

---

# **RAIL Documentation**

**David Bailey**

**Mar 24, 2021**



---

## Contents

---

<b>1</b>	<b>Getting Started with RAIL</b>	<b>1</b>
<b>2</b>	<b>api</b>	<b>7</b>
<b>3</b>	<b>Indices and tables</b>	<b>11</b>
	<b>Python Module Index</b>	<b>13</b>
	<b>Index</b>	<b>15</b>



---

## Getting Started with RAIL

---

```
[50]: from rail import Controls, Impact, Likelihood, Risks, ThreatEvents, ThreatSources,   
      ↪Tree, Vulnerabilities
```

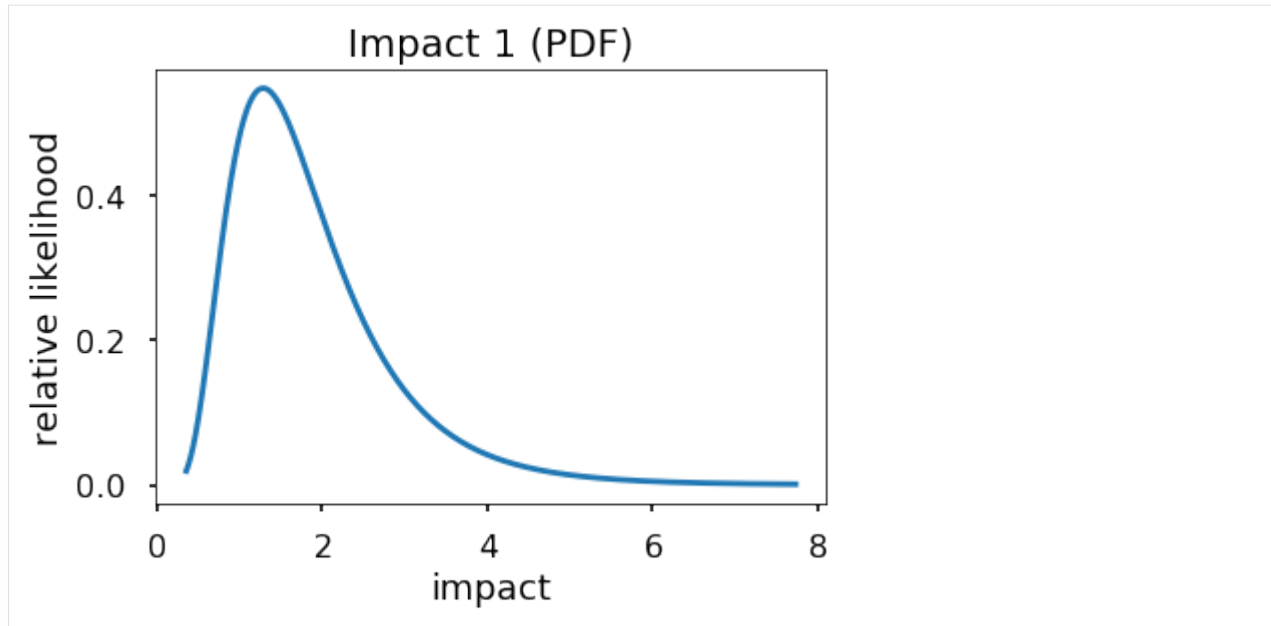
```
[51]: test_system = Tree(name="Test System")  
      test_system.add_child(name="Test Child")  
      test_system["Test Child"].add_child(name="Test Grandchild")  
      test_system.to_print()  
  
      /Test System  
      /Test System/Test Child  
      /Test System/Test Child/Test Grandchild
```

```
[52]: threat_sources = ThreatSources()  
      threat_sources.new(name="Threat Source 1")  
      threat_sources
```

```
[52]: {'Threat Source 1': {'name': 'Threat Source 1'}}
```

```
[53]: impact_one = Impact(name="Impact 1", mu=.5, sigma=.5)  
      impact_one.plot()
```

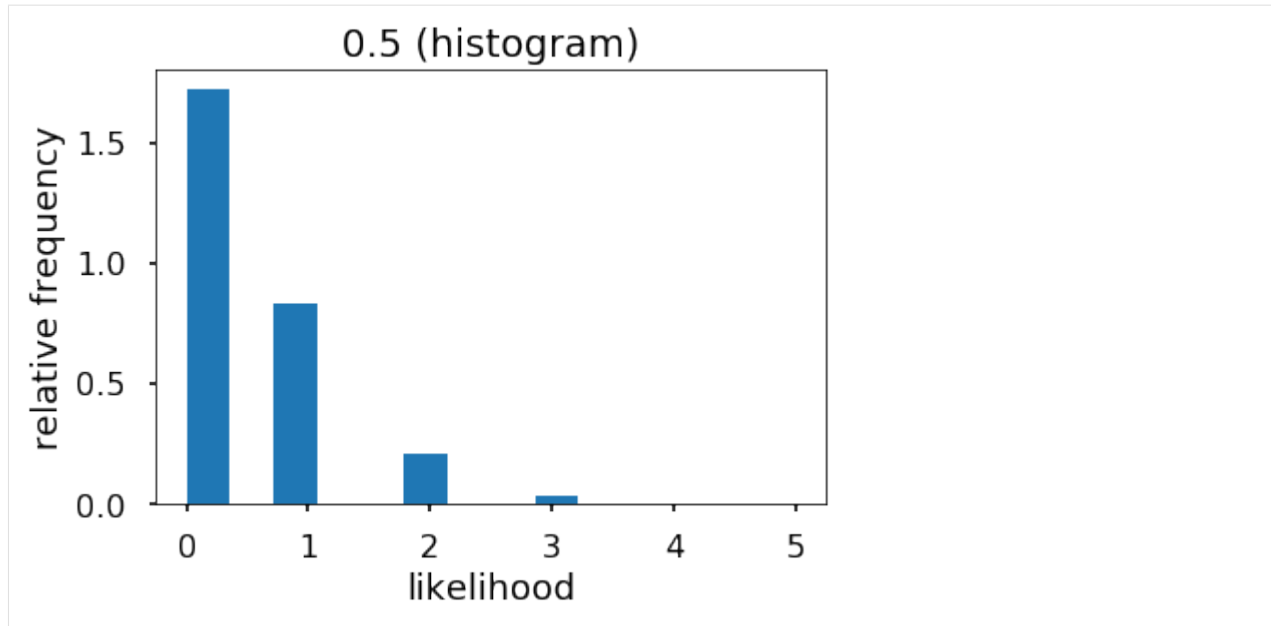
```
[53]: [<matplotlib.lines.Line2D at 0x12653ab50>]
```



```
[54]: test_likelihood = Likelihood(lam=.5)
test_likelihood.plot()

/usr/local/lib/python3.7/site-packages/matplotlib/axes/_axes.py:6499:
↳MatplotlibDeprecationWarning:
The 'normed' kwarg was deprecated in Matplotlib 2.1 and will be removed in 3.1. Use
↳'density' instead.
  alternative="'density'", removal="3.1")

[54]: (array([1.72032e+00, 0.00000e+00, 8.31600e-01, 0.00000e+00, 0.00000e+00,
 2.07760e-01, 0.00000e+00, 0.00000e+00, 3.64000e-02, 0.00000e+00,
 0.00000e+00, 3.36000e-03, 0.00000e+00, 5.60000e-04]),
array([0.          , 0.35714286, 0.71428571, 1.07142857, 1.42857143,
 1.78571429, 2.14285714, 2.5          , 2.85714286, 3.21428571,
 3.57142857, 3.92857143, 4.28571429, 4.64285714, 5.          ]),
<a list of 14 Patch objects>)
```



```
[55]: controls = Controls()
controls.new('Control 1', cost=100000, reduction=.01)
controls
```

```
[55]: {'Control 1': {'name': 'Control 1', 'cost': 100000, 'reduction': 0.01, 'implemented':
↪True}}
```

```
[56]: threat_events = ThreatEvents()
vulnerabilities = Vulnerabilities()
risks = Risks()

risks.new(
    vulnerabilities.new(
        threat_events.new(name="Threat Event 1", threat_source=threat_sources["Threat_
↪Source 1"]),
        test_system["Test Child"],
        [controls['Control 1']],
        test_likelihood,
        impact_one)

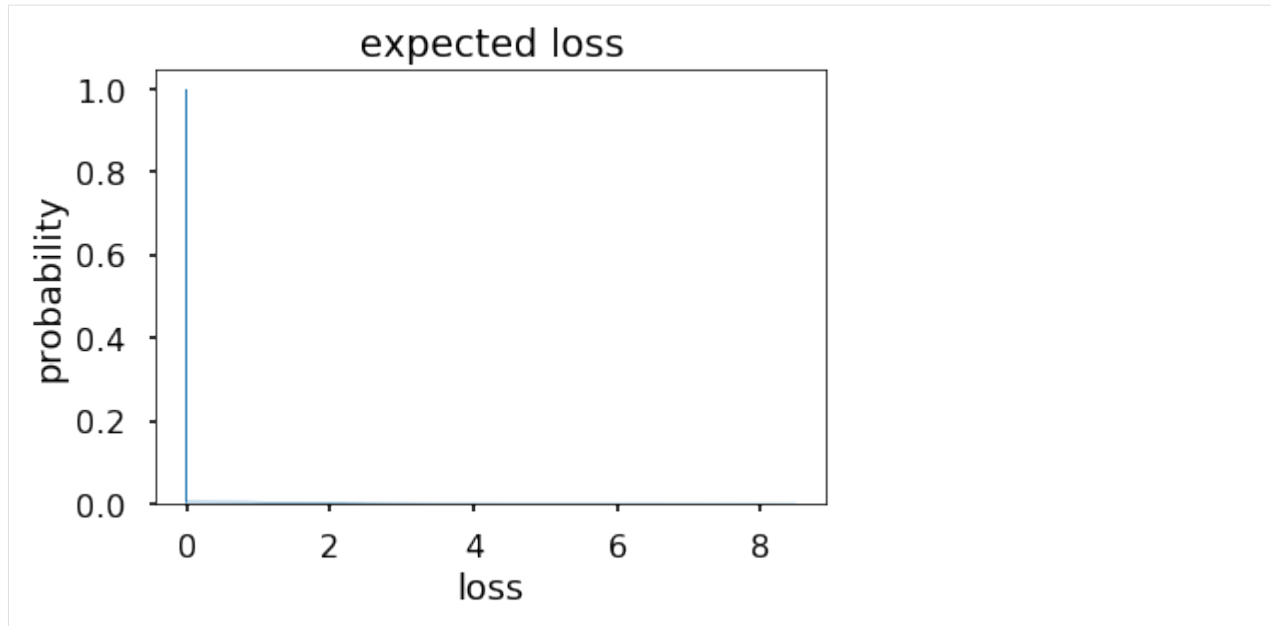
risks.dataframe
```

```
[56]:      Threat Source  Threat Event  System  Controls \
0 Threat Source 1  Threat Event 1  /Test System/Test Child  [Control 1]

      Impact  Impact (mean)  Likelihood (mean)
0 Impact 1 1.87          0.50
```

```
[57]: risks.plot()
```

```
[57]: (array([1.00e+00, 5.25e-03, 5.25e-03, ..., 1.00e-05, 1.00e-05, 1.00e-05]),
array([0.00000000e+00, 8.49821044e-04, 1.69964209e-03, ...,
8.49651079e+00, 8.49736062e+00, 8.49821044e+00]),
<a list of 1 Patch objects>)
```



```
[58]: risks.set_optimum_controls(controls)
```

```
[58]:
```

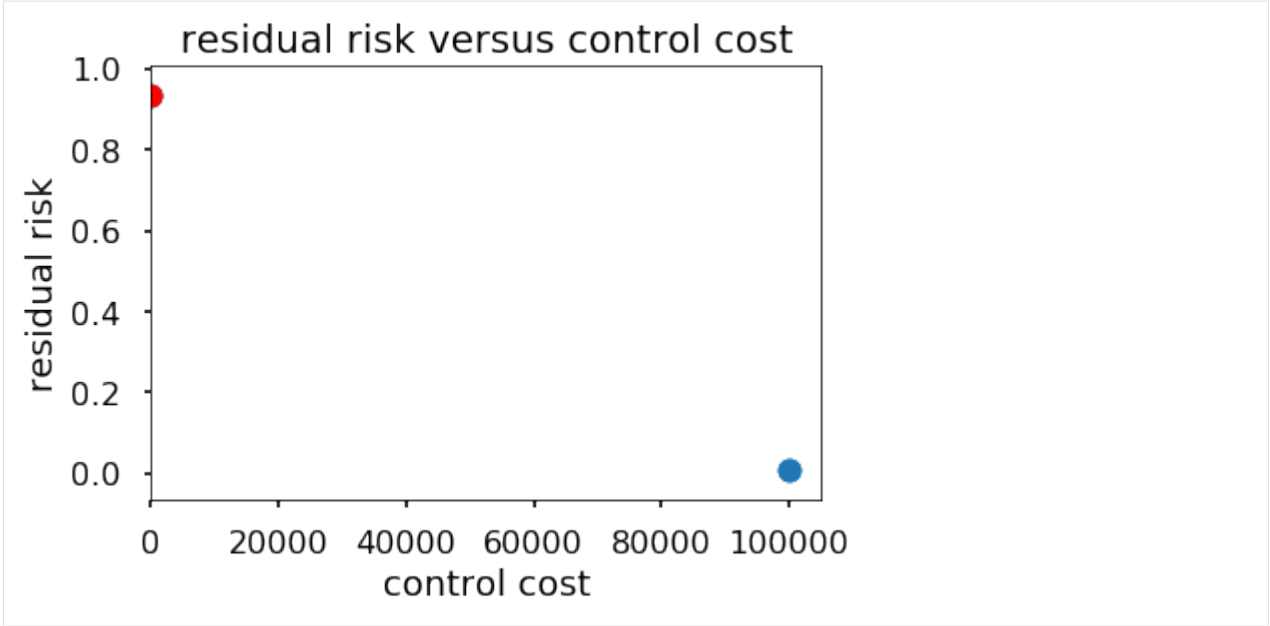
	cost	implemented	reduction
name			
Control 1	100000	False	0.01

```
[59]: from matplotlib import pyplot as plt
```

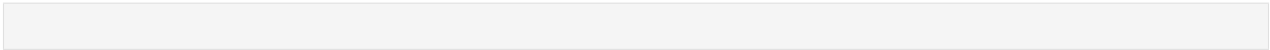
```
fig = plt.figure()
axes = fig.add_subplot(1,1,1)
risks.plot_risk_cost_matrix(controls, axes)
```

```
/usr/local/lib/python3.7/site-packages/matplotlib/axes/_base.py:3116:
↳MatplotlibDeprecationWarning:
The `xmin` argument was deprecated in Matplotlib 3.0 and will be removed in 3.2. Use
↳`left` instead.
alternative=`left`, obj_type='argument')
```





[ ]:





```
class rail.CPI
    Bases: object

    A class to retrieve United States CPI data and calculate inflation

    inflation (from_year: int, to_year: int) → float
        A method to retrieve United States CPI data and calculate inflation

class rail.Control (name: str, cost: float, reduction: float, implemented: bool = True)
    Bases: collections.UserDict

    A class to represent Controls

    evaluate_lognormal (iterations=1)

class rail.Controls
    Bases: collections.UserDict

    A class to hold multiple Controls

    costs ()
        A method to compute the deterministic costs of implemented controls in a Controls class

    costs_lognormal ()
        A method to compute the stochastic costs of implemented controls in a Controls class

    new (name: str, cost: float, reduction: float) → rail.control.Control
        A method to add a new controls to the Controls class

class rail.Impact (name: str, mu: float, sigma: float)
    Bases: collections.UserDict

    A class to represent an Impact

    from_lower_90_upper_90 (lower_90: float, upper_90: float)
        A method to create an impact from the lower 90th and upper 90th percentiles

    plot (num=1000, axes=None) → list
        A method to plot the impact
```

```
class rail.Likelihood (lam: float)
    Bases: collections.UserDict

    A class to represent a Likelihood

    plot (axes=None) → tuple
        A method to plot the likelihood

class rail.Risk (vulnerability: rail.vulnerability.Vulnerability, likelihood: rail.likelihood.Likelihood,
                impact: rail.impact.Impact)
    Bases: collections.UserDict

    evaluate_deterministic () → float

    evaluate_lognormal (iterations: int = 1000) → float

class rail.Risks
    Bases: collections.UserDict

    calculate_dataframe_deterministic_mean ()

    calculate_stochastic_risks (iterations: int = 100000)

    determine_optimum_controls (controls, controls_to_optimize, stochastic=False)

    expected_loss_deterministic_mean () → float

    expected_loss_stochastic_mean (iterations: int = 1000) → float

    new (vulnerability: rail.vulnerability.Vulnerability, likelihood: rail.likelihood.Likelihood, impact:
        rail.impact.Impact) → rail.risk.Risk

    plot (axes=None)

    plot_risk_cost_matrix (controls, axes=None)

    sensitivity_test (controls, iterations=1000)

    set_optimum_controls (controls)

class rail.ThreatEvent (name: str, threat_source: rail.threat_source.ThreatSource)
    Bases: collections.UserDict

    A class to represent Threat Events

class rail.ThreatEvents (**kwargs)
    Bases: collections.UserDict

    A class to hold multiple Threat Events

    new (name: str, threat_source: rail.threat_source.ThreatSource) → rail.threat_event.ThreatEvent
        A method to add a new threat event to the Threat Events class

class rail.ThreatSource (name: str)
    Bases: collections.UserDict

    A class to represent Threat Sources

class rail.ThreatSources (**kwargs)
    Bases: collections.UserDict

    A class to hold multiple Threat Sources

    new (name: str) → rail.threat_source.ThreatSource
        A method to add a new threat source to the Threat Sources class
```

**class** `rail.Tree` (*name: str, parent=None, sort: bool = True*)

Bases: `collections.UserDict`

A class to implement a tree structure

**add\_child** (*name: str*) → `rail.tree.Tree`

Add a child to the tree

**path** () → `str`

Print the path from the root to the child

**to\_dict\_list** () → `dict`

Print a tree in an alternating dict list format

**to\_latex** () → `None`

Print a tree in LaTeX format

**to\_print** () → `None`

Print all of a tree

**class** `rail.Vulnerabilities` (*\*\*kwargs*)

Bases: `collections.UserDict`

A class to hold multiple Vulnerabilities

**new** (*threat\_event: rail.threat\_event.ThreatEvent, system: rail.tree.Tree, controls: [<class 'rail.control.Control'>] = []*) → `rail.vulnerability.Vulnerability`

**class** `rail.Vulnerability` (*threat\_event: rail.threat\_event.ThreatEvent, system: rail.tree.Tree, controls: [<class 'rail.control.Control'>] = []*)

Bases: `collections.UserDict`

A class to represent Vulnerabilities



## CHAPTER 3

---

### Indices and tables

---

- `genindex`
- `modindex`
- `search`





**r**

rail, 7



**A**

`add_child()` (*rail.Tree method*), 9

**C**

`calculate_dataframe_deterministic_mean()` (*rail.Risks method*), 8

`calculate_stochastic_risks()` (*rail.Risks method*), 8

`Control` (*class in rail*), 7

`Controls` (*class in rail*), 7

`costs()` (*rail.Controls method*), 7

`costs_lognormal()` (*rail.Controls method*), 7

`CPI` (*class in rail*), 7

**D**

`determine_optimum_controls()` (*rail.Risks method*), 8

**E**

`evaluate_deterministic()` (*rail.Risk method*), 8

`evaluate_lognormal()` (*rail.Control method*), 7

`evaluate_lognormal()` (*rail.Risk method*), 8

`expected_loss_deterministic_mean()` (*rail.Risks method*), 8

`expected_loss_stochastic_mean()` (*rail.Risks method*), 8

**F**

`from_lower_90_upper_90()` (*rail.Impact method*), 7

**I**

`Impact` (*class in rail*), 7

`inflation()` (*rail.CPI method*), 7

**L**

`Likelihood` (*class in rail*), 7

**N**

`new()` (*rail.Controls method*), 7

`new()` (*rail.Risks method*), 8

`new()` (*rail.ThreatEvents method*), 8

`new()` (*rail.ThreatSources method*), 8

`new()` (*rail.Vulnerabilities method*), 9

**P**

`path()` (*rail.Tree method*), 9

`plot()` (*rail.Impact method*), 7

`plot()` (*rail.Likelihood method*), 8

`plot()` (*rail.Risks method*), 8

`plot_risk_cost_matrix()` (*rail.Risks method*), 8

**R**

`rail` (*module*), 7

`Risk` (*class in rail*), 8

`Risks` (*class in rail*), 8

**S**

`sensitivity_test()` (*rail.Risks method*), 8

`set_optimum_controls()` (*rail.Risks method*), 8

**T**

`ThreatEvent` (*class in rail*), 8

`ThreatEvents` (*class in rail*), 8

`ThreatSource` (*class in rail*), 8

`ThreatSources` (*class in rail*), 8

`to_dict_list()` (*rail.Tree method*), 9

`to_latex()` (*rail.Tree method*), 9

`to_print()` (*rail.Tree method*), 9

`Tree` (*class in rail*), 8

**V**

`Vulnerabilities` (*class in rail*), 9

`Vulnerability` (*class in rail*), 9