Use of Prodrisk for Investment Analysis in Røldal-Suldal

Revenue Calculations and Describing Power System Operation

Torbjørn Ims Østby (Lyse) & Amund Skretting Bergset (SINTEF)

Lyse

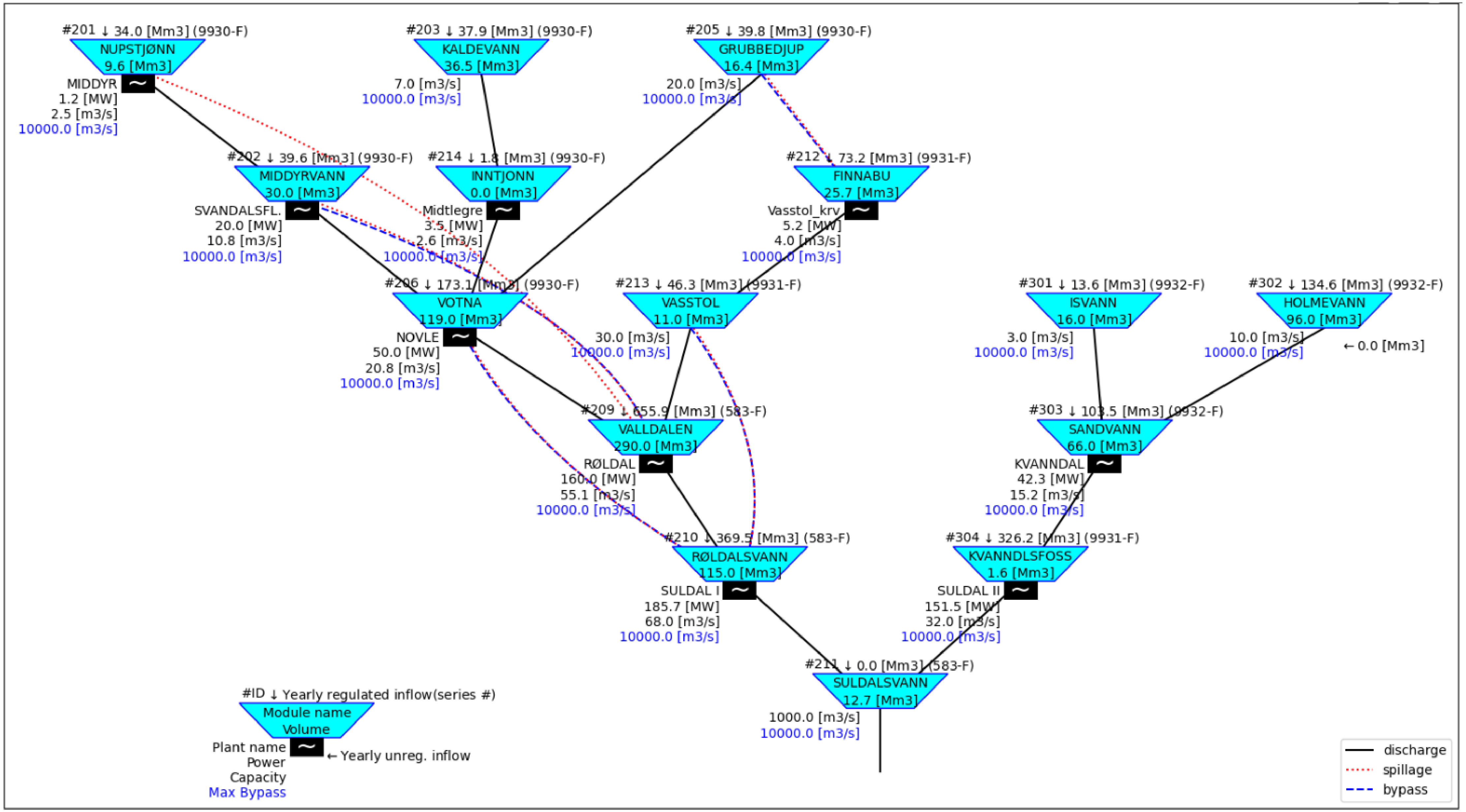
Investment analysis with Prodrisk

- Revenue
  - Which Power plants (topology) to build?
  - And how large?
  - What are the cost of todays environmental restrictions?
  - What is the cost of possible future environmental restrictions
  - Cost of «avbøtende tiltak»?
  - Implications for taxes and “konsesjonsavgift”
- Dispatch/Hydrology
  - How would water levels, river discharges etc. be in the new system + base case with climate change
- Risk analysis
  - System robustness

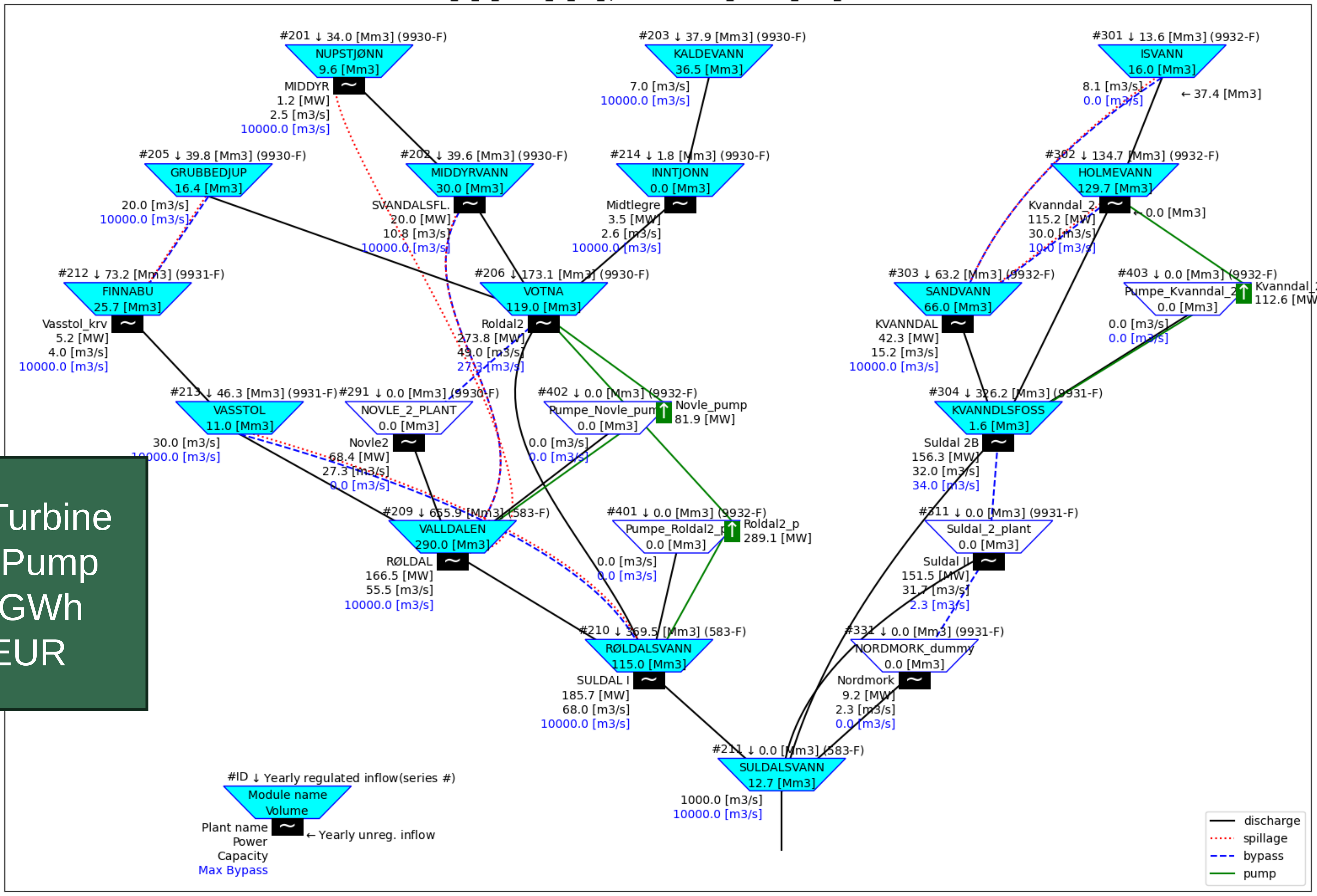
➢ Build a framework to effectively answer the questions

Lyse

Røldal Suldal Kraftsystem (RSK) Topology Today



Røldal Suldal Kraftsystem (RSK) Topology Applied for

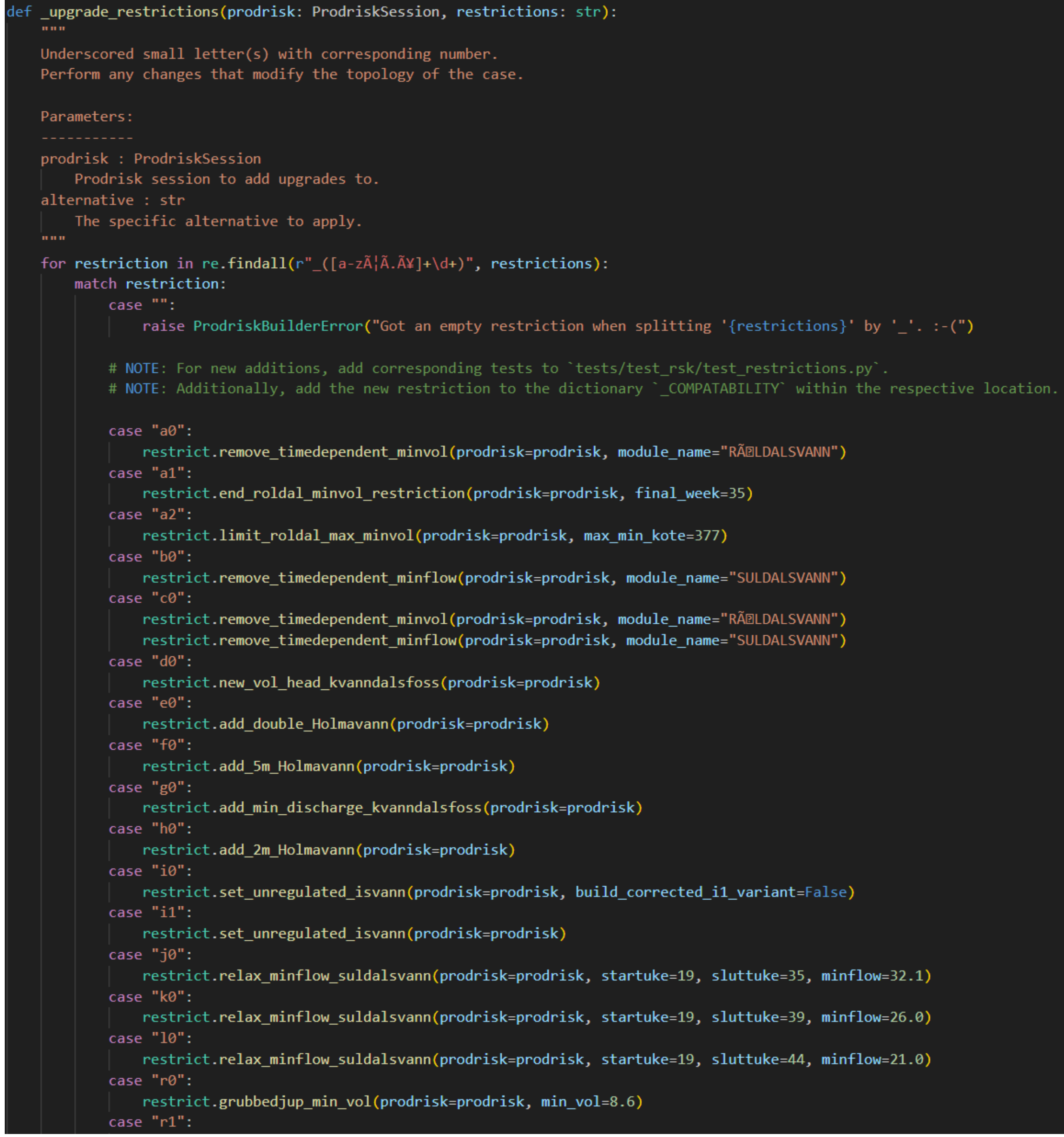


+ 650 MW Turbine
+ 490 MW Pump
+ 50 new GWh
~ 670 MEUR

Lyse

Workflow
Implementing a new case

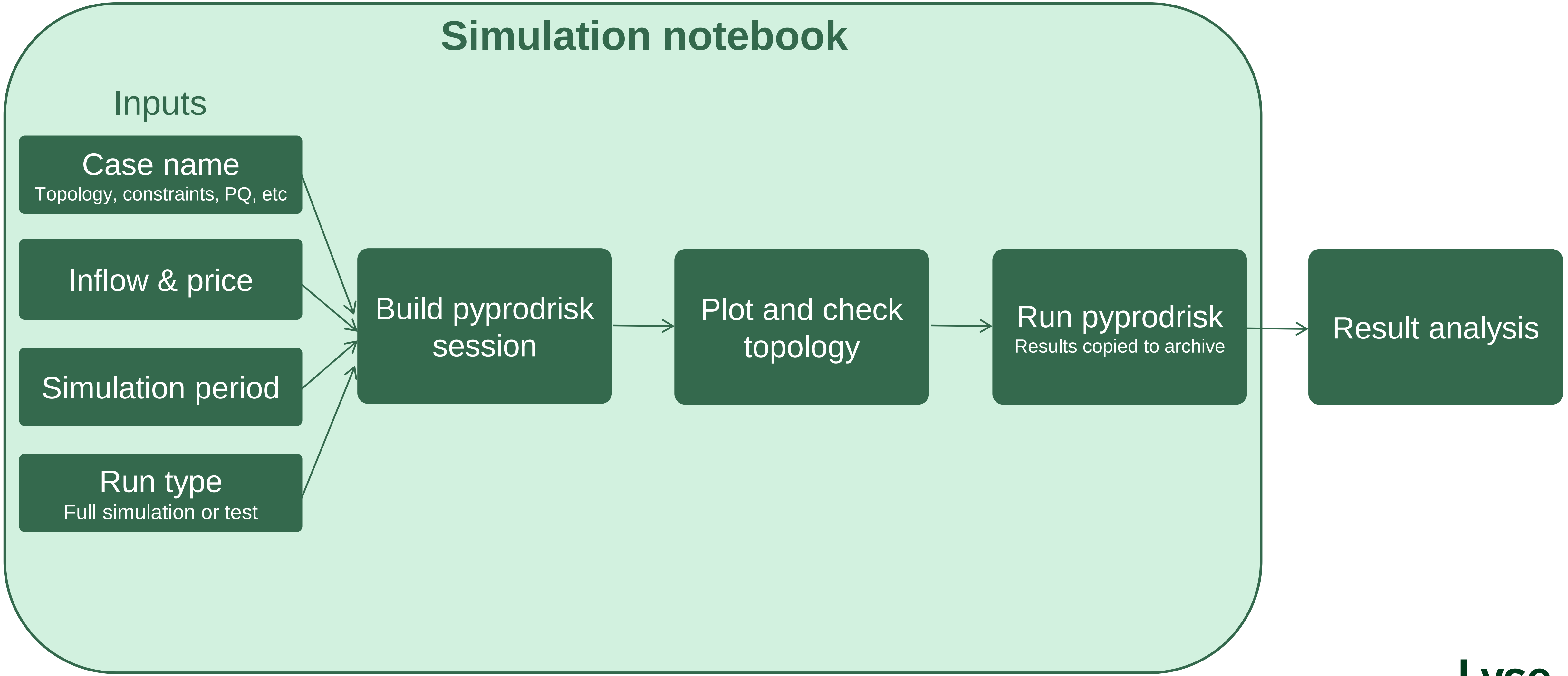
- Separate function for each modeling change
  - Topology/constraints/PQ curves etc
- Implement test for each function
  - All tests are automatically run when the code is edited
- Unique code in case name
  - e.g. "T8" or "w1"
- Case name = combination of codes
  - e.g. **T8_w5_i1_x1+U13_b0_q0_u2**



```python
def _upgrade_restrictions(prodrisk: ProdriskSession, restrictions: str):
    """
    Underscored small letter(s) with corresponding number.
    Perform any changes that modify the topology of the case.

    Parameters:
    -----------
    prodrisk : ProdriskSession
        Prodrisk session to add upgrades to.
    alternative : str
        The specific alternative to apply.
    """
    for restriction in re.findall(r"_([a-zÃ¦Ã.Ã¥]+\d+)", restrictions):
        match restriction:
            case "":
                raise ProdriskBuilderError("Got an empty restriction when splitting '{restrictions}' by '_'. :-(")

            # NOTE: For new additions, add corresponding tests to `tests/test_rsk/test_restrictions.py`.
            # NOTE: Additionally, add the new restriction to the dictionary `_COMPATABILITY` within the respective location.

            case "a0":
                restrict.remove_timedependent_minvol(prodrisk=prodrisk, module_name="RÃLDALSVANN")
            case "a1":
                restrict.end_roldal_minvol_restriction(prodrisk=prodrisk, final_week=35)
            case "a2":
                restrict.limit_roldal_max_minvol(prodrisk=prodrisk, max_min_kote=377)
            case "b0":
                restrict.remove_timedependent_minflow(prodrisk=prodrisk, module_name="SULDALSVANN")
            case "c0":
                restrict.remove_timedependent_minvol(prodrisk=prodrisk, module_name="RÃLDALSVANN")
                restrict.remove_timedependent_minflow(prodrisk=prodrisk, module_name="SULDALSVANN")
            case "d0":
                restrict.new_vol_head_kvanndalsfoss(prodrisk=prodrisk)
            case "e0":
                restrict.add_double_Holmavann(prodrisk=prodrisk)
            case "f0":
                restrict.add_5m_Holmavann(prodrisk=prodrisk)
            case "g0":
                restrict.add_min_discharge_kvanndalsfoss(prodrisk=prodrisk)
            case "h0":
                restrict.add_2m_Holmavann(prodrisk=prodrisk)
            case "i0":
                restrict.set_unregulated_isvann(prodrisk=prodrisk, build_corrected_i1_variant=False)
            case "i1":
                restrict.set_unregulated_isvann(prodrisk=prodrisk)
            case "j0":
                restrict.relax_minflow_suldalsvann(prodrisk=prodrisk, startuke=19, sluttuke=35, minflow=32.1)
            case "k0":
                restrict.relax_minflow_suldalsvann(prodrisk=prodrisk, startuke=19, sluttuke=39, minflow=26.0)
            case "l0":
                restrict.relax_minflow_suldalsvann(prodrisk=prodrisk, startuke=19, sluttuke=44, minflow=21.0)
            case "r0":
                restrict.grubbedjup_min_vol(prodrisk=prodrisk, min_vol=8.6)
            case "r1":
```

Workflow

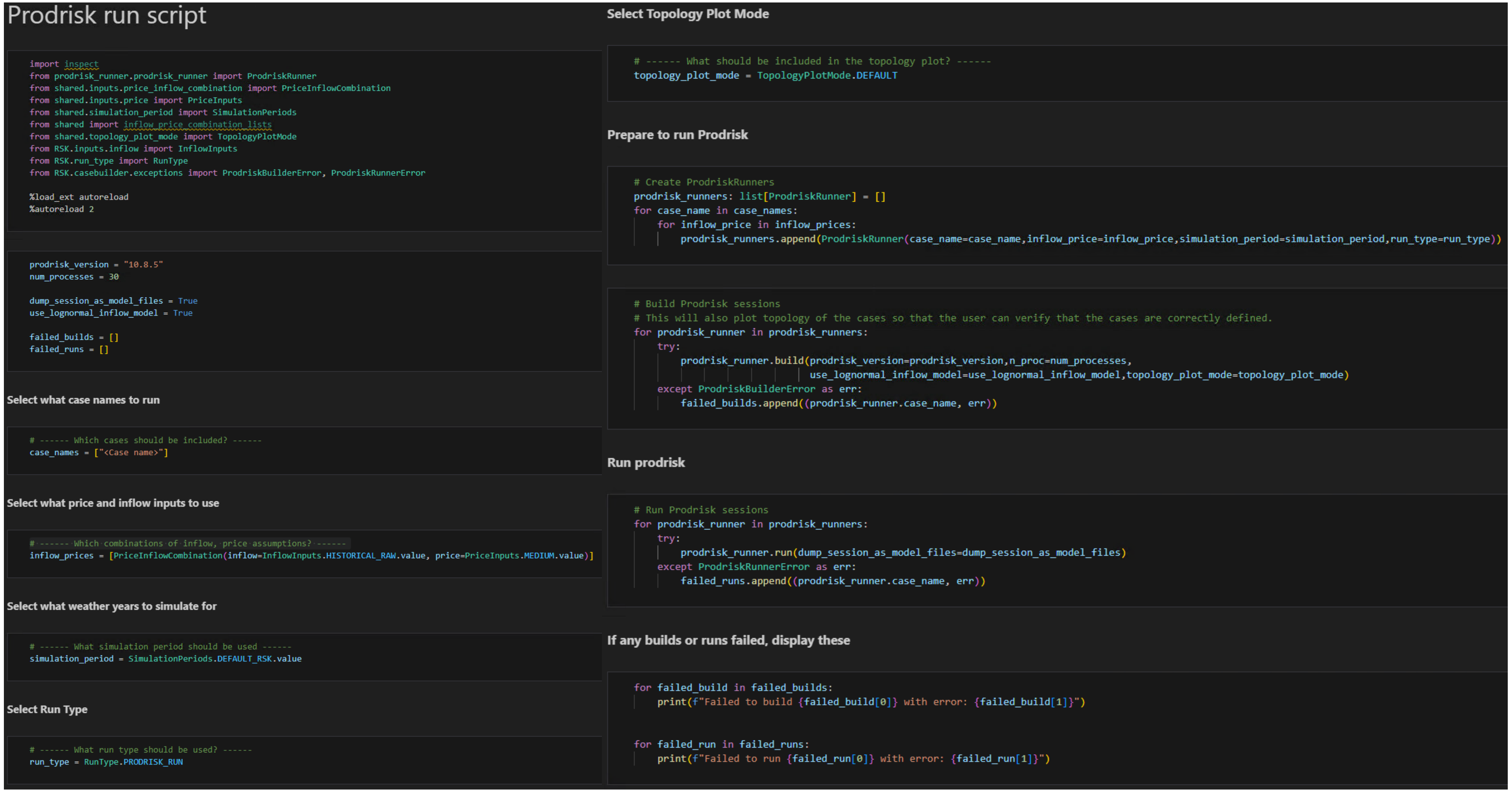
Running a simulation



Lyse

Workflow

Running a simulation



Prodrisk run script

```
import inspect
from prodrisk_runner.prodrisk_runner import ProdriskRunner
from shared.inputs.price_inflow_combination import PriceInflowCombination
from shared.inputs.price import PriceInputs
from shared.simulation_period import SimulationPeriods
from shared import inflow_price_combination_lists
from shared.topology_plot_mode import TopologyPlotMode
from RSK.inputs.inflow import InflowInputs
from RSK.run_type import RunType
from RSK.casebuilder.exceptions import ProdriskBuilderError, ProdriskRunnerError

%load_ext autoreload
%autoreload 2
```

```
prodrisk_version = "10.8.5"
num_processes = 30

dump_session_as_model_files = True
use_lognormal_inflow_model = True

failed_builds = []
failed_runs = []
```

Select what case names to run

```
# ------ Which cases should be included? ------
case_names = ["<Case name>"]
```

Select what price and inflow inputs to use

```
# ------ Which combinations of inflow, price assumptions? ------
inflow_prices = [PriceInflowCombination(inflow=InflowInputs.HISTORICAL_RAW.value, price=PriceInputs.MEDIUM.value)]
```

Select what weather years to simulate for

```
# ------ What simulation period should be used ------
simulation_period = SimulationPeriods.DEFAULT_RSK.value
```

Select Run Type

```
# ------ What run type should be used? ------
run_type = RunType.PRODRISK_RUN
```

Select Topology Plot Mode

```
# ------ What should be included in the topology plot? ------
topology_plot_mode = TopologyPlotMode.DEFAULT
```

Prepare to run Prodrisk

```
# Create ProdriskRunners
prodrisk_runners: list[ProdriskRunner] = []
for case_name in case_names:
    for inflow_price in inflow_prices:
        prodrisk_runners.append(ProdriskRunner(case_name=case_name,inflow_price=inflow_price,simulation_period=simulation_period,run_type=run_type))
```

```
# Build Prodrisk sessions
# This will also plot topology of the cases so that the user can verify that the cases are correctly defined.
for prodrisk_runner in prodrisk_runners:
    try:
        prodrisk_runner.build(prodrisk_version=prodrisk_version,n_proc=num_processes,
                              use_lognormal_inflow_model=use_lognormal_inflow_model,topology_plot_mode=topology_plot_mode)
    except ProdriskBuilderError as err:
        failed_builds.append((prodrisk_runner.case_name, err))
```

Run prodrisk

```
# Run Prodrisk sessions
for prodrisk_runner in prodrisk_runners:
    try:
        prodrisk_runner.run(dump_session_as_model_files=dump_session_as_model_files)
    except ProdriskRunnerError as err:
        failed_runs.append((prodrisk_runner.case_name, err))
```

If any builds or runs failed, display these

```
for failed_build in failed_builds:
    print(f"Failed to build {failed_build[0]} with error: {failed_build[1]}")


for failed_run in failed_runs:
    print(f"Failed to run {failed_run[0]} with error: {failed_run[1]}")
```

Lyse

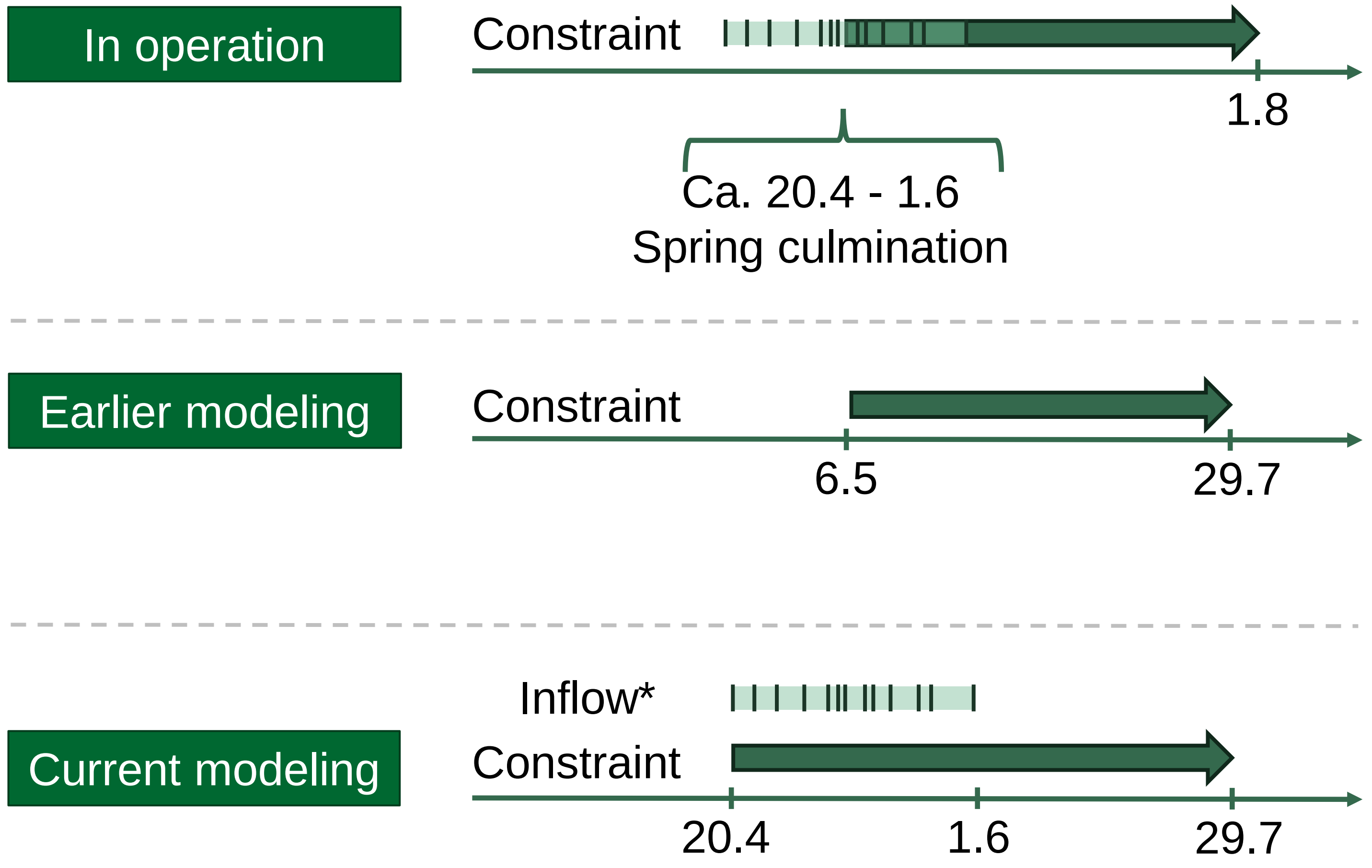
Modeling challenges

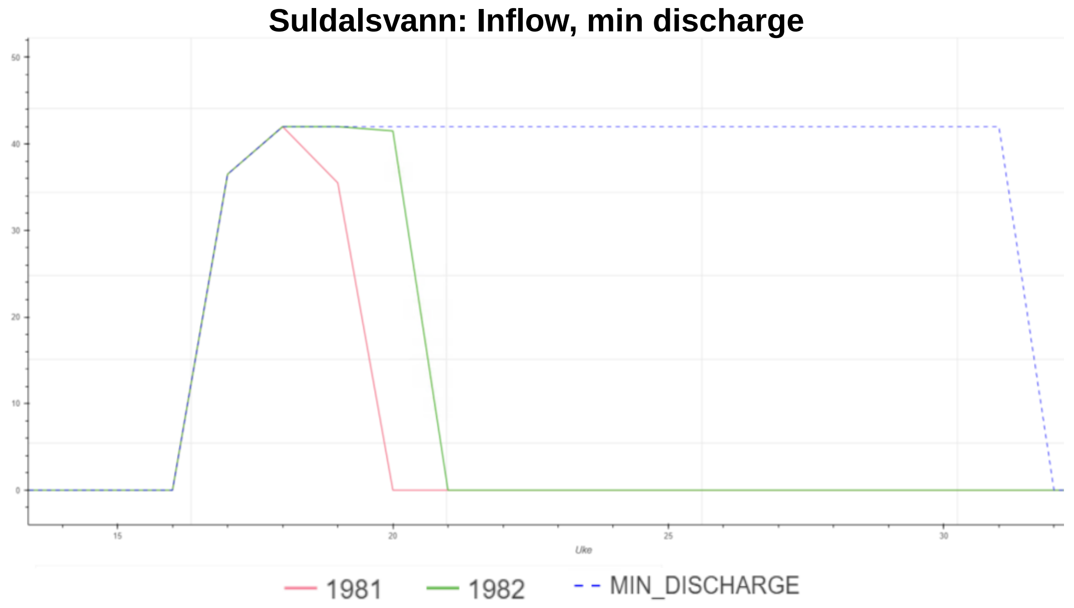
- Constraints starting at spring culmination → ***scenario-dependent***
  - Minimum discharge in Suldalsvann
  - Minimum reservoir in Røldalsvann

- Not possible with the SDDP methodology in Prodrisk

Modeling challenges

Minimum discharge constraint in Suldalsvann





Difference in simulated income from *earlier* to *current* modeling: **- 0.05%**

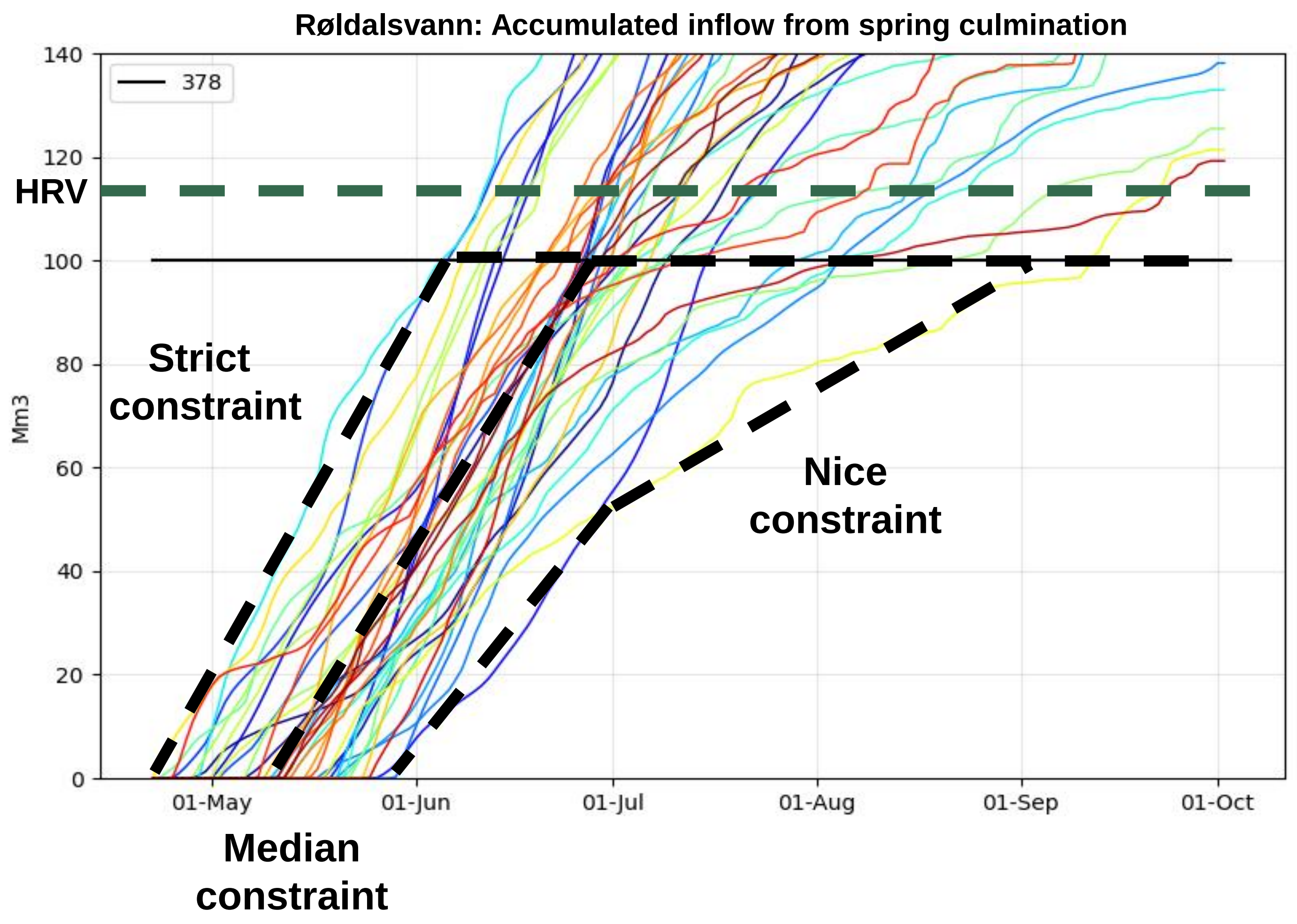
* Additional inflow to Suldalsvann that cancels out the constraint

Modeling challenges

Minimum reservoir constraint in Røldalsvann

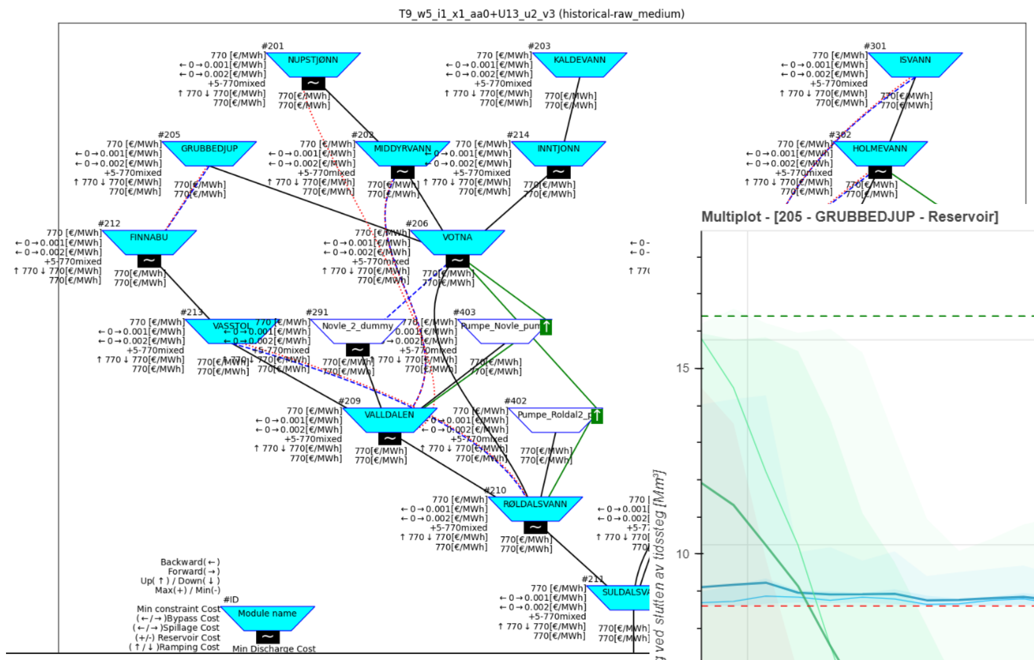
- Follows accumulated inflow from spring culmination until 100 Mm3

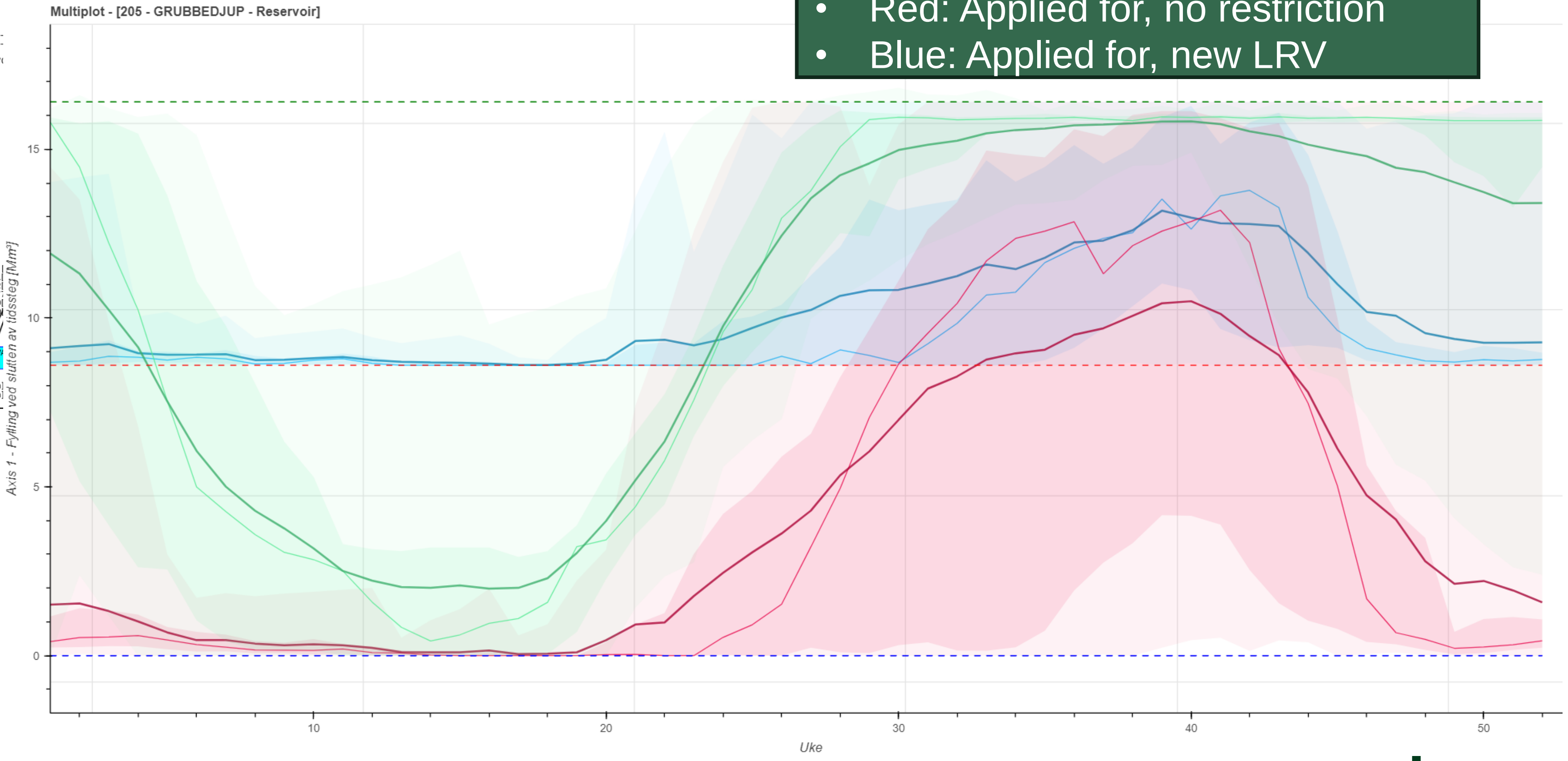
- Compared against simulation with
  - *Nice* constraint in strategy
  - *Scenario-dependent* constraint in final simulation
- Similar simulated income as with *median* constraint



Lyse

Visualization of results, model configuration and input



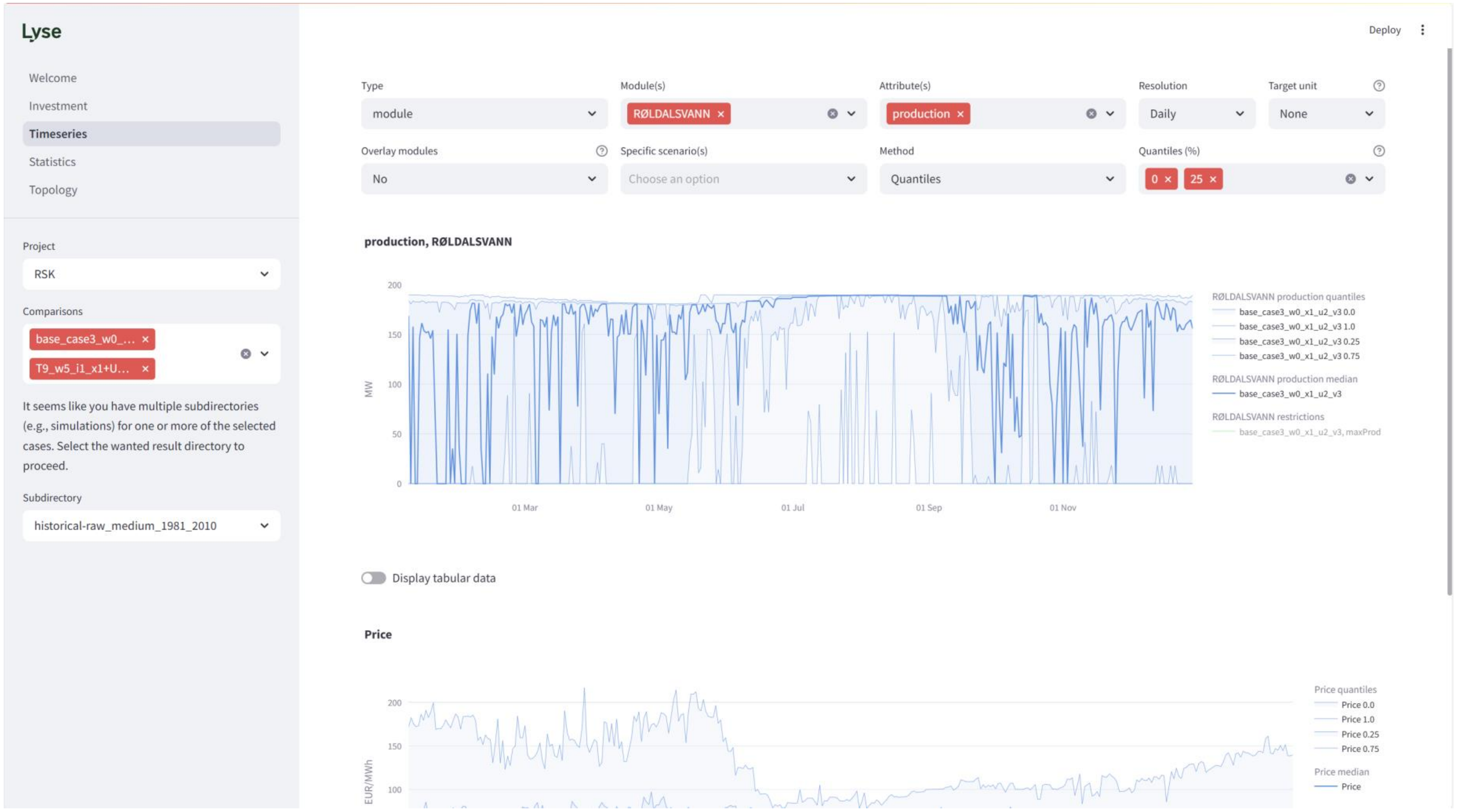


Water level GRUBBEDJUP Quantiles
- Green: Historical
- Red: Applied for, no restriction
- Blue: Applied for, new LRV

Lyse

AD: Dashboard webapp

Nice streamlit where you can plot and compare «everything»
easy to use – even for biologist and economists



Project: RSK

Comparisons: base_case3_w0_... ×, T9_w5_i1_x1+U... ×

It seems like you have multiple subdirectories (e.g., simulations) for one or more of the selected cases. Select the wanted result directory to proceed.

Subdirectory: historical-raw_medium_1981_2010

Type: module

Module(s): RØLDALSVANN ×

Attribute(s): production ×

Resolution: Daily

Target unit: None

Overlay modules: No

Specific scenario(s): Choose an option

Method: Quantiles

Quantiles (%): 0 ×, 25 ×

production, RØLDALSVANN

MW
200
150
100
50
0
01 Mar
01 May
01 Jul
01 Sep
01 Nov
RØLDALSVANN production quantiles
base_case3_w0_x1_u2_v3 0.0
base_case3_w0_x1_u2_v3 1.0
base_case3_w0_x1_u2_v3 0.25
base_case3_w0_x1_u2_v3 0.75
RØLDALSVANN production median
base_case3_w0_x1_u2_v3
RØLDALSVANN restrictions
base_case3_w0_x1_u2_v3, maxProd

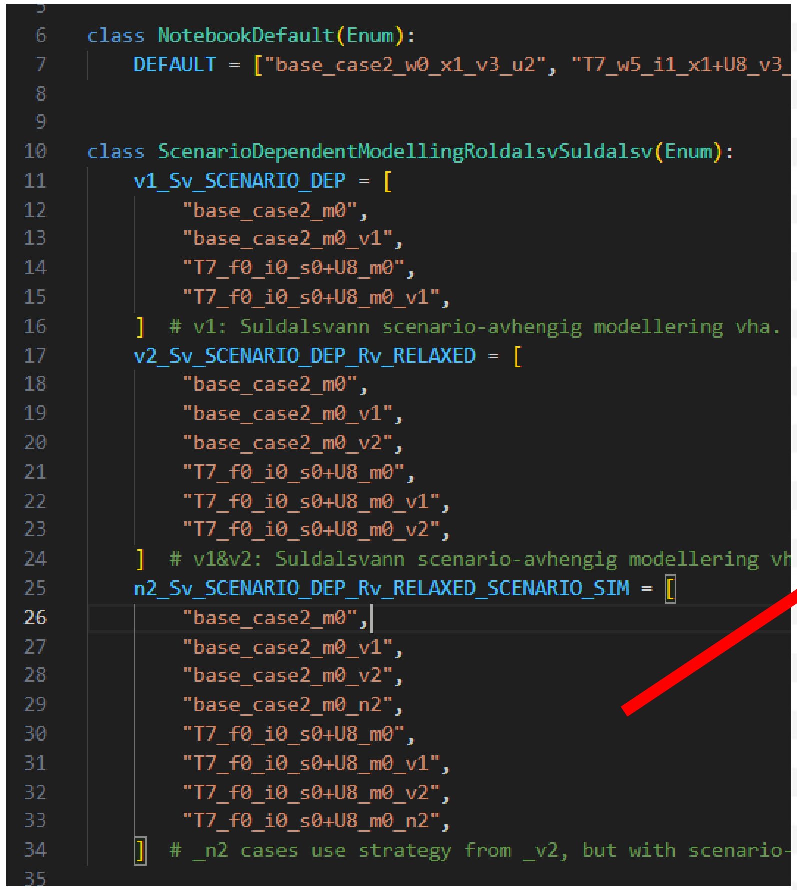
Display tabular data

Price

Lyse

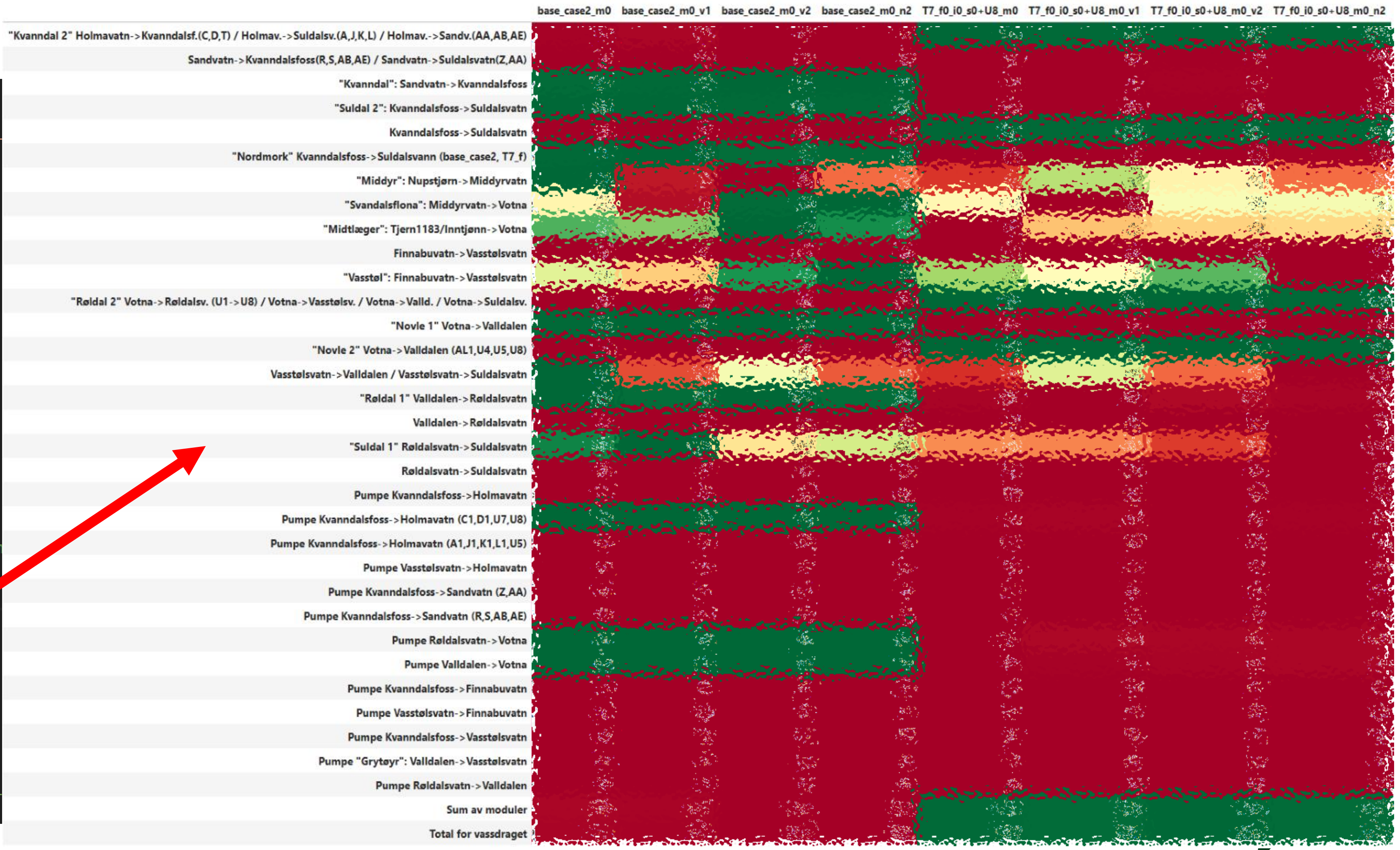
Impose a change and evaluate the results

Case comparison list



```
class NotebookDefault(Enum):
    DEFAULT = ["base_case2_w0_x1_v3_u2", "T7_w5_i1_x1+U8_v3_


class ScenarioDependentModellingRoldalsvSuldalsv(Enum):
    v1_Sv_SCENARIO_DEP = [
        "base_case2_m0",
        "base_case2_m0_v1",
        "T7_f0_i0_s0+U8_m0",
        "T7_f0_i0_s0+U8_m0_v1",
    ]  # v1: Suldalsvann scenario-avhengig modellering vha.
    v2_Sv_SCENARIO_DEP_Rv_RELAXED = [
        "base_case2_m0",
        "base_case2_m0_v1",
        "base_case2_m0_v2",
        "T7_f0_i0_s0+U8_m0",
        "T7_f0_i0_s0+U8_m0_v1",
        "T7_f0_i0_s0+U8_m0_v2",
    ]  # v1&v2: Suldalsvann scenario-avhengig modellering vh
    n2_Sv_SCENARIO_DEP_Rv_RELAXED_SCENARIO_SIM = [
        "base_case2_m0",
        "base_case2_m0_v1",
        "base_case2_m0_v2",
        "base_case2_m0_n2",
        "T7_f0_i0_s0+U8_m0",
        "T7_f0_i0_s0+U8_m0_v1",
        "T7_f0_i0_s0+U8_m0_v2",
        "T7_f0_i0_s0+U8_m0_n2",
    ]  # _n2 cases use strategy from _v2, but with scenario-
```

```
Creating heatmap for ('inflow', 'price', 'result type'): ('historical-raw', 'medium', 'income + vol change [MEUR]'):
```



Take home message

When investing 7-8 mrd. NOK you want to be sure that you have minimized uncertainties with the Prodrisk domain.

⇒ We have built a framework for modelling:

- ⇒ Building cases and inputs
- ⇒ Visualization
- ⇒ income calculations

Lyse