Sensing is life

End-to-End Causal Analysis in Python with cause2e

Daniel Grünbaum, Maike Stern, Elmar W. Lang (R&D @ ams OSRAM / Universität Regensburg)
16.11.21

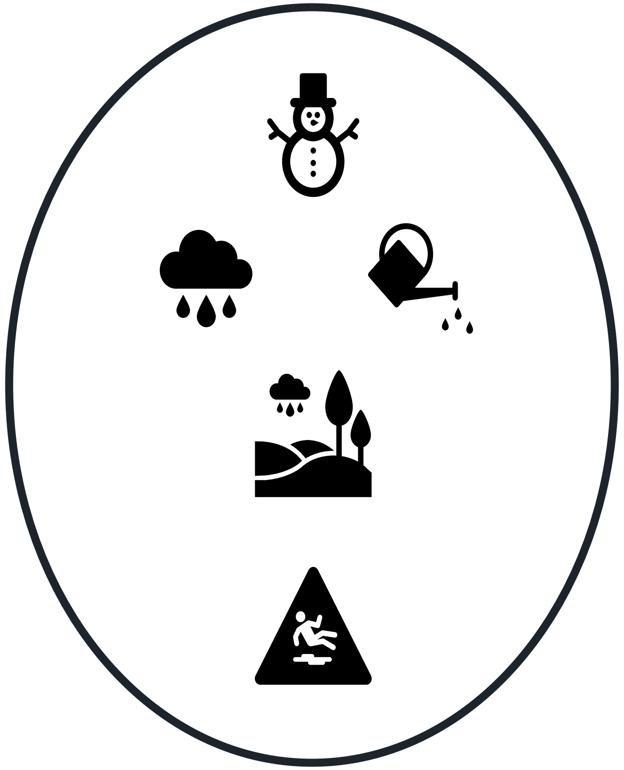
Graph-based Causal Inference

Sprinkler example

Example: „Does my lawn get slippery if I turn on the sprinkler?“

Given: observational data of the form

Season	Sprinkler	Rain	Wet	Slippery
Summer	1	0	1	1
Spring	0	1	1	1
Winter	0	0	0	0



Goal: Determine all causal mechanisms!

Graph-based Causal Inference

Causal discovery

Goal: Learn the causal graph from data and domain knowledge!

Season	Sprinkler	Rain	Wet	Slippery
Summer	1	0	1	1
Spring	0	1	1	1
Winter	0	0	0	0

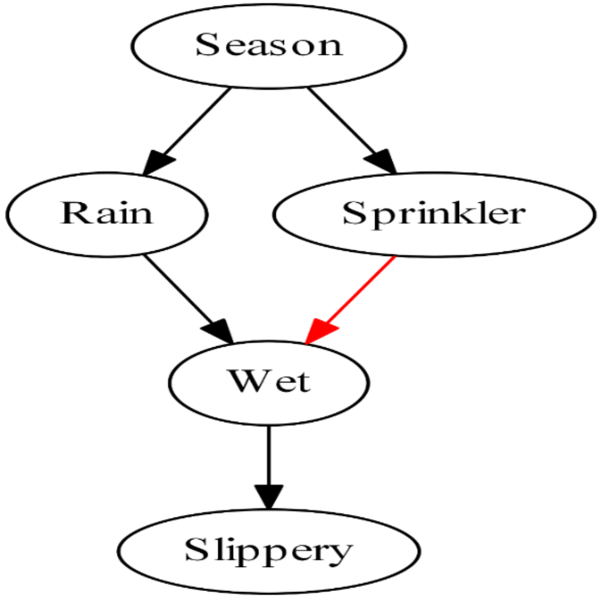
Domain knowledge

Nothing causes `Season`

`Slippery` causes nothing

`Sprinkler` does not cause `Rain`

`Sprinkler` directly causes `Wet`



Software: Py-causal [1], Causal Discovery Toolbox [2]

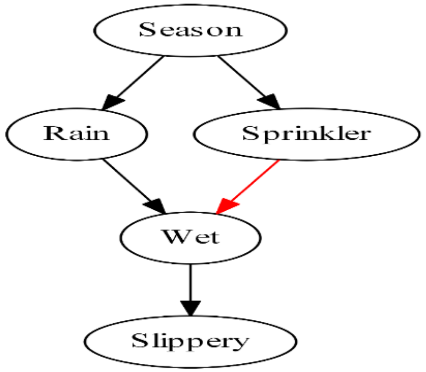
Daniel Grünbaum, Maike Stern, Elmar W. Lang

Graph-based Causal Inference

Causal estimation

Goal: Estimate causal effects from data and the causal graph!

Season	Sprinkler	Rain	Wet	Slippery
Summer	1	0	1	1
Spring	0	1	1	1
Winter	0	0	0	0



„Turning on the sprinkler strongly increases the chances of the lawn being slippery!“

Software: DoWhy [3]

Goals of cause2e

Connect causal discovery and causal estimation

Domain knowledge

Nothing causes `Season`

`Slippery` causes nothing

`Sprinkler` does not cause `Rain`

`Sprinkler` directly causes `Wet`

Data

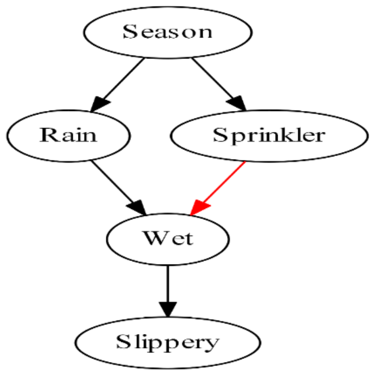
Season	Sprinkler	Rain	Wet	Slippery
Summer	1	0	1	1
Spring	0	1	1	1
Winter	0	0	0	0

Causal estimates

`Wet` does not causally affect `Rain`

Turning on `Sprinkler` strongly increases `Slippery`

Graphical model



Daniel Grünbaum, Maike Stern, Elmar W. Lang

Goals of cause2e

End-to-end causal analysis

Main steps:

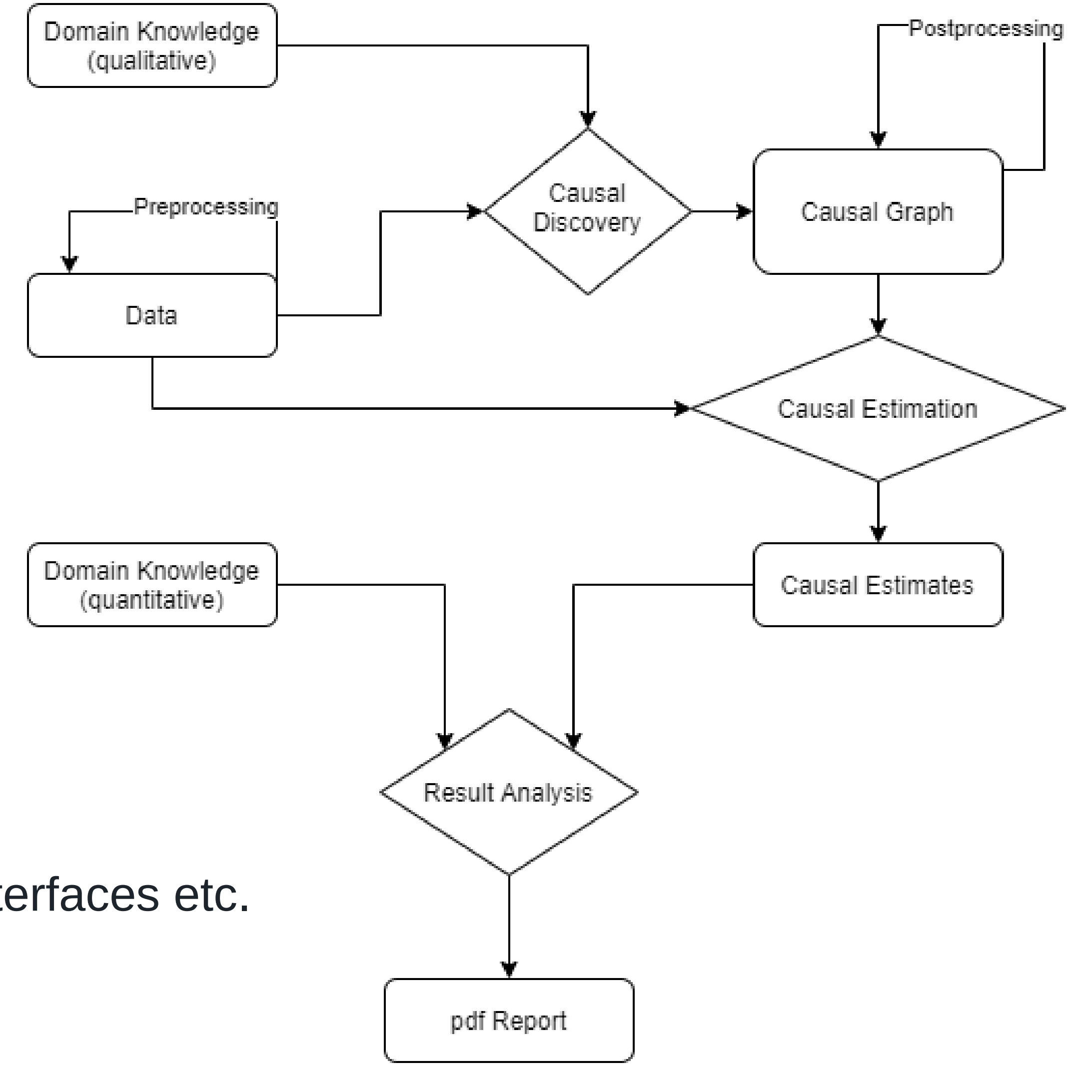
- causal discovery via py-causal
- causal estimation via DoWhy

Additional steps:

- data reading + preprocessing
- graph postprocessing
- result analysis + validation
- reporting

Benefits:

- full causal analysis
- less time spent coding, bug hunting, figuring out interfaces etc.



Daniel Grünbaum, Maike Stern, Elmar W. Lang

Example Analysis

Step-by-step walkthrough

Let‘s solve the sprinkler example together!

Full code provided!
All figures automatically generated by cause2e!

Preparations

Setting the stage

handles paths for data, output and temporary files

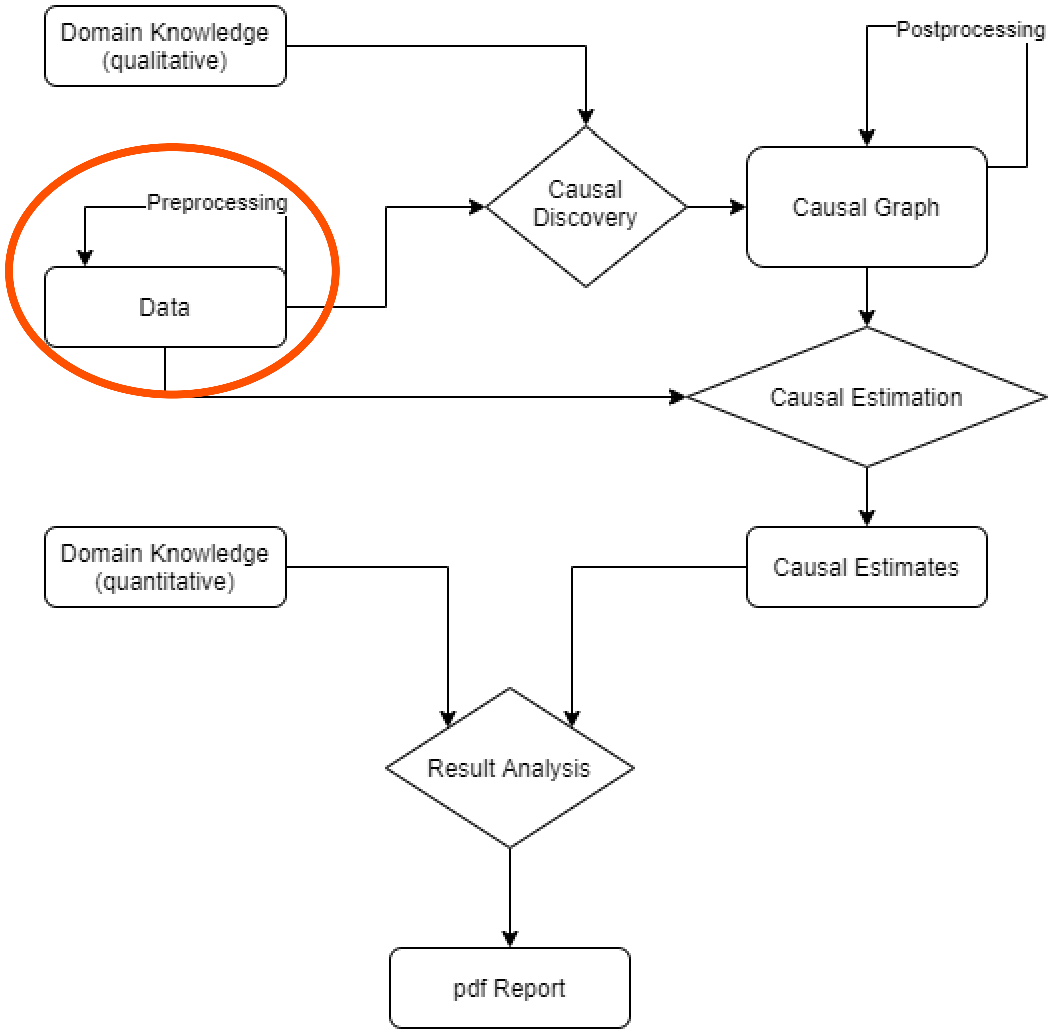
```
import os
from cause2e import path_mgr, knowledge, discovery

cwd = os.getcwd()
paths = path_mgr.PathManager(experiment_name='sprinkler',
                             data_name='sprinkler.csv',
                             data_dir=cwd,
                             output_dir=os.path.join(cwd, 'output')
                             )

learner = discovery.StructureLearner(paths)
```

Data Preprocessing

Where are we?

done

next



Data Preprocessing

Read and specify types

specify datatypes

```
learner.read_csv(index_col=0)

learner.categorical = learner.variables
learner.continuous = set()
```

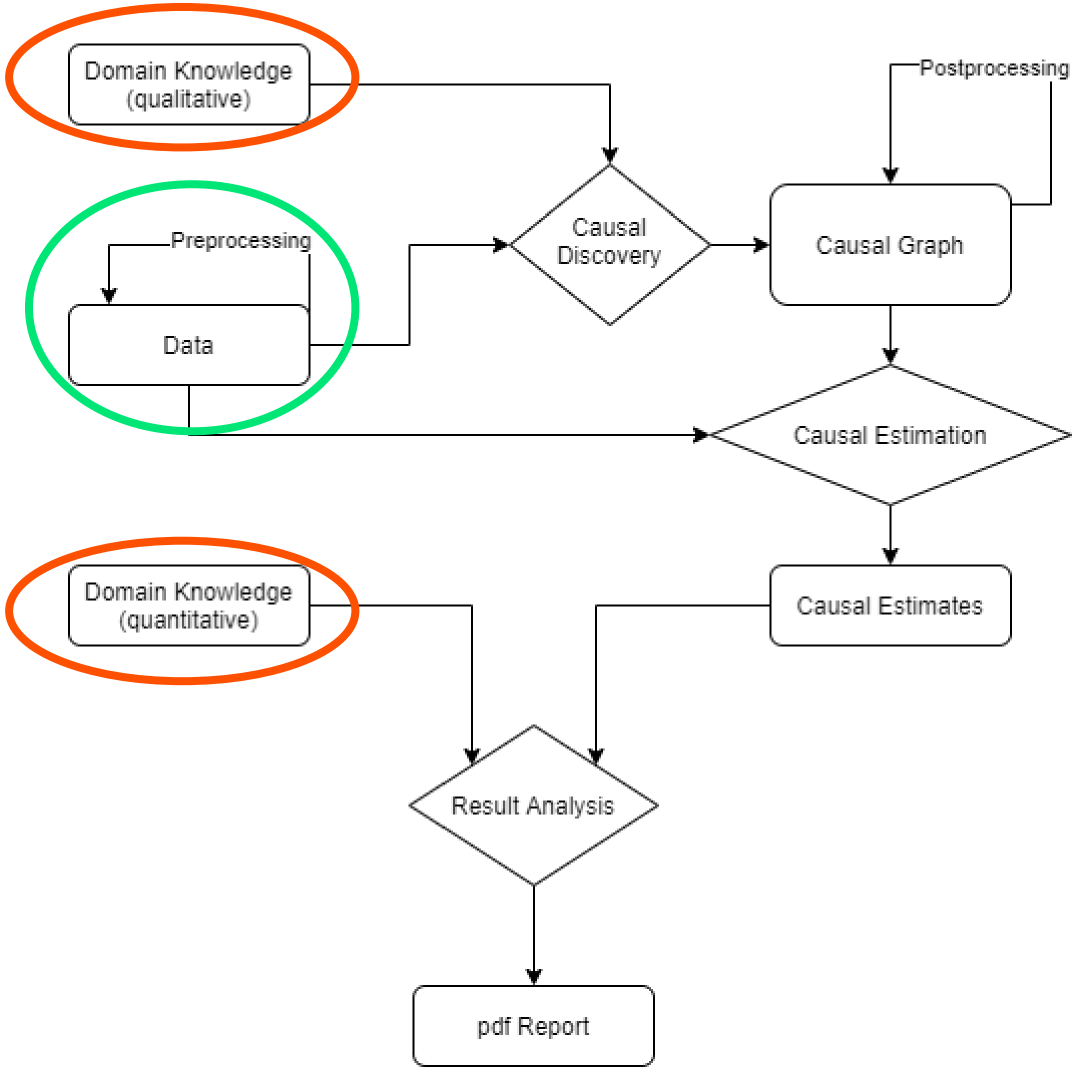
other available preprocessing options: variable selection, normalization, combination…

Passing Domain Knowledge

Where are we?

done

next



Passing Domain Knowledge

Constraining the causal graph

Specify required and forbidden direct causal influences

```
edge_creator = knowledge.EdgeCreator()
edge_creator.forbid_edges_from_groups({'Season'}, incoming=learner.variables)
edge_creator.forbid_edges_from_groups({'Slippery'}, outgoing=learner.variables)
edge_creator.forbid_edge('Sprinkler', 'Rain')
edge_creator.require_edge('Sprinkler', 'Wet')
```

Domain knowledge

Nothing causes `Season`

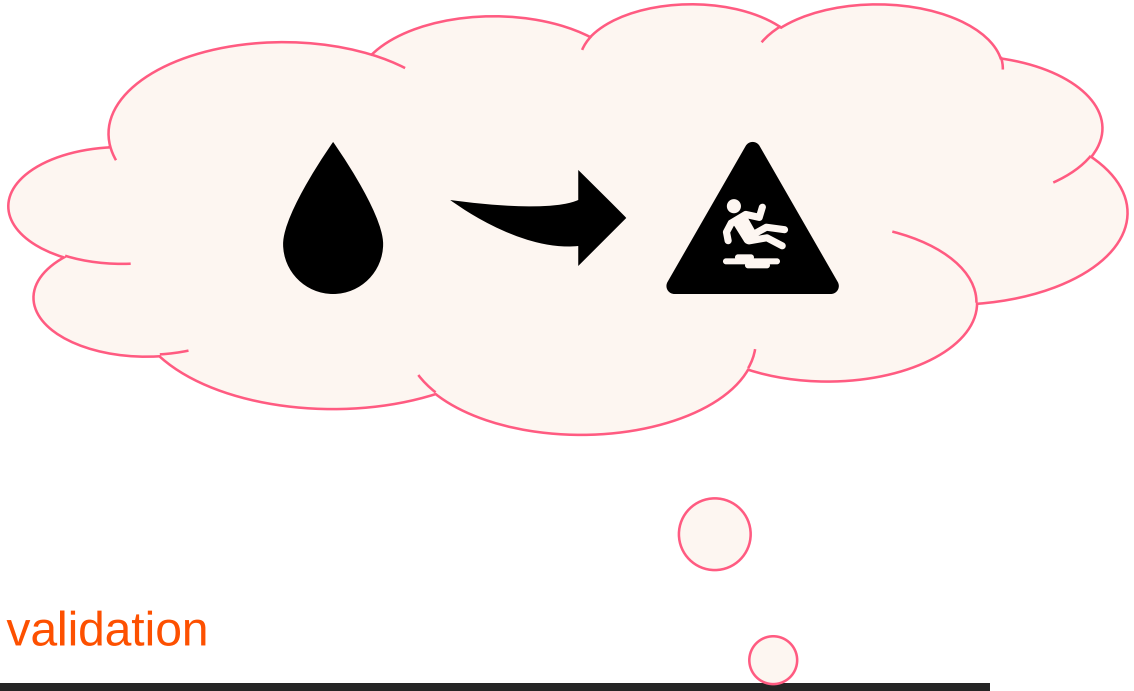
`Slippery` causes nothing

`Sprinkler` does not cause `Rain`

`Sprinkler` directly causes `Wet`

Passing Domain Knowledge

Stating your expectations upfront



Specify known quantitative causal effects for model validation

```
validation_creator = knowledge.ValidationCreator()
validation_creator.add_expected_effect(('Sprinkler', 'Wet', 'nonparametric-ate'), ('greater', 0))
validation_creator.add_expected_effect(('Wet', 'Slippery', 'nonparametric-ate'), ('greater', 0))
validation_creator.add_expected_effect(('Sprinkler', 'Rain', 'nonparametric-nde'), ('less', 0))
validation_creator.add_expected_effect(('Slippery', 'Season', 'nonparametric-nie'), ('between', 0.2, 0.4))
```

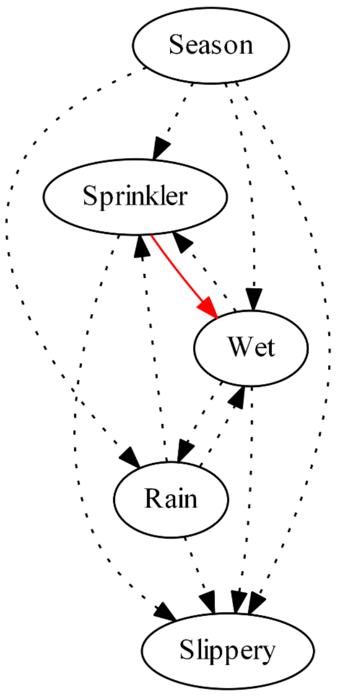

Daniel Grünbaum, Maike Stern, Elmar W. Lang

Passing Domain Knowledge

Checking qualitative inputs

The knowledge graph summarizes our qualitative domain knowledge:

```
learner.set_knowledge(edge_creator=edge_creator, validation_creator=validation_creator)
```



Red: required edges

Missing: forbidden edges

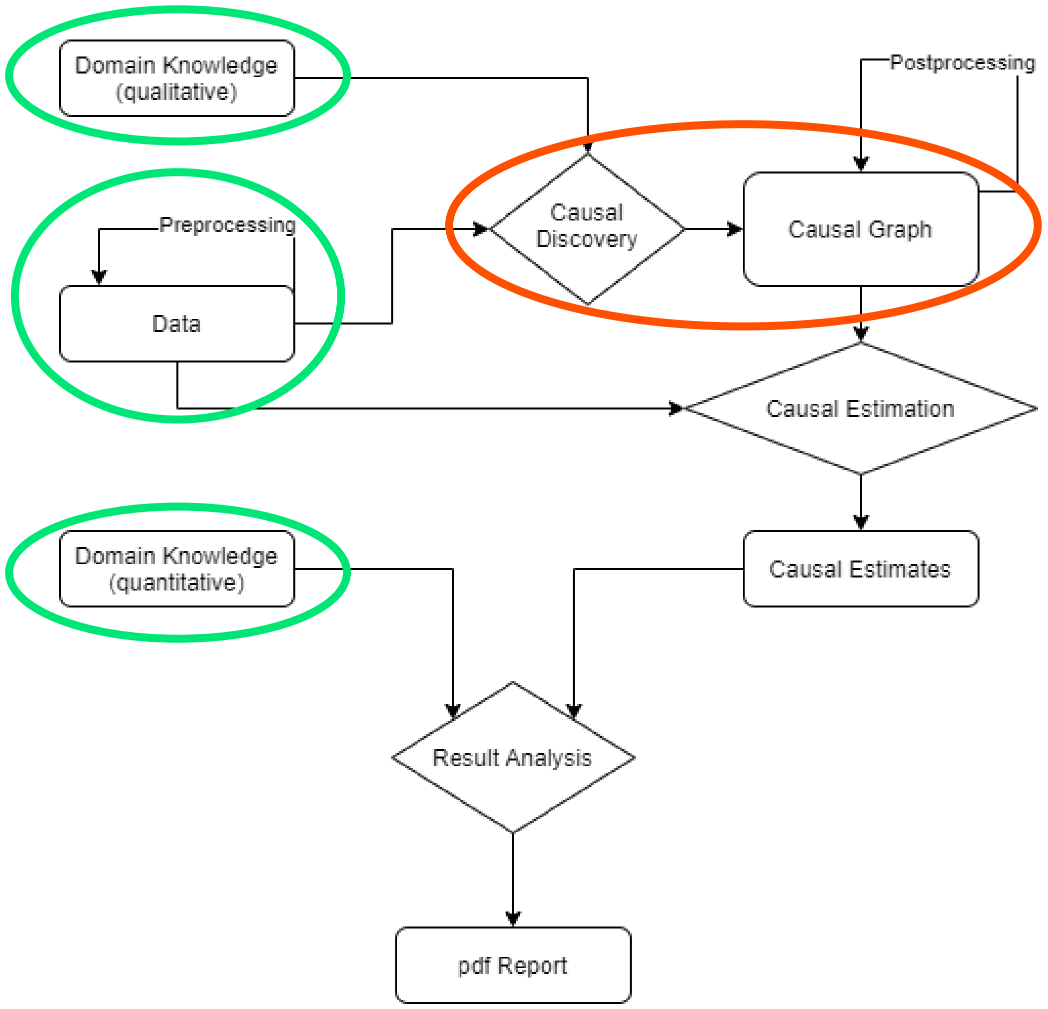
Dotted: remaining allowed edges

Causal Discovery

Where are we?

done

next

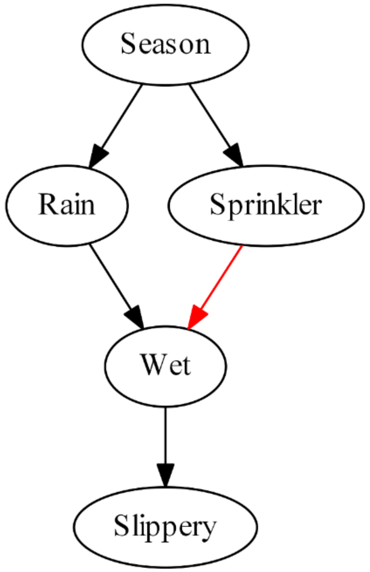


Causal Discovery

Recovering the causal graph from data and domain knowledge

- Run Fast Greedy Equivalence Search (or other discovery algorithms)

```
learner.run_quick_search()
```



Ignored allowed edges:

Source	Destination
Rain	Sprinkler
Wet	Sprinkler
Sprinkler	Slippery
Season	Slippery
Season	Wet
Wet	Rain
Rain	Slippery

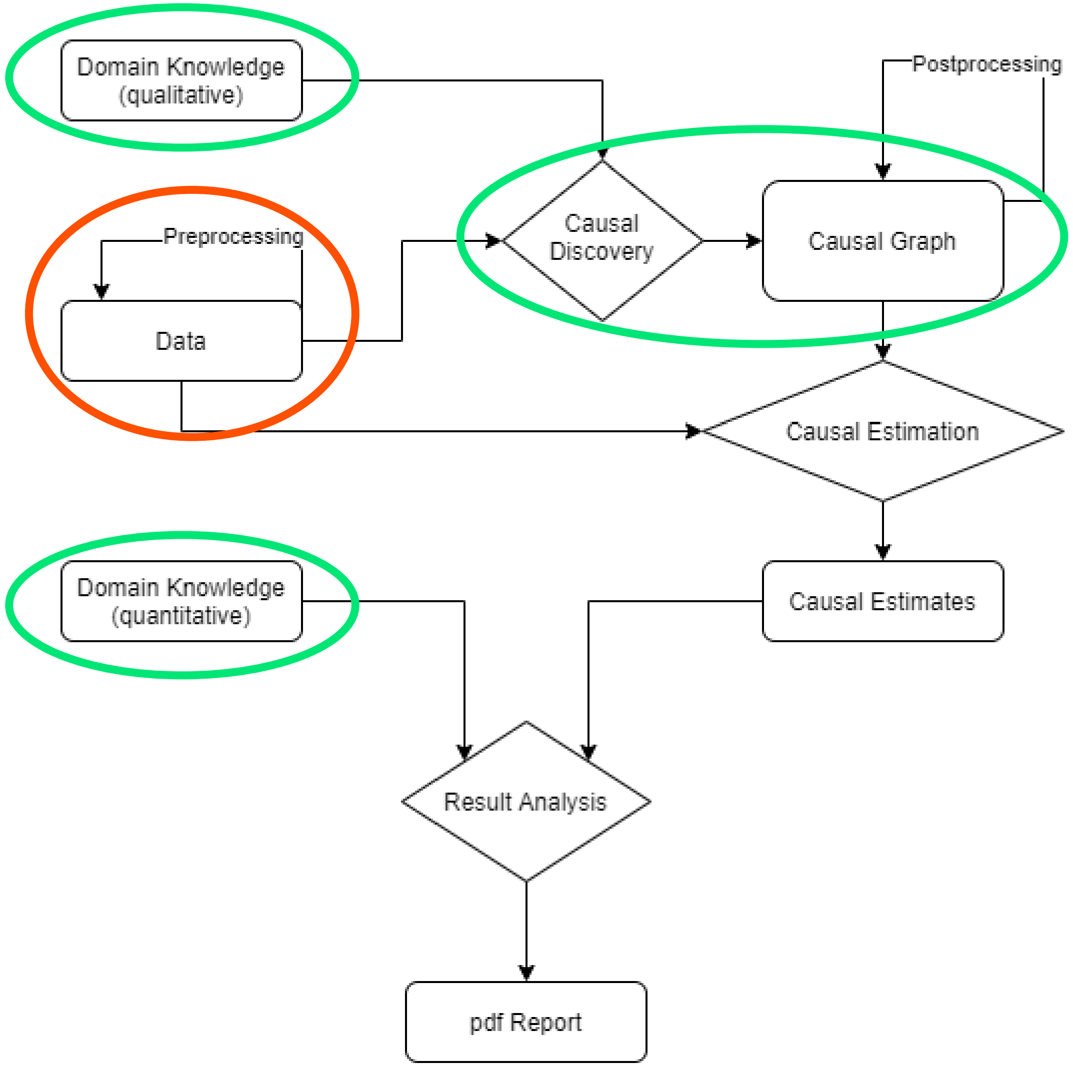
- optional: graph postprocessing and saving

Data Preprocessing

Where are we?

done

next



Data Preprocessing

Binarize categorical treatment

Spring means „on“

Winter means „off“

```
learner.binarize_variable('Season', one_val='Spring', zero_val='Winter')
```

alternative method: aggregate the effect over all possible encodings

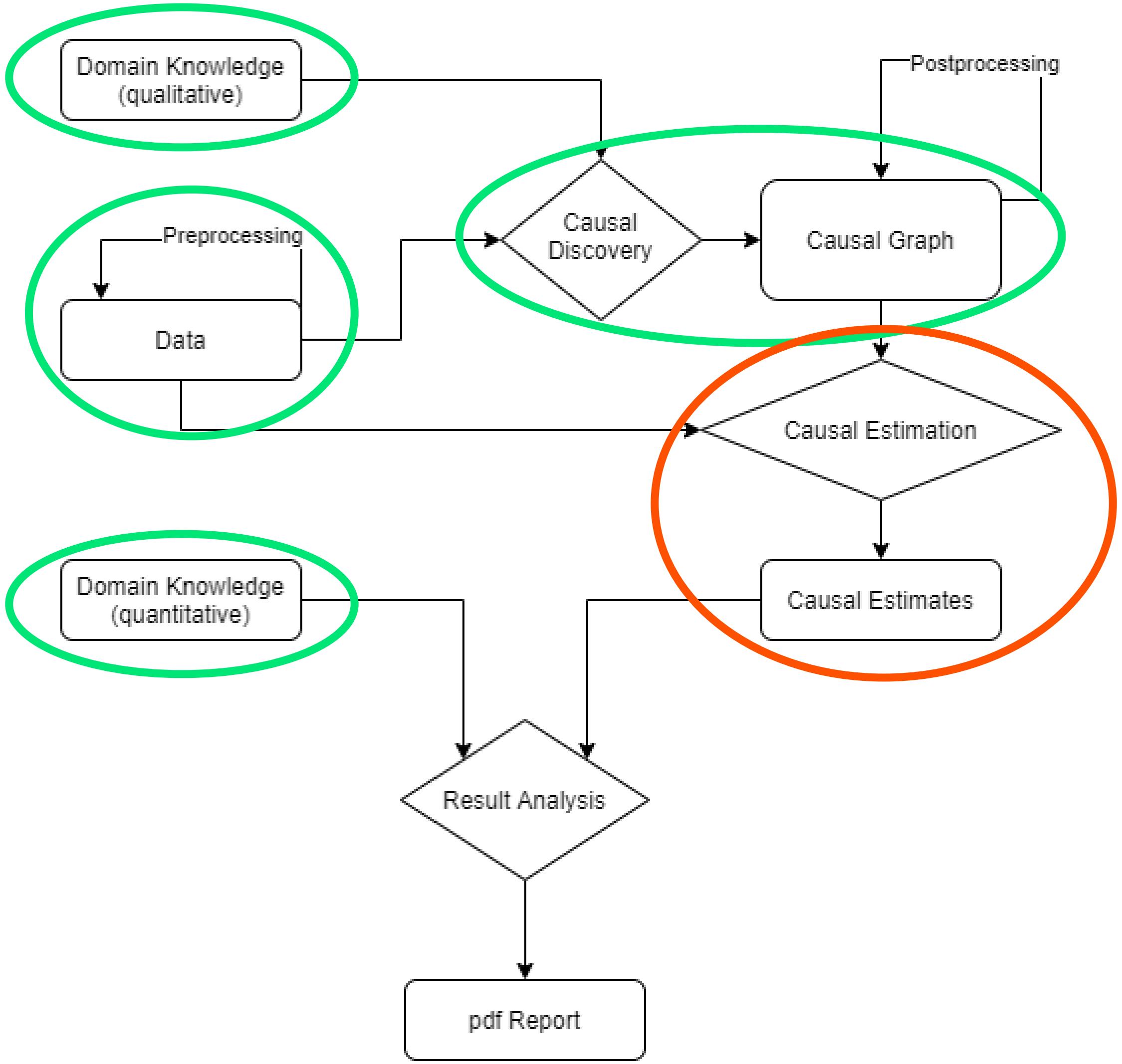
Daniel Grünbaum, Maike Stern, Elmar W. Lang

Causal Estimation

Where are we?

done

next



Daniel Grünbaum, Maike Stern, Elmar W. Lang

Quantitative Estimation

Getting quantitative causal effects

- estimate all possible causal effects with linear regression(s)

```
learner.run_all_quick_analyses()
```

Effect	Definition	Type of influence
Average treatment effect (ATE)	How does Y change if X is changed?	Overall
Natural direct effect (NDE)	How does Y change if X is changed and all other variables are fixed?	Direct
Natural indirect effect (NIE)	ATE - NDE	Indirect

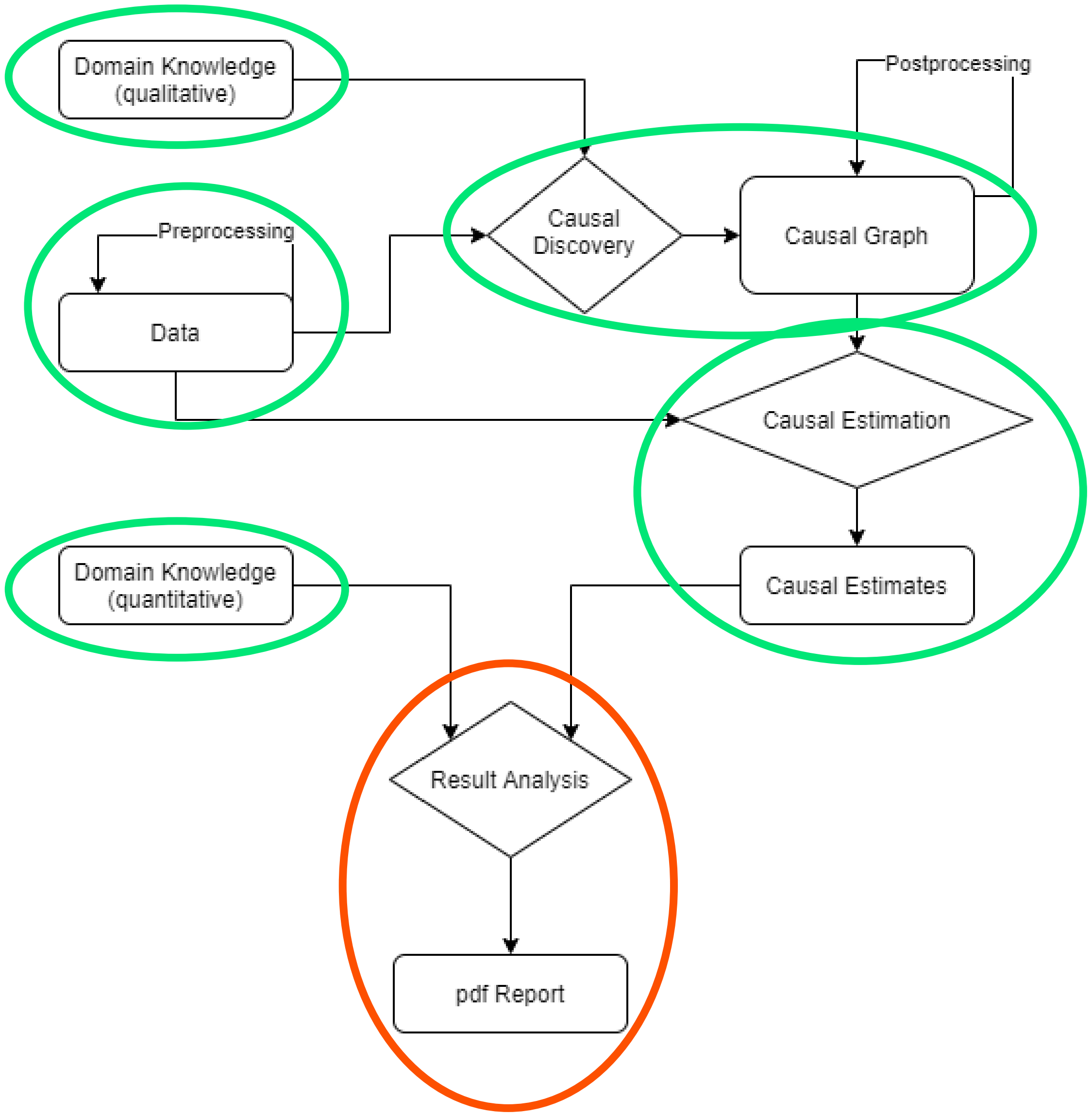
- write results to pdf report

Result Analysis

Where are we?

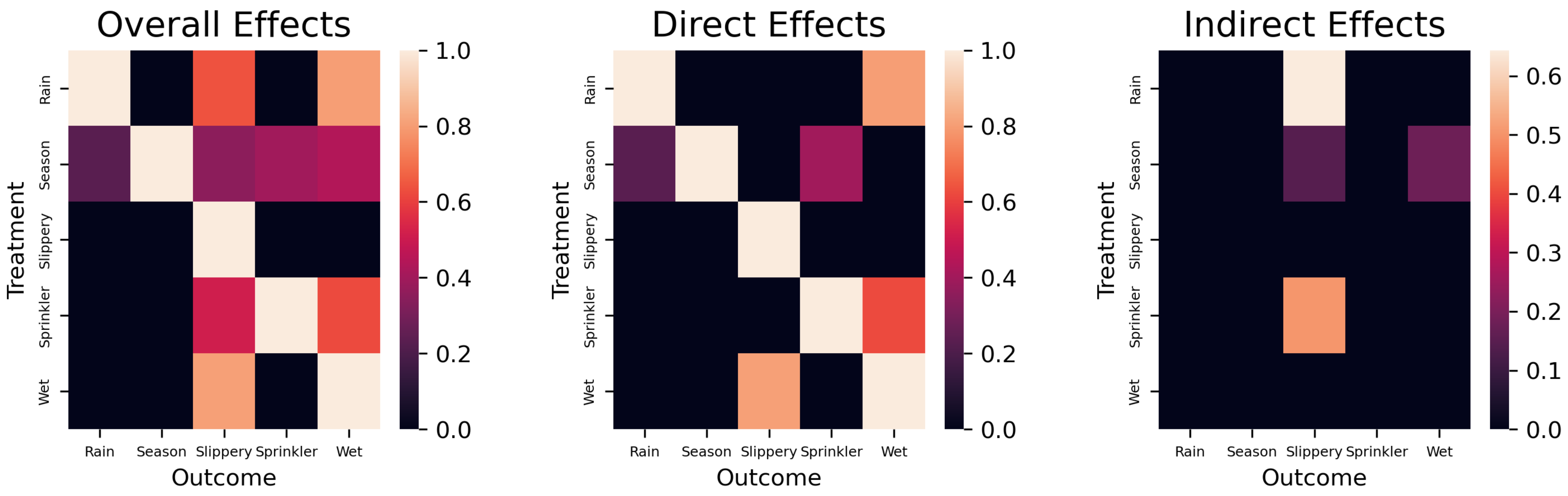
done

next



Result Analysis

Heatmaps



Daniel Grünbaum, Maike Stern, Elmar W. Lang

Result Analysis

Largest effects

10 Largest Overall Effects:

Treatment	Outcome	Estimated_effect
Wet	Slippery	0.81
Rain	Wet	0.80
Rain	Slippery	0.64
Sprinkler	Wet	0.62
Sprinkler	Slippery	0.52
Season	Wet	0.44
Season	Sprinkler	0.40
Season	Slippery	0.35
Season	Rain	0.23
Slippery	Rain	0.00

10 Largest Direct Effects:

Treatment	Outcome	Estimated_effect
Wet	Slippery	0.81
Rain	Wet	0.80
Sprinkler	Wet	0.62
Season	Sprinkler	0.40
Season	Rain	0.23
Slippery	Rain	0.00
Slippery	Season	0.00
Slippery	Wet	0.00
Slippery	Sprinkler	0.00
Rain	Slippery	0.00

Turning on the sprinkler does make my lawn slippery!

10 Largest Indirect Effects:

Treatment	Outcome	Estimated_effect
Rain	Slippery	0.64
Sprinkler	Slippery	0.50
Season	Wet	0.18
Season	Slippery	0.14
Slippery	Rain	0.00
Slippery	Season	0.00
Slippery	Wet	0.00
Slippery	Sprinkler	0.00
Rain	Season	0.00
Rain	Wet	0.00

Result Analysis

Full tables

Overall Effects

	Rain	Season	Slippery	Sprinkler	Wet
Rain	1.00	0.00	0.64	0.00	0.80
Season	0.23	1.00	0.35	0.40	0.44
Slippery	0.00	0.00	1.00	0.00	0.00
Sprinkler	0.00	0.00	0.52	1.00	0.62
Wet	0.00	0.00	0.81	0.00	1.00

Direct Effects

	Rain	Season	Slippery	Sprinkler	Wet
Rain	1.00	0.00	0.00	0.00	0.80
Season	0.23	1.00	0.00	0.40	0.00
Slippery	0.00	0.00	1.00	0.00	0.00
Sprinkler	0.00	0.00	0.00	1.00	0.62
Wet	0.00	0.00	0.81	0.00	1.00

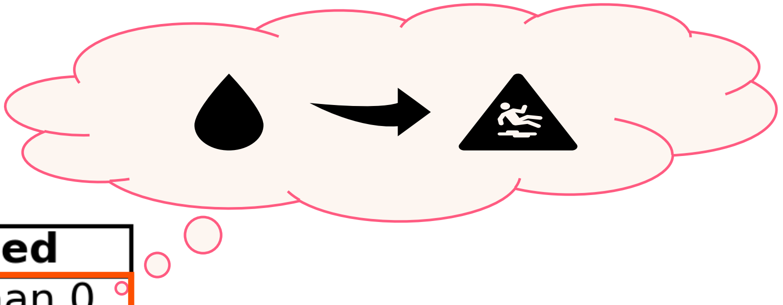
Turning on the sprinkler does make my lawn slippery!

Indirect Effects

	Rain	Season	Slippery	Sprinkler	Wet
Rain	0.00	0.00	0.64	0.00	0.00
Season	0.00	0.00	0.14	0.00	0.18
Slippery	0.00	0.00	0.00	0.00	0.00
Sprinkler	0.00	0.00	0.50	0.00	0.00
Wet	0.00	0.00	0.00	0.00	0.00

Result Analysis

Validation using quantitative domain knowledge



Passed validations (2/4):

Effect Type	Treatment	Outcome	Estimated	Expected
overall	Wet	Slippery	0.81	greater than 0
overall	Sprinkler	Wet	0.62	greater than 0

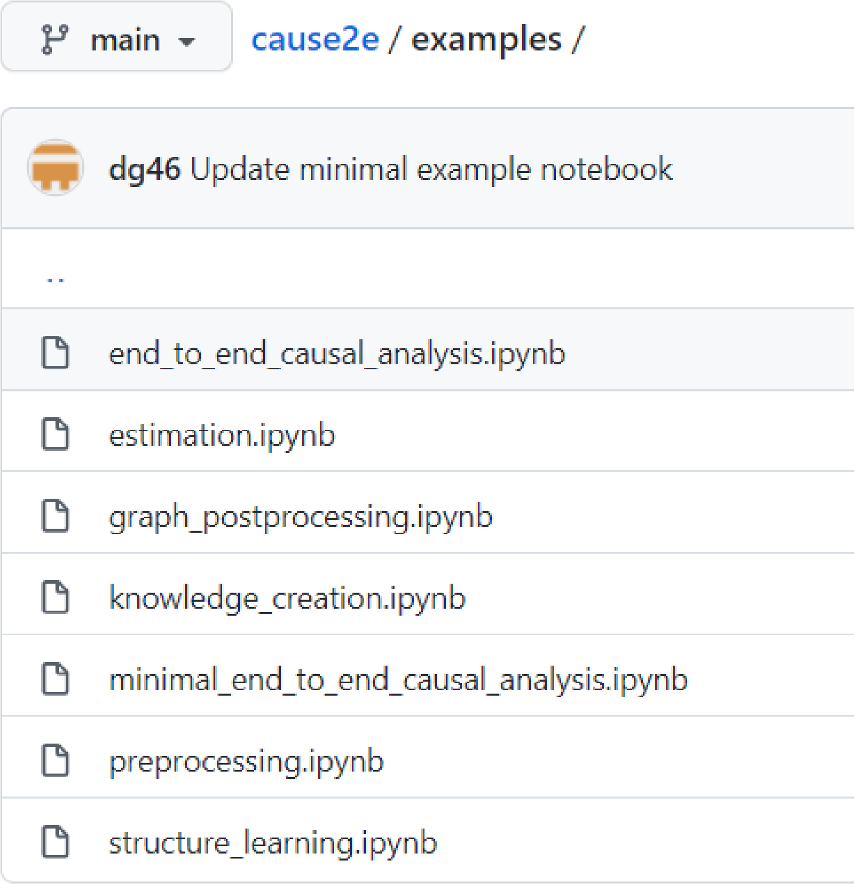
Failed validations (2/4):

Effect Type	Treatment	Outcome	Estimated	Expected
direct	Sprinkler	Rain	0.00	less than 0
indirect	Slippery	Season	0.00	between 0.2 and 0.4

Sources

Your turn!

Cause2e: https://github.com/MLResearchAtOSRAM/cause2e



Sources:

- Sprinkler example follows Judea Pearl‘s book **Causality**
- [1] **Py-causal** by Chirayu Wongchokprasitti et al.: https://github.com/bd2kccd/py-causal
- [2] **Causal Discovery Toolbox** by Divian Kalainathan and Olivier Goudet: https://github.com/FenTechSolutions/CausalDiscoveryToolbox
- [3] **DoWhy** by Amit Sharma and Emre Kiciman: https://github.com/microsoft/dowhy

Pointers

More causal inference

Causal Inference Working Group:

- Regular exchange for causality enthusiasts and curious newcomers
- About 20 members from academia and industry
- Wiki for schedule and more info: https://gitlab.com/causal-inference/working-group/-/wikis/home
- Feel free to join or drop by for just one meeting!

Contact me to learn or discuss about causal inference! daniel.gruenbaum@ams-osram.com

Interested in a **PhD in causal reinforcement learning** at ams OSRAM?
Contact us! maike.stern@ams-osram.com

Thank you! Questions?