carbon_model.rb

Lodging carbon model

This model is used by Brighter Planet's carbon emission web service to estimate the **greenhouse gas emissions of a lodging** (e.g. a hotel stay).

Calculations

The final estimate is the result of the **calculations** detailed below. These calculations are performed in reverse order, starting with the last calculation listed and finishing with the `emission` calculation. Each calculation is named according to the value it returns.

Methods

To accomodate varying client input, each calculation may have one or more **methods**. These are listed under each calculation in order from most to least preferred. Each method is named according to the values it requires. If any of these values is not available the method will be ignored. If all the methods for a calculation are ignored, the calculation will not return a value. "Default" methods do not require any values, and so a calculation with a default method will always return a value.

Standard compliance

Each method lists any established calculation standards with which it **complies**. When compliance with a standard is requested, all methods that do not comply with that standard are ignored. This means that any values a particular method requires will have been calculated using a compliant method, because those are the only methods available. If any value did not have a compliant method in its calculation then it would be undefined, and

```
module BrighterPlanet
  module Lodging
    module CarbonModel
      def self.included(base)
        base.decide :emission, :with => :characteristics do
```

the current method would have been ignored.

Collaboration

Contributions to this carbon model are actively encouraged and warmly welcomed. This library includes a comprehensive test suite to ensure that your changes do not cause regressions. All changes should include test coverage for new functionality. Please see sniff, our emitter testing framework, for more information.

Emission calculation

Returns the `emission` estimate ($kg\ CO_2e$).

Emission from rooms, duration, and emission factor

Complies: GHG Protocol Scope 3, ISO 14064-1, Climate Registry Protocol

Multiplies `rooms` by `duration` (*seconds*) (converted to nights) and the `emission factor` ($kg\ CO_2e\ /\ room\text{-}night$) to give (*kg CO_2e).

```
          committee :emission do


            quorum 'from rooms, duration, and emission factor', :needs =>
[:rooms, :duration, :emission_factor],

              :complies => [:ghg_protocol_scope_3, :iso, :tcr] do
|characteristics|

                characteristics[:rooms] * characteristics[:duration] /
86400.0 * characteristics[:emission_factor]
            end
          end
```

Emission factor calculation

Returns the `emission factor` ($kg\ CO_2e\ /\ room\text{-}night$)

Emission factor from fuel intensities and eGRID

Complies: GHG Protocol Scope 3, ISO 14064-1, Climate Registry Protocol

```
          committee :emission_factor do


            quorum 'from fuel intensities and eGRID', :needs =>
[:natural_gas_intensity, :fuel_oil_intensity, :electricity_intensity,
:district_heat_intensity, :egrid_subregion, :egrid_region],

              :complies => [:ghg_protocol_scope_3, :iso, :tcr] do
|characteristics|
```

Calculates an energy-based emission factor for natural gas by dividing its `co2 emission factor` ($kg / cubic\ m$) by its `energy content` ($MJ / cubic\ m$) to give $kg\ CO_2 / MJ$

Calculates an energy-based emission factor for fuel oil by dividing its `co2 emission factor` (kg / l) by its `energy content` (MJ / l) to give $kg\ CO_2 / MJ$

Calculates an energy-based emission factor for district heat by dividing the energy-based natural gas emission factor by 0.817 and the energy-based fuel oil emission factor by 0.846 (to account for boiler inefficiencies), averaging the two, and dividing by 0.95 (to account for transmission losses) to give $kg\ CO_2 / MJ$

Calculates an electricity emission factor by dividing the eGRID subregion electricity emission factor by 1 – the eGRID region loss factor (to account for transmission and distribution losses) to give $kg\ CO_2 / kWh$

Multiplies `natural gas intensity` ($cubic\ m / room\text{-}night$) by the volume-based natural gas emission factor ($kg\ CO_2 / room\text{-}night$), `fuel oil intensity` ($l / room\text{-}night$) by the volume-based fuel oil emission factor ($kg\ CO_2 / l$), `electricity intensity` ($kWh / room\text{-}night$) by the electricity emission factor ($kg\ CO_2 / kWh$), and `district heat intensity` ($MJ / room\text{-}night$) by the energy-based district heat emission factor ($kg\ CO_2 / MJ$), and adds these together to give $kg\ CO_2 / room\text{-}night$.

Natural gas intensity calculation

Returns the `natural gas intensity` ($cubic\ m / room\text{-}night$).

Natural gas intensity from census division

```
                natural_gas = Fuel.find_by_name "Pipeline Natural Gas"
                natural_gas_energy_ef = natural_gas.co2_emission_factor /
natural_gas.energy_content



                fuel_oil = Fuel.find_by_name "Distillate Fuel Oil No. 2"
                fuel_oil_energy_ef = fuel_oil.co2_emission_factor /
fuel_oil.energy_content



                district_heat_ef = (((natural_gas_energy_ef / 0.817) +
(fuel_oil_energy_ef / 0.846)) / 2) / 0.95 # kg / MJ









                electricity_ef =
characteristics[:egrid_subregion].electricity_emission_factor / (1 -
characteristics[:egrid_region].loss_factor)



                (characteristics[:natural_gas_intensity] *
natural_gas.co2_emission_factor) +
                (characteristics[:fuel_oil_intensity] *
fuel_oil.co2_emission_factor) +
                (characteristics[:district_heat_intensity] *
district_heat_ef) +
                (characteristics[:electricity_intensity] * electricity_ef)
            end
          end



          committee :natural_gas_intensity do



            quorum 'from census division', :needs => :census_division,
```

Complies: GHG Protocol Scope 3, ISO 14064-1, Climate Registry Protocol

Looks up the census division `natural gas intensity` (*cubic m / room-night*).

Natural gas intensity from lodging class

Complies: GHG Protocol Scope 3, ISO 14064-1, Climate Registry Protocol

Looks up the lodging class `natural gas intensity` (*cubic m / room-night*).

Fuel oil intensity calculation

Returns the `fuel oil intensity` (*l / room-night*).

Fuel oil intensity from census division

Complies: GHG Protocol Scope 3, ISO 14064-1, Climate Registry Protocol

Looks up the census division `fuel oil intensity` (*l / room-night*).

Fuel oil intensity from lodging class

Complies: GHG Protocol Scope 3, ISO 14064-1, Climate Registry Protocol

```
              :complies => [:ghg_protocol_scope_3, :iso, :tcr] do
|characteristics|


characteristics[:census_division].lodging_building_natural_gas_intensity
            end


            quorum 'from lodging class', :needs => :lodging_class,

              :complies => [:ghg_protocol_scope_3, :iso, :tcr] do
|characteristics|

                characteristics[:lodging_class].natural_gas_intensity
            end
          end


          committee :fuel_oil_intensity do



            quorum 'from census division', :needs => :census_division,

              :complies => [:ghg_protocol_scope_3, :iso, :tcr] do
|characteristics|


characteristics[:census_division].lodging_building_fuel_oil_intensity
            end


            quorum 'from lodging class', :needs => :lodging_class,

              :complies => [:ghg_protocol_scope_3, :iso, :tcr] do
|characteristics|
```

Looks up the lodging class `fuel oil intensity` (*l / room-night*).

Electricity intensity calculation

Returns the `electricity intensity` (*kWh / room-night*).

Electricity intensity from census division

Complies: GHG Protocol Scope 3, ISO 14064-1, Climate Registry Protocol

Looks up the census division `electricity intensity` (*kWh / room-night*).

Electricity intensity from lodging class

Complies: GHG Protocol Scope 3, ISO 14064-1, Climate Registry Protocol

Looks up the lodging class `electricity intensity` (*kWh / room-night*).

District heat intensity calculation

Returns the `district heat intensity` (*MJ / room-night*).

District heat intensity from census division

Complies: GHG Protocol Scope 3, ISO 14064-1, Climate Registry Protocol

```
            characteristics[:lodging_class].fuel_oil_intensity
          end
        end


        committee :electricity_intensity do


          quorum 'from census division', :needs => :census_division,

            :complies => [:ghg_protocol_scope_3, :iso, :tcr] do
|characteristics|

characteristics[:census_division].lodging_building_electricity_intensity
          end


          quorum 'from lodging class', :needs => :lodging_class,

            :complies => [:ghg_protocol_scope_3, :iso, :tcr] do
|characteristics|

            characteristics[:lodging_class].electricity_intensity
          end
        end


        committee :district_heat_intensity do


          quorum 'from census division', :needs => :census_division,

            :complies => [:ghg_protocol_scope_3, :iso, :tcr] do
```

Looks up the census division `district heat intensity` (*MJ / room-night*).

District heat intensity from lodging class

Complies: GHG Protocol Scope 3, ISO 14064-1, Climate Registry Protocol

Looks up the lodging class `district heat intensity` (*MJ / room-night*).

Lodging class calculation

Returns the lodging class.

Lodging class from client input

Complies: All

Uses the client-input loding class.

Default lodging class

Complies: GHG Protocol Scope 3, ISO 14064-1, Climate Registry Protocol

Uses an artificial lodging class that represents the U.S. average.

eGRID region calculation

```
|characteristics|


characteristics[:census_division].lodging_building_district_heat_intensity
            end


            quorum 'from lodging class', :needs => :lodging_class,

              :complies => [:ghg_protocol_scope_3, :iso, :tcr] do
|characteristics|

                characteristics[:lodging_class].district_heat_intensity
            end
          end


          committee :lodging_class do









            quorum 'default',

              :complies => [:ghg_protocol_scope_3, :iso, :tcr] do

                LodgingClass.find_by_name 'Average'
            end
          end


          committee :egrid_region do
```

Returns the lodging's eGRID region.

eGRID region from eGRID subregion

Complies: GHG Protocol Scope 3, ISO 14064-1, Climate Registry Protocol

Looks up the eGRID subregion `eGRID region`.

```
          quorum 'from eGRID subregion', :needs => :egrid_subregion,
            :complies => [:ghg_protocol_scope_3, :iso, :tcr] do
|characteristics|
              characteristics[:egrid_subregion].egrid_region
            end
          end
```

eGRID subregion calculation

Returns the lodging's eGRID subregion.

eGRID subregion from zip code

Complies: GHG Protocol Scope 3, ISO 14064-1, Climate Registry Protocol

Looks up the zip code `eGRID subregion`.

```
          committee :egrid_subregion do

            quorum 'from zip code', :needs => :zip_code,
              :complies => [:ghg_protocol_scope_3, :iso, :tcr] do
|characteristics|
              characteristics[:zip_code].egrid_subregion
            end

            quorum 'default', :complies => [:ghg_protocol_scope_3, :iso,
:tcr] do
              EgridSubregion.find_by_abbreviation 'US'
            end
          end
```

Default eGRID subregion

Complies: GHG Protocol Scope 3, ISO 14064-1, Climate Registry Protocol

Uses an artificial eGRID subregion that represents the U.S. average.

Census division calculation

Returns the lodging's census division.

```
          committee :census_division do
```

Census division from state

Complies: GHG Protocol Scope 3, ISO 14064-1, Climate Registry Protocol

Looks up the state `census division`.

```
            quorum 'from state', :needs => :state,

              :complies => [:ghg_protocol_scope_3, :iso, :tcr] do
|characteristics|

                characteristics[:state].census_division
              end
            end
```

State calculation

Returns the lodging's state.

```
          committee :state do
```

State from zip code

Complies: GHG Protocol Scope 3, ISO 14064-1, Climate Registry Protocol

Looks up the zip code `state`.

```
            quorum 'from zip code', :needs => :zip_code,

              :complies => [:ghg_protocol_scope_3, :iso, :tcr] do
|characteristics|

                characteristics[:zip_code].state
              end
            end
```

Zip code calculation

Returns the client-input zip code.

Duration calculation

Returns the stay's `duration` (*seconds*).

```
          committee :duration do
```

Duration from client input

Complies: All

Uses the client-input `duration` (*seconds*).

Default duration

Uses 86400 *seconds* (1 night).

```
    quorum 'default' do

      86400.0
    end
  end
```

Rooms calculation

Returns the number of `rooms` used.

```
  committee :rooms do
```

Rooms from client input

Complies: All

Uses the client-input number of `rooms`.

Default rooms

Uses 1 room.

```
    quorum 'default' do

      1
    end
  end
 end
end
```