null     int64
 20  Walking area per animal (m2)
60 non-null     float64
 21  Grazing (hours per day)
60 non-null     int64
 22  Closed cubicles
60 non-null     int64
 23  Milk production [kg/animal/day]
56 non-null     float64
 24  Milk [% protein]
60 non-null     float64
 25  Milk [% fat]
60 non-null     float64
 26  Urea content in milk [mg/100g]
56 non-null     float64
 27  Weight milking cows [kg]
60 non-null     int64
 28  Weight dry cows [kg]
60 non-null     int64
 29  Weight heifers (pregnant) [kg]
60 non-null     int64
 30  Weight heifers (not pregnant) [kg]
60 non-null     int64
 31  Days in pregnancy (milking cows)
60 non-null     int64
```

```
 32  Days in pregnancy (dry cows)
60 non-null     int64
 33  Days in pregnancy (heifers)
60 non-null     int64
 34  Energy value of feed (heifers; MJ/kg dry matter)
60 non-null     int64
 35  Weight gain heifers [kg/day]
60 non-null     float64
 36  CO2 inside [ppm]
60 non-null     int64
 37  CO2 outside [ppm]
52 non-null     float64
 38  NH3 inside [mg/m3]
33 non-null     float64
 39  NH3 outside [mg/m3]
60 non-null     int64
 40  Number of animals
60 non-null     int64
 41  Dairy cows (milking + dry)
60 non-null     int64
 42  % closed cubicles
60 non-null     float64
 43  Occupation rate (%)
60 non-null     float64
 44  Dairy cows (%)
60 non-null     float64
 45  Heifers vs. dairy cows (%)
60 non-null     float64
 46  Dry cows vs. dairy cows (%)
60 non-null     float64
 47  Heat production milking cows (hpu)
56 non-null     float64
 48  Heat production dry cows (hpu)
56 non-null     float64
 49  Heat production heifers (pregnant) (hpu)
60 non-null     float64
 50  Heat production heifers (not pregnant) (hpu)
60 non-null     float64
 51  Total heat production (hpu)
60 non-null     float64
 52  Total heat production corrected for temperature (hpu)
59 non-null     float64
 53  Ventilation rate [m3/h]
51 non-null     float64
 54  Ventilation rate [m3/h per animal]
51 non-null     float64
 55  NH3 Emission [kg/year per animal place]
32 non-null     float64
 56   C1:                   cows >= 70%
32 non-null     float64
 57  C2:                           Occupation rate >= 90%
32 non-null     float64
 58  C3:                                                                   milk production > 25
32 non-null     float64
 59  Summary
32 non-null     float64
 60   C1:                   heifers < 30%
32 non-null     float64
 61  C2:                           Occupation rate >= 80%
32 non-null     float64
 62  C3:                                                                   milk production > 25.1
32 non-null     float64
 63  C4:                                         urea content in milk > 15
```

```
32 non-null     float64
 64  C5:                            dry cows < 25%
32 non-null     float64
 65  Summary.1
32 non-null     float64
dtypes: datetime64[ns](1), float64(38), int64(23), object(4)
memory usage: 31.4+ KB
```

```python
fmap = create_pco2_function_mapping_from_parameters(test_parameters)
```

```python
columnmapping
```

```
{'stal': {'nh3': ['NH3 concentratie stal (ppm)'],
  'co2': ['CO2 concentratie stal (ppm)'],
  'temp': ['Temperatuur stal ©']},
 'buiten': {'nh3': ['NH3 concentratie buiten (ppm)'],
  'co2': ['CO2 concentratie buiten1 (ppm)',
   'CO2 concentratie buiten2 (ppm)',
   'CO2 concentratie buiten3 (ppm)'],
  'temp': []}}
```

```python
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 5728 entries, 2025-08-19 00:00:00 to 2025-08-22 23:59:00
Data columns (total 7 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   NH3 concentratie stal (ppm)     5728 non-null   float64
 1   NH3 concentratie buiten (ppm)   5728 non-null   float64
 2   CO2 concentratie stal (ppm)     5728 non-null   float64
 3   CO2 concentratie buiten1 (ppm)  5728 non-null   float64
 4   CO2 concentratie buiten2 (ppm)  5728 non-null   float64
 5   CO2 concentratie buiten3 (ppm)  5728 non-null   float64
 6   Temperatuur stal ©              5728 non-null   float64
dtypes: float64(7)
memory usage: 358.0 KB
```

```python
bezetting
```

```
{'melkvee': {'aantal': 130},
 'droogstaande koeien': {'aantal': 6},
 'drachtig jongvee': {'aantal': 0},
 'niet drachtig jongvee': {'aantal': 0}}
```

```python
fmap.get('melkvee')(**bezetting['melkvee'])
```

```
np.float64(36.46325050213528)
```

```python
PCO2_temperatuurcorrectie(
    fmap.get('niet drachtig jongvee')(**bezetting['niet drachtig jongvee']),
```

```
    temperatuur
)
```

```
Tijd
2025-08-19 00:00:00    0.0
2025-08-19 00:01:00    0.0
2025-08-19 00:02:00    0.0
2025-08-19 00:03:00    0.0
2025-08-19 00:04:00    0.0
                      ...
2025-08-22 23:55:00    0.0
2025-08-22 23:56:00    0.0
2025-08-22 23:57:00    0.0
2025-08-22 23:58:00    0.0
2025-08-22 23:59:00    0.0
Length: 5728, dtype: float64
```

```
PCO2_temperatuurcorrectie(
    fmap.get('drachtig jongvee')(**bezetting['drachtig jongvee']),
    temperatuur
)
```

```
Tijd
2025-08-19 00:00:00    0.0
2025-08-19 00:01:00    0.0
2025-08-19 00:02:00    0.0
2025-08-19 00:03:00    0.0
2025-08-19 00:04:00    0.0
                      ...
2025-08-22 23:55:00    0.0
2025-08-22 23:56:00    0.0
2025-08-22 23:57:00    0.0
2025-08-22 23:58:00    0.0
2025-08-22 23:59:00    0.0
Length: 5728, dtype: float64
```

```
PCO2_temperatuurcorrectie(
    fmap.get('droogstaande koeien')(**bezetting['droogstaande koeien']),
    temperatuur
)
```

```
Tijd
2025-08-19 00:00:00    1.063956
2025-08-19 00:01:00    1.063528
2025-08-19 00:02:00    1.063528
2025-08-19 00:03:00    1.063956
2025-08-19 00:04:00    1.063956
                         ...
2025-08-22 23:55:00    1.084063
2025-08-22 23:56:00    1.084063
2025-08-22 23:57:00    1.084063
2025-08-22 23:58:00    1.084491
2025-08-22 23:59:00    1.084063
Length: 5728, dtype: float64
```

```
PCO2_temperatuurcorrectie(
    fmap.get('melkvee')(**bezetting['melkvee']),
```

```
    temperatuur
).rename('melkvee')
```

```
Tijd
2025-08-19 00:00:00    36.273642
2025-08-19 00:01:00    36.259056
2025-08-19 00:02:00    36.259056
2025-08-19 00:03:00    36.273642
2025-08-19 00:04:00    36.273642
                         ...
2025-08-22 23:55:00    36.959151
2025-08-22 23:56:00    36.959151
2025-08-22 23:57:00    36.959151
2025-08-22 23:58:00    36.973736
2025-08-22 23:59:00    36.959151
Name: melkvee, Length: 5728, dtype: float64
```

```
pd.concat(
        [
            PCO2_temperatuurcorrectie(
                fmap.get(category)(**params)*5,
                temperatuur
            ).rename(category)
            for category, params in bezetting.items()
        ], axis=1
    ).sum(axis=1)
```

```
Tijd
2025-08-19 00:00:00    186.687989
2025-08-19 00:01:00    186.612923
2025-08-19 00:02:00    186.612923
2025-08-19 00:03:00    186.687989
2025-08-19 00:04:00    186.687989
                          ...
2025-08-22 23:55:00    190.216069
2025-08-22 23:56:00    190.216069
2025-08-22 23:57:00    190.216069
2025-08-22 23:58:00    190.291135
2025-08-22 23:59:00    190.216069
Length: 5728, dtype: float64
```

```
emissie = calculate_emission(
    data=data,
    pco2_parameters=test_parameters,
    bezetting=bezetting,
    interpolate=dict(interval='7min', method='linear')
).resample('1h').mean()
```

```
emissie / emissie.mean()
```

```
Tijd
2025-08-19 00:00:00    0.420778
2025-08-19 01:00:00    0.428174
2025-08-19 02:00:00    0.395387
2025-08-19 03:00:00    0.496850
2025-08-19 04:00:00    0.615691
                         ...
```

```
2025-08-22 19:00:00    0.489988
2025-08-22 20:00:00    0.245229
2025-08-22 21:00:00    0.267093
2025-08-22 22:00:00    0.573332
2025-08-22 23:00:00    0.595650
Freq: h, Length: 96, dtype: float64
```

```
test_dataframe.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 5760 entries, 2025-08-19 00:00:00 to 2025-08-22 23:59:00
Data columns (total 34 columns):
 #   Column                          Non-Null Count  Dtype
---  ------                          --------------  -----
 0   NH3 concentratie stal (ppm)     5758 non-null   float64
 1   NH3 concentratie buiten (ppm)   5758 non-null   float64
 2   CO2 concentratie stal (ppm)     5758 non-null   float64
 3   CO2 concentratie buiten1 (ppm)  5752 non-null   float64
 4   CO2 concentratie buiten2 (ppm)  5750 non-null   float64
 5   CO2 concentratie buiten3 (ppm)  5748 non-null   float64
 6   Temperatuur stal ©              5752 non-null   float64
 7   Temperatuur buiten1 ©           5752 non-null   float64
 8   Temperatuur buiten2 ©           5750 non-null   float64
 9   Temperatuur buiten3 ©           5748 non-null   float64
 10  Luchtvochtigheid stal (%)       5752 non-null   float64
 11  Luchtvochtigheid buiten1 (%)    5752 non-null   float64
 12  Luchtvochtigheid buiten2 (%)    5750 non-null   float64
 13  Luchtvochtigheid buiten3 (%)    5748 non-null   float64
 14  Temperatuur meetbuis ©          5758 non-null   float64
 15  Luchtvochtigheid meetbuis (%)   5758 non-null   float64
 16  Windrichting (graden)           96 non-null     float64
 17  Windsnelheid (km/u)             96 non-null     float64
 18  NH3 concentratie stal (mg/m3)   5760 non-null   float64
 19  NH3 concentratie buiten (mg/m3) 5760 non-null   int64
 20  CO2 concentratie stal (mg/m3)   5760 non-null   float64
 21  CO2 concentatie buiten (mg/m3)  5760 non-null   float64
 22  CO2 correctie (ppm)             5760 non-null   int64
 23  CO2 correctie (mg/m3)           5760 non-null   float64
 24  PCO2 correctie                  5760 non-null   float64
 25  Ventilatiedebiet (m3/u)         5760 non-null   float64
 26  Ventilatiedebiet (m3/dier/u)    5758 non-null   float64
 27  NH3 emissie (kg/u)              5760 non-null   float64
 28  NH3 emissie (kg/j)              5760 non-null   float64
 29  NH3 emissie (kg/dp/j)           5760 non-null   float64
 30  Ventilatiedebiet (m3/u).1       5752 non-null   float64
 31  NH3 emissie (kg/u).1            5752 non-null   float64
 32  NH3 emissie (kg/j).1            5752 non-null   float64
 33  NH3 emissie (kg/dp/j).1         5752 non-null   float64
dtypes: float64(32), int64(2)
memory usage: 1.5 MB
```

```
test_dataframe
```

Tijd	NH3 concentratie stal (ppm)	NH3 concentratie buiten (ppm)	CO2 concentratie stal (ppm)	CO2 concentratie buiten1 (ppm)	CO2 concentratie buiten2 (ppm)	CO2 concentratie buiten3 (ppm)	Temperatuur ©	Temperatuur buiten1 ©	Temperatuur buiten2 ©	Temperatuur buiten3 ©	…	P.	CO2 correctiefactor	Ventilatiedebiet (m3/u)	Ventilatiedebiet (m3/dier/u)	NH3 emissie (kg/u)	NH3 emissie (kg/j)	NH3 emissie (kg/dp/j)	Ventilatiedebiet (m3/u).1	NH3 emissie (kg/u).1	NH3 emissie (kg/j).1	NH3 emissie (kg/dp/j).1
2025-08-01 00:00:00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	21.3	19.4	19.2	20.3	…	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2025-08-01 00:01:00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	21.4	19.4	19.3	20.3	…	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2025-08-01 00:02:00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	21.4	19.4	19.2	20.3	…	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2025-08-01 00:03:00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	21.3	19.4	19.2	20.3	…	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2025-08-01 00:04:00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	21.3	19.3	19.2	20.3	…	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
[illegible] 23:55:00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	16.6	15.0	14.9	16.4	…	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible] 23:56:00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	16.6	15.0	14.9	16.4	…	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible] 23:57:00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	16.6	15.0	14.9	16.5	…	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible] 23:58:00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	16.5	15.0	14.9	16.5	…	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible] 23:59:00	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	16.6	15.0	14.9	16.5	…	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

```python
print(json.dumps(extract_column_names(test_dataframe), indent=2))
```

```
{
  "stal": {
    "nh3": [
      "NH3 concentratie stal (ppm)",
      "NH3 concentratie stal (mg/m3)"
    ],
    "co2": [
      "CO2 concentratie stal (ppm)",
```

```
      "CO2 concentratie stal (mg/m3)"
    ],
    "temp": [
      "Temperatuur stal \u00a9"
    ]
  },
  "buiten": {
    "nh3": [
      "NH3 concentratie buiten (ppm)",
      "NH3 concentratie buiten (mg/m3)"
    ],
    "co2": [
      "CO2 concentratie buiten1 (ppm)",
      "CO2 concentratie buiten2 (ppm)",
      "CO2 concentratie buiten3 (ppm)",
      "CO2 concentatie buiten (mg/m3)"
    ],
    "temp": [
      "Temperatuur buiten1 \u00a9",
      "Temperatuur buiten2 \u00a9",
      "Temperatuur buiten3 \u00a9"
    ]
  }
}
```

```python
import nbdev; nbdev.nbdev_export()
```

Bibliography