

interpolate_implementation

November 14, 2020

```
[1]: import pandas as pd
import pyam
import numpy as np
```

<IPython.core.display.Javascript object>

```
[2]: obs = pd.DataFrame([
    ['model_a', 'scen_a', 'World', 'all', 'EJ/yr', 0, 1, 6., 10],
    ['model_a', 'scen_a', 'World', 'last', 'EJ/yr', 0, 0.5, 3, np.nan],
    ['model_a', 'scen_b', 'World', 'first', 'EJ/yr', 0, np.nan, 2, 7],
    ['model_a', 'scen_b', 'World', 'middle', 'EJ/yr', 0, 1, np.nan, 7],
    ['model_a', 'scen_b', 'World', 'first two', 'EJ/yr', 0, np.nan, np.nan, 7],
    ['model_a', 'scen_b', 'World', 'last two', 'EJ/yr', 0, 1, np.nan, np.nan],
],
    columns=pyam.IAMC_IDX + [2000, 2005, 2010, 2017],
)
```

```
[3]: def interpolate(df, time, **kwargs):
    # set up
    interp_kwargs = dict(method='slinear', axis=1)
    interp_kwargs.update(kwargs)
    if not pyam.islistable(time):
        time = [time]
    columns = np.sort(df.year + time)
    meta = df.meta
    df = df.timeseries()

    # calculate a separate dataframe with full interpolation
    newdf = (
        df
        .reindex(columns=columns)
        .interpolate(**interp_kwargs)
    )

    # replace only columns asked for
    for col in time:
        df[col] = newdf[col]
```

```

# update data
return pyam.IamDataFrame(
    df,
    meta=meta
)

```

```
[4]: obs
```

```

[4]:      model scenario region  variable  unit  2000  2005  2010  2017
0  model_a  scen_a  World    all  EJ/yr    0    1.0   6.0  10.0
1  model_a  scen_a  World    last EJ/yr    0    0.5   3.0   NaN
2  model_a  scen_b  World    first EJ/yr    0   NaN   2.0   7.0
3  model_a  scen_b  World  middle EJ/yr    0    1.0   NaN   7.0
4  model_a  scen_b  World  first two EJ/yr    0   NaN   NaN   7.0
5  model_a  scen_b  World  last two  EJ/yr    0    1.0   NaN   NaN

```

```

[5]: times = 2007
interpolate(pyam.IamDataFrame(obs), times).timeseries()

```

```

[5]:      model  scenario region variable  unit  2000  2005  2007  2010  2017
model_a scen_a  World  all  EJ/yr  0.0  1.0  3.000000  6.0  10.0
      last  EJ/yr  0.0  0.5  1.500000  3.0  NaN
      scen_b  World  first  EJ/yr  0.0  NaN  1.400000  2.0  7.0
      first two EJ/yr  0.0  NaN  2.882353  NaN  7.0
      last two  EJ/yr  0.0  1.0  NaN  NaN  NaN
      middle  EJ/yr  0.0  1.0  2.000000  NaN  7.0

```

```

[6]: times = [2007, 2005]
interpolate(pyam.IamDataFrame(obs), times).timeseries()

```

```

[6]:      model  scenario region variable  unit  2000  2005  2007  2010  2017
model_a scen_a  World  all  EJ/yr  0.0  1.000000  3.000000  6.0  10.0
      last  EJ/yr  0.0  0.500000  1.500000  3.0  NaN
      scen_b  World  first  EJ/yr  0.0  1.000000  1.400000  2.0  7.0
      first two EJ/yr  0.0  2.058824  2.882353  NaN  7.0
      last two  EJ/yr  0.0  1.000000  NaN  NaN  NaN
      middle  EJ/yr  0.0  1.000000  2.000000  NaN  7.0

```

```

[7]: times = [2007, 2010]
interpolate(pyam.IamDataFrame(obs), times).timeseries()

```

```

[7]:      model  scenario region variable  unit  2000  2005  2007  2010  2017
model_a scen_a  World  all  EJ/yr  0.0  1.0  3.000000  6.000000  10.0

```

		last	EJ/yr	0.0	0.5	1.500000	3.000000	NaN
scen_b	World	first	EJ/yr	0.0	NaN	1.400000	2.000000	7.0
		first two	EJ/yr	0.0	NaN	2.882353	4.117647	7.0
		last two	EJ/yr	0.0	1.0	NaN	NaN	NaN
		middle	EJ/yr	0.0	1.0	2.000000	3.500000	7.0

```
[8]: times = [2007, 2012]
      interpolate(pyam.IamDataFrame(obs), times).timeseries()
```

```
[8]:
```

					2000	2005	2007	2010	2012	\
model	scenario	region	variable	unit						
model_a	scen_a	World	all	EJ/yr	0.0	1.0	3.000000	6.0	7.142857	
			last	EJ/yr	0.0	0.5	1.500000	3.0	NaN	
	scen_b	World	first	EJ/yr	0.0	NaN	1.400000	2.0	3.428571	
			first two	EJ/yr	0.0	NaN	2.882353	NaN	4.941176	
			last two	EJ/yr	0.0	1.0	NaN	NaN	NaN	
			middle	EJ/yr	0.0	1.0	2.000000	NaN	4.500000	
										2017
model	scenario	region	variable	unit						
model_a	scen_a	World	all	EJ/yr	10.0					
			last	EJ/yr	NaN					
	scen_b	World	first	EJ/yr	7.0					
			first two	EJ/yr	7.0					
			last two	EJ/yr	NaN					
			middle	EJ/yr	7.0					

```
[ ]:
```