Tyche Example

Set up.

One only needs to execute the following line once, in order to make sure recent enough packages are installed.

In [ ]:
```
!pip install 'numpy>=1.17.2' 'pandas>=0.25.1'
```

Import packages.

In [1]:
```
import os
import sys
sys.path.insert(0, os.path.abspath("../src"))
```

In [2]:
```
import numpy             as np
import matplotlib.pyplot as pl
import pandas            as pd
import re                as re
import scipy.stats       as st
import seaborn           as sb

# The `tyche` package is located at <https://github.com/NREL/portfoli
o/tree/master/production-function/framework/src/tyche/>.
import tyche             as ty

from copy import deepcopy
```

Load data.

The data are stored in a set of tab-separated value files in a folder.

In [3]:
```
designs = ty.Designs("../data/simple_electrolysis")
```

In [4]:
```
investments = ty.Investments("../data/simple_electrolysis")
```

Compile the production and metric functions for each technology in the dataset.

In [5]:
```
designs.compile()
```

Examine the data.

The `functions` table specifies where the Python code for each technology resides.

In [6]:
```
designs.functions
```

Out[6]:

Technology	Style	Module	Capital	Fixed	Production	Metrics	Notes
Simple electrolysis	numpy	simple_electrolysis	capital_cost	fixed_cost	production	metrics	

Right now, only the style `numpy` is supported.

The `indices` table defines the subscripts for variables.

In [7]:
```
designs.indices
```

Out[7]:

Technology	Type	Index	Offset	Description	Notes
Simple electrolysis	Capital	Catalyst	0	Catalyst	
	Fixed	Rent	0	Rent	
	Input	Electricity	1	Electricity	
		Water	0	Water	
	Metric	Cost	0	Cost	
		GHG	2	Greenhouse gas emissions	
		Jobs	1	Jobs	
	Output	Hydrogen	1	Hydrogen	
		Oxygen	0	Oxygen	

The `designs` table contains the cost, input, efficiency, and price data for a scenario.

In [8]:
```
designs.designs
```

Out[8]:

Technology	Scenario	Variable	Index	Value	Units	Notes
Simple electrolysis	Base Electrolysis	Input	Electricity	279	kJ/mole	
			Water	19.04	g/mole	
		Input efficiency	Electricity	0.85	1	
			Water	0.95	1	
		Input price	Electricity	3.33e-5	USD/kJ	
			Water	4.8e-3	USD/mole	
		Lifetime	Catalyst	3	yr	Effective lifetime of Al-Ni catalyst.
		Output efficiency	Hydrogen	0.90	1	
			Oxygen	0.90	1	
		Output price	Hydrogen	1.0e-2	USD/g	
			Oxygen	3.0e-3	USD/g	
		Scale	NaN	6650	mole/yr	Rough estimate for a 50W setup.
	Fast Progress on Electrolysis	Input	Electricity	279	kJ/mole	
			Water	19.04	g/mole	
		Input efficiency	Electricity	st.truncnorm(-3, 0.75, loc=0.97, scale=0.04)	1	
			Water	st.truncnorm(-3, 2, loc=0.97, scale=0.01)	1	
		Input price	Electricity	3.33e-5	USD/kJ	
			Water	4.8e-3	USD/mole	
		Lifetime	Catalyst	3	yr	Effective lifetime of Al-Ni catalyst.
		Output efficiency	Hydrogen	st.beta(3, 2, loc=0.90, scale=0.03)	1	
			Oxygen	st.beta(3, 2, loc=0.90, scale=0.06)	1	
		Output price	Hydrogen	1.0e-2	USD/g	
			Oxygen	3.0e-3	USD/g	
		Scale	NaN	6650	mole/yr	Rough estimate for a 50W setup.
	Moderate Progress on Electrolysis	Input	Electricity	279	kJ/mole	
			Water	19.04	g/mole	

				Value	Units	Notes
Technology	Scenario	Variable	Index			
		Input efficiency	Electricity	st.truncnorm(-2, 1.75, loc=0.93, scale=0.04)	1	
			Water	st.truncnorm(-2, 3, loc=0.97, scale=0.01)	1	
		Input price	Electricity	3.33e-5	USD/kJ	
			Water	4.8e-3	USD/mole	
		Lifetime	Catalyst	3	yr	Effective lifetime of Al-Ni catalyst.
		Output efficiency	Hydrogen	st.beta(2, 2, loc=0.90, scale=0.03)	1	
			Oxygen	st.beta(2, 2, loc=0.90, scale=0.06)	1	
		Output price	Hydrogen	1.0e-2	USD/g	
			Oxygen	3.0e-3	USD/g	
		Scale	NaN	6650	mole/yr	Rough estimate for a 50W setup.
		Input	Electricity	279	kJ/mole	
			Water	19.04	g/mole	
		Input efficiency	Electricity	st.truncnorm(-1, 2.75, loc=0.89, scale=0.04)	1	
			Water	st.truncnorm(-1, 4, loc=0.96, scale=0.01)	1	
		Input price	Electricity	3.33e-5	USD/kJ	
			Water	4.8e-3	USD/mole	
	Slow Progress on Electrolysis	Lifetime	Catalyst	3	yr	Effective lifetime of Al-Ni catalyst.
		Output efficiency	Hydrogen	st.beta(1, 2, loc=0.90, scale=0.03)	1	
			Oxygen	st.beta(1, 2, loc=0.90, scale=0.06)	1	
		Output price	Hydrogen	1.0e-2	USD/g	
			Oxygen	3.0e-3	USD/g	
		Scale	NaN	6650	mole/yr	Rough estimate for a 50W setup.

The `parameters` table contains additional techno-economic parameters for each technology.

In [9]:
```
designs.parameters
```

Out[9]:

Technology	Scenario	Parameter	Offset	Value	Units	Notes
Simple electrolysis	Base Electrolysis	Electricity consumption	3	237	kJ	
		GHG factor for electricity	9	0.138	gCO2e/kJ	based on 1 kWh = 0.5 kg CO2e
		GHG factor for water	8	0.00108	gCO2e/g	based on 244,956 gallons = 1 Mg CO2e
		Hydrogen production	1	2.00	g	
		Jobs	4	1.5e-4	job/mole	
		Oxygen production	0	16.00	g	
		Reference capital cost for catalyst	6	0.63	USD	
		Reference fixed cost for rent	7	1000	USD/yr	
		Reference scale	5	6650	mole/yr	
		Water consumption	2	18.08	g	
	Fast Progress on Electrolysis	Electricity consumption	3	237	kJ	
		GHG factor for electricity	9	0.138	gCO2e/kJ	based on 1 kWh = 0.5 kg CO2e
		GHG factor for water	8	0.00108	gCO2e/g	based on 244,956 gallons = 1 Mg CO2e
		Hydrogen production	1	2.00	g	
		Jobs	4	1.5e-4	job/mole	
		Oxygen production	0	16.00	g	
		Reference capital cost for catalyst	6	0.63	USD	
		Reference fixed cost for rent	7	1000	USD/yr	
		Reference scale	5	6650	mole/yr	
		Water consumption	2	18.08	g	
	Moderate Progress on Electrolysis	Electricity consumption	3	237	kJ	
		GHG factor for electricity	9	0.138	gCO2e/kJ	based on 1 kWh = 0.5 kg CO2e
		GHG factor for water	8	0.00108	gCO2e/g	based on 244,956 gallons = 1 Mg CO2e

			Offset	Value	Units	Notes
Technology	Scenario	Parameter				
		Hydrogen production	1	2.00	g	
		Jobs	4	1.5e-4	job/mole	
		Oxygen production	0	16.00	g	
		Reference capital cost for catalyst	6	0.63	USD	
		Reference fixed cost for rent	7	1000	USD/yr	
		Reference scale	5	6650	mole/yr	
		Water consumption	2	18.08	g	
		Electricity consumption	3	237	kJ	
		GHG factor for electricity	9	0.138	gCO2e/kJ	based on 1 kWh = 0.5 kg CO2e
		GHG factor for water	8	0.00108	gCO2e/g	based on 244,956 gallons = 1 Mg CO2e
	Slow Progress on Electrolysis	Hydrogen production	1	2.00	g	
		Jobs	4	1.5e-4	job/mole	
		Oxygen production	0	16.00	g	
		Reference capital cost for catalyst	6	0.63	USD	
		Reference fixed cost for rent	7	1000	USD/yr	
		Reference scale	5	6650	mole/yr	
		Water consumption	2	18.08	g	

The `results` table specifies the units of measure for results of computations.

In [10]:
```
designs.results
```

Out[10]:

Technology	Variable	Index	Units	Notes
Simple electrolysis	Cost	Cost	USD/mole	
	Metric	Cost	USD/gH2	
		GHG	gCO2e/gH2	
		Jobs	job/gH2	
	Output	Hydrogen	g/mole	
		Oxygen	g/mole	

The `tranches` table specifies multually exclusive possibilities for investments: only one `Tranch` may be selected for each `Cateogry`.

In [11]:
```
investments.tranches
```

Out[11]:

Category	Tranche	Scenario	Notes
Electrolysis R&D	High Electrolysis R&D	Fast Progress on Electrolysis	
	Low Electrolysis R&D	Slow Progress on Electrolysis	
	Medium Electrolysis R&D	Moderate Progress on Electrolysis	
	No Electrolysis R&D	Base Electrolysis	

The `investments` table bundles a consistent set of tranches (one per category) into an overall investment.

In [12]:
```
investments.investments
```

Out[12]:

Investment	Category	Tranche	Amount	Notes
High R&D Spending	Electrolysis R&D	High Electrolysis R&D	5000000.0	
Low R&D Spending	Electrolysis R&D	Low Electrolysis R&D	1000000.0	
Medium R&D Spending	Electrolysis R&D	Medium Electrolysis R&D	2500000.0	
No R&D Spending	Electrolysis R&D	No Electrolysis R&D	0.0	

Evaluate the scenarios in the dataset.

```
In [13]: scenario_results = designs.evaluate_scenarios(sample_count=50)
```

```
In [14]: scenario_results.xs(1, level="Sample", drop_level=False)
```

Out[14]:

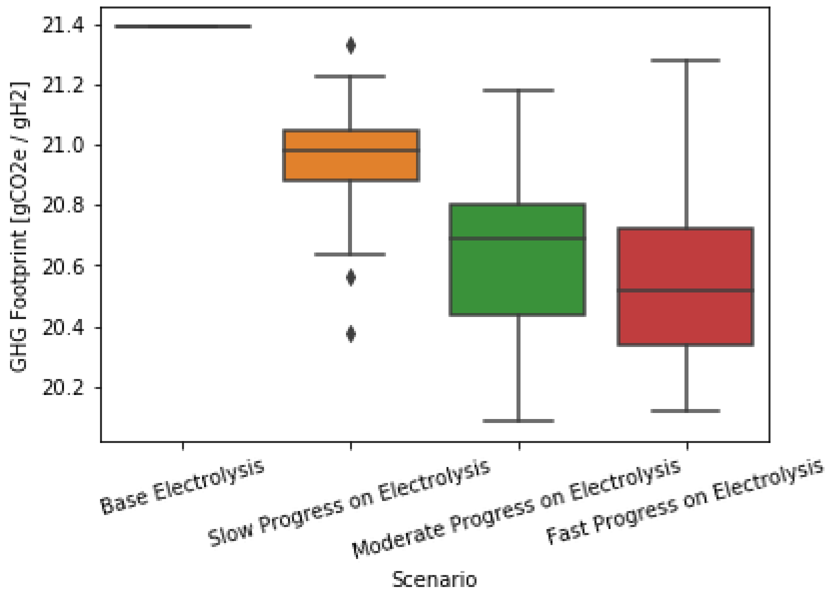
Technology	Scenario	Sample	Variable	Index	Value	Units
Simple electrolysis	Base Electrolysis	1	Cost	Cost	0.183900	USD/mole
			Metric	Cost	0.102121	USD/gH2
				GHG	21.391959	gCO2e/gH2
				Jobs	0.000083	job/gH2
			Output	Hydrogen	1.800796	g/mole
				Oxygen	14.406372	g/mole
	Fast Progress on Electrolysis	1	Cost	Cost	0.182967	USD/mole
			Metric	Cost	0.096508	USD/gH2
				GHG	20.319064	gCO2e/gH2
				Jobs	0.000079	job/gH2
			Output	Hydrogen	1.895883	g/mole
				Oxygen	15.412336	g/mole
	Moderate Progress on Electrolysis	1	Cost	Cost	0.182973	USD/mole
			Metric	Cost	0.098984	USD/gH2
				GHG	20.839843	gCO2e/gH2
				Jobs	0.000081	job/gH2
			Output	Hydrogen	1.848505	g/mole
				Oxygen	14.958624	g/mole
	Slow Progress on Electrolysis	1	Cost	Cost	0.184589	USD/mole
			Metric	Cost	0.100312	USD/gH2
				GHG	20.934504	gCO2e/gH2
				Jobs	0.000082	job/gH2
			Output	Hydrogen	1.840147	g/mole
				Oxygen	14.734273	g/mole

Save results.

```
In [15]: scenario_results.to_csv("output/simple_electrolysis/example-scenario.
csv")
```

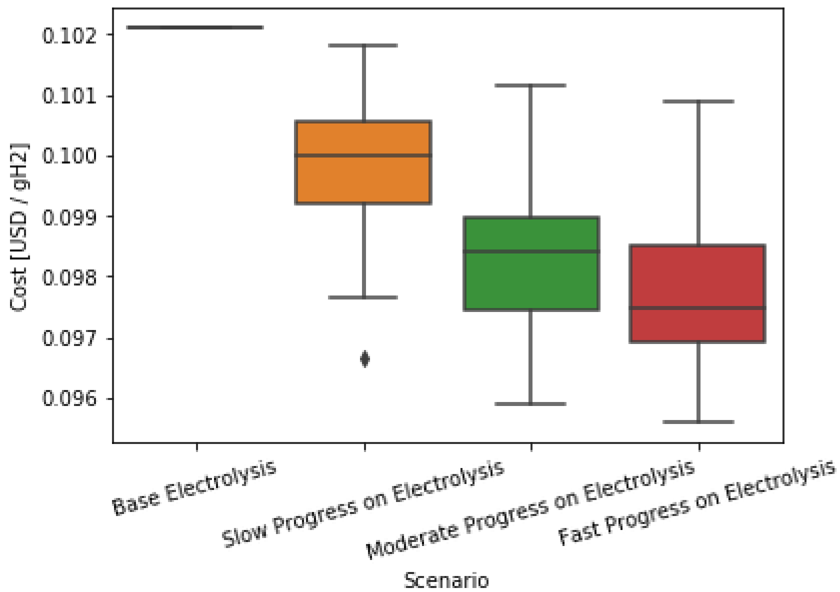
Plot GHG metric.

In [16]:
```
g = sb.boxplot(
    x="Scenario",
    y="Value",
    data=scenario_results.xs(
        ["Metric", "GHG"],
        level=["Variable", "Index"]
    ).reset_index()[["Scenario", "Value"]],
    order=["Base Electrolysis", "Slow Progress on Electrolysis", "Mod
erate Progress on Electrolysis", "Fast Progress on Electrolysis"]
)
g.set(ylabel="GHG Footprint [gCO2e / gH2]")
g.set_xticklabels(g.get_xticklabels(), rotation=15);
```



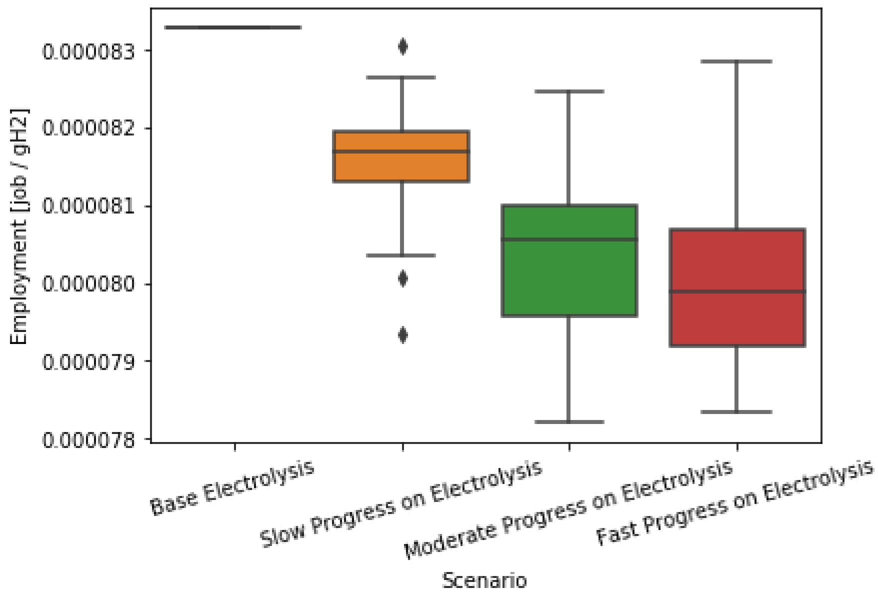
Plot cost metric.

In [17]:
```
g = sb.boxplot(
    x="Scenario",
    y="Value",
    data=scenario_results.xs(
        ["Metric", "Cost"],
        level=["Variable", "Index"]
    ).reset_index()[["Scenario", "Value"]],
    order=["Base Electrolysis", "Slow Progress on Electrolysis", "Mod
erate Progress on Electrolysis", "Fast Progress on Electrolysis"]
)
g.set(ylabel="Cost [USD / gH2]")
g.set_xticklabels(g.get_xticklabels(), rotation=15);
```



Plot employment metric.

In [18]:
```
g = sb.boxplot(
    x="Scenario",
    y="Value",
    data=scenario_results.xs(
        ["Metric", "Jobs"],
        level=["Variable", "Index"]
    ).reset_index()[["Scenario", "Value"]],
    order=["Base Electrolysis", "Slow Progress on Electrolysis", "Mod
erate Progress on Electrolysis", "Fast Progress on Electrolysis"]
)
g.set(ylabel="Employment [job / gH2]")
g.set_xticklabels(g.get_xticklabels(), rotation=15);
```



Evaluate the investments in the dataset.

In [19]:
```
investment_results = investments.evaluate_investments(designs, sample
_count=50)
```

Costs of investments.

In [20]:
```
investment_results.amounts
```

Out[20]:

	Amount
Investment	
High R&D Spending	5000000.0
Low R&D Spending	1000000.0
Medium R&D Spending	2500000.0
No R&D Spending	0.0

Benefits of investments.

In [21]:
```
investment_results.metrics.xs(1, level="Sample", drop_level=False)
```

Out[21]:

Investment	Category	Tranche	Scenario	Sample	Technology	Index	Value	Ur
No R&D Spending	Electrolysis R&D	No Electrolysis R&D	Base Electrolysis	1	Simple electrolysis	Cost	0.102121	USD/g
						GHG	21.391959	gCO2e/g
						Jobs	0.000083	job/g
High R&D Spending	Electrolysis R&D	High Electrolysis R&D	Fast Progress on Electrolysis	1	Simple electrolysis	Cost	0.097777	USD/g
						GHG	20.438118	gCO2e/g
						Jobs	0.000080	job/g
Medium R&D Spending	Electrolysis R&D	Medium Electrolysis R&D	Moderate Progress on Electrolysis	1	Simple electrolysis	Cost	0.100710	USD/g
						GHG	21.079248	gCO2e/g
						Jobs	0.000082	job/g
Low R&D Spending	Electrolysis R&D	Low Electrolysis R&D	Slow Progress on Electrolysis	1	Simple electrolysis	Cost	0.101387	USD/g
						GHG	21.176323	gCO2e/g
						Jobs	0.000082	job/g

In [22]:
```
investment_results.summary.xs(1, level="Sample", drop_level=False)
```

Out[22]:

Investment	Sample	Index	Value	Units
No R&D Spending	1	Cost	0.102121	USD/gH2
		GHG	21.391959	gCO2e/gH2
		Jobs	0.000083	job/gH2
High R&D Spending	1	Cost	0.097777	USD/gH2
		GHG	20.438118	gCO2e/gH2
		Jobs	0.000080	job/gH2
Medium R&D Spending	1	Cost	0.100710	USD/gH2
		GHG	21.079248	gCO2e/gH2
		Jobs	0.000082	job/gH2
Low R&D Spending	1	Cost	0.101387	USD/gH2
		GHG	21.176323	gCO2e/gH2
		Jobs	0.000082	job/gH2

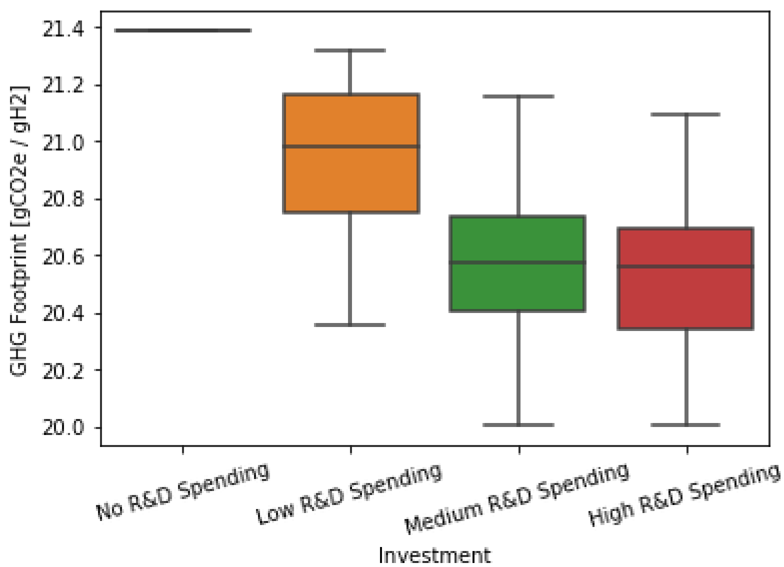
Save results.

In [23]:
```
investment_results.amounts.to_csv("output/simple_electrolysis/example
-investment-amounts.csv")
```

In [24]:
```
investment_results.metrics.to_csv("output/simple_electrolysis/example
-investment-metrics.csv")
```

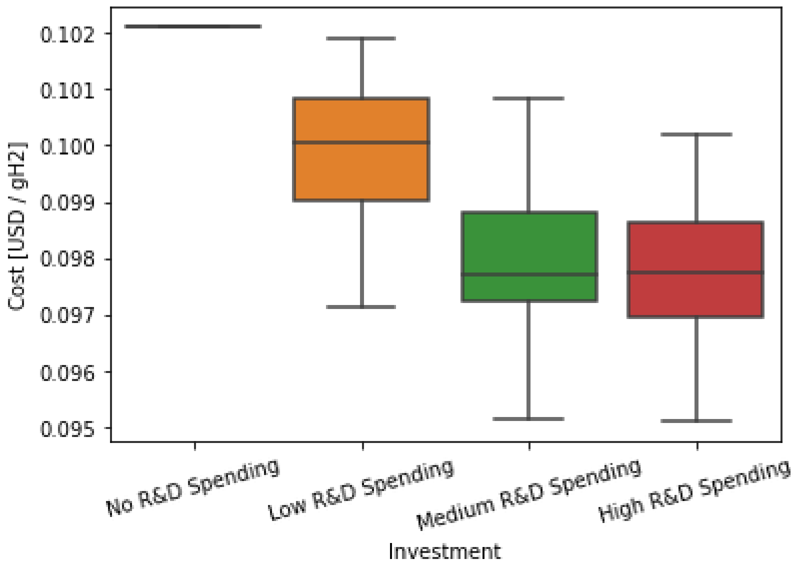
Plot GHG metric.

In [25]:
```
g = sb.boxplot(
    x="Investment",
    y="Value",
    data=investment_results.metrics.xs(
        "GHG",
        level="Index"
    ).reset_index()[["Investment", "Value"]],
    order=["No R&D Spending", "Low R&D Spending", "Medium R&D Spendin
g", "High R&D Spending"]
)
g.set(ylabel="GHG Footprint [gCO2e / gH2]")
g.set_xticklabels(g.get_xticklabels(), rotation=15);
```



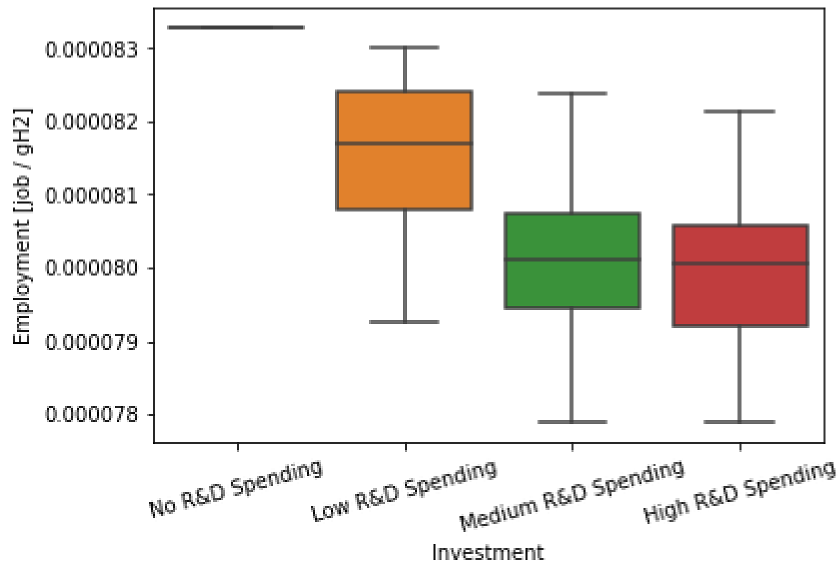
Plot cost metric.

In [26]:
```
g = sb.boxplot(
    x="Investment",
    y="Value",
    data=investment_results.metrics.xs(
        "Cost",
        level="Index"
    ).reset_index()[["Investment", "Value"]],
    order=["No R&D Spending", "Low R&D Spending", "Medium R&D Spendin
g", "High R&D Spending"]
)
g.set(ylabel="Cost [USD / gH2]")
g.set_xticklabels(g.get_xticklabels(), rotation=15);
```



Plot employment metric.

In [27]:
```python
g = sb.boxplot(
    x="Investment",
    y="Value",
    data=investment_results.metrics.xs(
        "Jobs",
        level="Index"
    ).reset_index()[["Investment", "Value"]],
    order=["No R&D Spending", "Low R&D Spending", "Medium R&D Spendin
g", "High R&D Spending"]
)
g.set(ylabel="Employment [job / gH2]")
g.set_xticklabels(g.get_xticklabels(), rotation=15);
```



Sensitity analysis.

Vary the four efficiencies in the design.

In [28]:
```python
# Four variables are involved.
variables = [
    ("Input efficiency" , "Water"      ),
    ("Input efficiency" , "Electricity"),
    ("Output efficiency", "Oxygen"     ),
    ("Output efficiency", "Hydrogen"   ),
]
```

In [29]:
```python
# Let efficiencies range from 0.75 to 0.975.
efficiencies = np.arange(0.750, 1.000, 0.025)
efficiencies
```

Out[29]:
```
array([0.75 , 0.775, 0.8  , 0.825, 0.85 , 0.875, 0.9  , 0.925, 0.95 ,
       0.975])
```

Start from the base case.

In [30]:
```
base_design = designs.designs.xs("Base Electrolysis", level=1, drop_level=False)
base_design
```

Out[30]:

Technology	Scenario	Variable	Index	Value	Units	Notes
Simple electrolysis	Base Electrolysis	Input	Electricity	279	kJ/mole	
			Water	19.04	g/mole	
		Input efficiency	Electricity	0.85	1	
			Water	0.95	1	
		Input price	Electricity	3.33e-5	USD/kJ	
			Water	4.8e-3	USD/mole	
		Lifetime	Catalyst	3	yr	Effective lifetime of Al-Ni catalyst.
		Output efficiency	Hydrogen	0.90	1	
			Oxygen	0.90	1	
		Output price	Hydrogen	1.0e-2	USD/g	
			Oxygen	3.0e-3	USD/g	
		Scale	NaN	6650	mole/yr	Rough estimate for a 50W setup.

In [31]:
```
base_parameters = designs.parameters.xs("Base Electrolysis", level=1,
drop_level=False)
base_parameters
```

Out[31]:

Technology	Scenario	Parameter	Offset	Value	Units	Notes
Simple electrolysis	Base Electrolysis	Electricity consumption	3	237	kJ	
		GHG factor for electricity	9	0.138	gCO2e/kJ	based on 1 kWh = 0.5 kg CO2e
		GHG factor for water	8	0.00108	gCO2e/g	based on 244,956 gallons = 1 Mg CO2e
		Hydrogen production	1	2.00	g	
		Jobs	4	1.5e-4	job/mole	
		Oxygen production	0	16.00	g	
		Reference capital cost for catalyst	6	0.63	USD	
		Reference fixed cost for rent	7	1000	USD/yr	
		Reference scale	5	6650	mole/yr	
		Water consumption	2	18.08	g	

Generate the new scenarios and append them to the previous ones.

In [32]:
```
sensitivities = deepcopy(designs)
sensitivities.designs = sensitivities.designs[0:0]
sensitivities.parameters = sensitivities.parameters[0:0]
```

In [33]:
```
# Iterate over variables and efficiencies.
for variable, index in variables:
    for efficiency in efficiencies:

        # Name the scenario.
        scenario = "Let " + variable + " @ " + index + " = " + str(ro
und(efficiency, 3))

        # Alter the base case.
        vary_design = base_design.rename(index={"Base Electrolysis" :
scenario}, level=1)
        vary_design.loc[("Simple electrolysis", scenario, variable, i
ndex), "Value"] = efficiency

        # Keep the parameters the same.
        vary_parameters = base_parameters.rename(index={"Base Electro
lysis" : scenario}, level=1)

        # Append the results to the existing table of scenarios.
        sensitivities.designs = sensitivities.designs.append(vary_des
ign)
        sensitivities.parameters = sensitivities.parameters.append(va
ry_parameters)
```

Remember to compile the design, since we've added scenarios.

In [34]:
```
sensitivities.compile()
```

See how many rows there are in the tables now.

In [35]:
```
sensitivities.designs.shape
```

Out[35]:
```
(480, 3)
```

In [36]:
```
sensitivities.parameters.shape
```

Out[36]:
```
(400, 4)
```

In [37]:
```
sensitivities.designs
```

Out[37]:

Technology	Scenario	Variable	Index	Value	Units	Notes
Simple electrolysis	Let Input efficiency @ Water = 0.75	Input	Electricity	279	kJ/mole	
			Water	19.04	g/mole	
		Input efficiency	Electricity	0.85	1	
			Water	0.75	1	
		Input price	Electricity	3.33e-5	USD/kJ	
	...	...	...	...	...	...
	Let Output efficiency @ Hydrogen = 0.975	Output efficiency	Hydrogen	0.975	1	
			Oxygen	0.90	1	
		Output price	Hydrogen	1.0e-2	USD/g	
			Oxygen	3.0e-3	USD/g	
		Scale	NaN	6650	mole/yr	Rough estimate for a 50W setup.

480 rows × 3 columns

Compute the results.

In [38]:
```
results = sensitivities.evaluate_scenarios(1)
results
```
Out[38]:

					Value	Units
Technology	Scenario	Sample	Variable	Index		
Simple electrolysis	Let Input efficiency @ Electricity = 0.75	1	Cost	Cost	0.190164	USD/mole
			Metric	Cost	0.119657	USD/gH2
				GHG	24.239606	gCO2e/gH2
				Jobs	0.000094	job/gH2
			Output	Hydrogen	1.589241	g/mole
	...	...	...	...	...	...
	Let Output efficiency @ Oxygen = 0.975	1	Metric	Cost	0.100121	USD/gH2
				GHG	21.391959	gCO2e/gH2
				Jobs	0.000083	job/gH2
			Output	Hydrogen	1.800796	g/mole
				Oxygen	15.606903	g/mole

240 rows × 2 columns

Plot the cost results.

In [39]:
```
cost_results = results.xs("Cost", level="Variable").reset_index()[["S
cenario", "Value"]]
```
In [40]:
```
cost_results[0:10]
```
Out[40]:

	Scenario	Value
0	Let Input efficiency @ Electricity = 0.75	0.190164
1	Let Input efficiency @ Electricity = 0.775	0.188595
2	Let Input efficiency @ Electricity = 0.8	0.187026
3	Let Input efficiency @ Electricity = 0.825	0.185457
4	Let Input efficiency @ Electricity = 0.85	0.183900
5	Let Input efficiency @ Electricity = 0.875	0.184132
6	Let Input efficiency @ Electricity = 0.9	0.184364
7	Let Input efficiency @ Electricity = 0.925	0.184597
8	Let Input efficiency @ Electricity = 0.95	0.184829
9	Let Input efficiency @ Electricity = 0.975	0.185061

In [41]:
```
cost_results["Variable"  ] = cost_results["Scenario"].apply(lambda x:
re.sub(r'^Let (.*) @ (.*) =.*$', '\\1[\\2]', x))
cost_results["Efficiency"] = cost_results["Scenario"].apply(lambda x:
float(re.sub(r'^.*= (.*)$', '\\1', x)))
cost_results["Cost [USD/mole]"] = cost_results["Value"]
```

In [42]:
```
cost_results = cost_results[["Variable", "Efficiency", "Cost [USD/mol
e]"]]
cost_results[0:10]
```

Out[42]:

	Variable	Efficiency	Cost [USD/mole]
0	Input efficiency[Electricity]	0.750	0.190164
1	Input efficiency[Electricity]	0.775	0.188595
2	Input efficiency[Electricity]	0.800	0.187026
3	Input efficiency[Electricity]	0.825	0.185457
4	Input efficiency[Electricity]	0.850	0.183900
5	Input efficiency[Electricity]	0.875	0.184132
6	Input efficiency[Electricity]	0.900	0.184364
7	Input efficiency[Electricity]	0.925	0.184597
8	Input efficiency[Electricity]	0.950	0.184829
9	Input efficiency[Electricity]	0.975	0.185061

In [43]:
```python
# Here is a really simple plot.
cost_results.plot(
    x="Efficiency",
    y="Cost [USD/mole]",
    c=cost_results["Variable"].apply(lambda v: {
        "Input efficiency[Water]"       : "blue"  ,
        "Input efficiency[Electricity]" : "orange",
        "Output efficiency[Oxygen]"     : "green" ,
        "Output efficiency[Hydrogen]"   : "red"   ,
    }[v]),
    kind="scatter"
)
```

Out[43]: <matplotlib.axes._subplots.AxesSubplot at 0x7fb5b6e478d0>