MNRAS **000**, 000–000 (0000) Preprint 8 October 2020 Compiled using MNRAS LATEX style file v3.0

Source code for `astroML.time_series.generate`

8 October 2020

1 SOURCE CODE FOR ASTROML.TIMESERIES.GENERATE

Generate a damped random walk light curve.
This uses a damped random walk model to generate a light curve similar to that of a QSO Kelly et al. (2009).
From:: https://www.astroml.org/_modules/astroML/time_series/generate.html

The differential equation is

$$dX = -X(t)dt + \sigma\sqrt{\tau}\epsilon(t)\sqrt{(dt)} + (b\tau dt) \tag{1}$$

where $\epsilon(t)$ is white noise with zero mean and unit variance, and

$$\overline{X(t)} = b\tau \tag{2}$$

and

$$SF_\infty = \sigma\sqrt{\tau/2} \tag{3}$$

$$\Rightarrow$$

$$\sqrt{2}SF_\infty = \sigma\sqrt{\tau} \tag{4}$$

so

$$dX(t) = -X(t)dt + \sqrt{2}SF_\infty\epsilon(t)\sqrt{(dt)} + \overline{X}dt \tag{5}$$

$$dX(t) = (\overline{X} - X(t))dt + \sqrt{2}SF_\infty\epsilon(t)\sqrt{(dt)} \tag{6}$$

```
def generate_damped_RW(t_rest, tau=300., z=2.0, xmean=0, SFinf=0.3, random_state=None):

t_rest = np.atleast_1d(t_rest)

if t_rest.ndim != 1:
    raise ValueError('t_rest should be a 1D array')

random_state = check_random_state(random_state)

t_obs = t_rest * (1. + z) / tau

N    = len(t_rest)
x    = np.zeros(N)
x[0] = random_state.normal(xmean, SFinf)
E    = random_state.normal(0, 1, N)

for i in range(1, N):
    dt = t_obs[i] - t_obs[i - 1]
    x[i] = (x[i - 1]
            - dt * (x[i - 1] - xmean)
            + np.sqrt(2) * SFinf * E[i] * np.sqrt(dt))
```

REFERENCES

Kelly B. C., Bechtold J., Siemiginowska A., 2009, ApJ, 698, 895