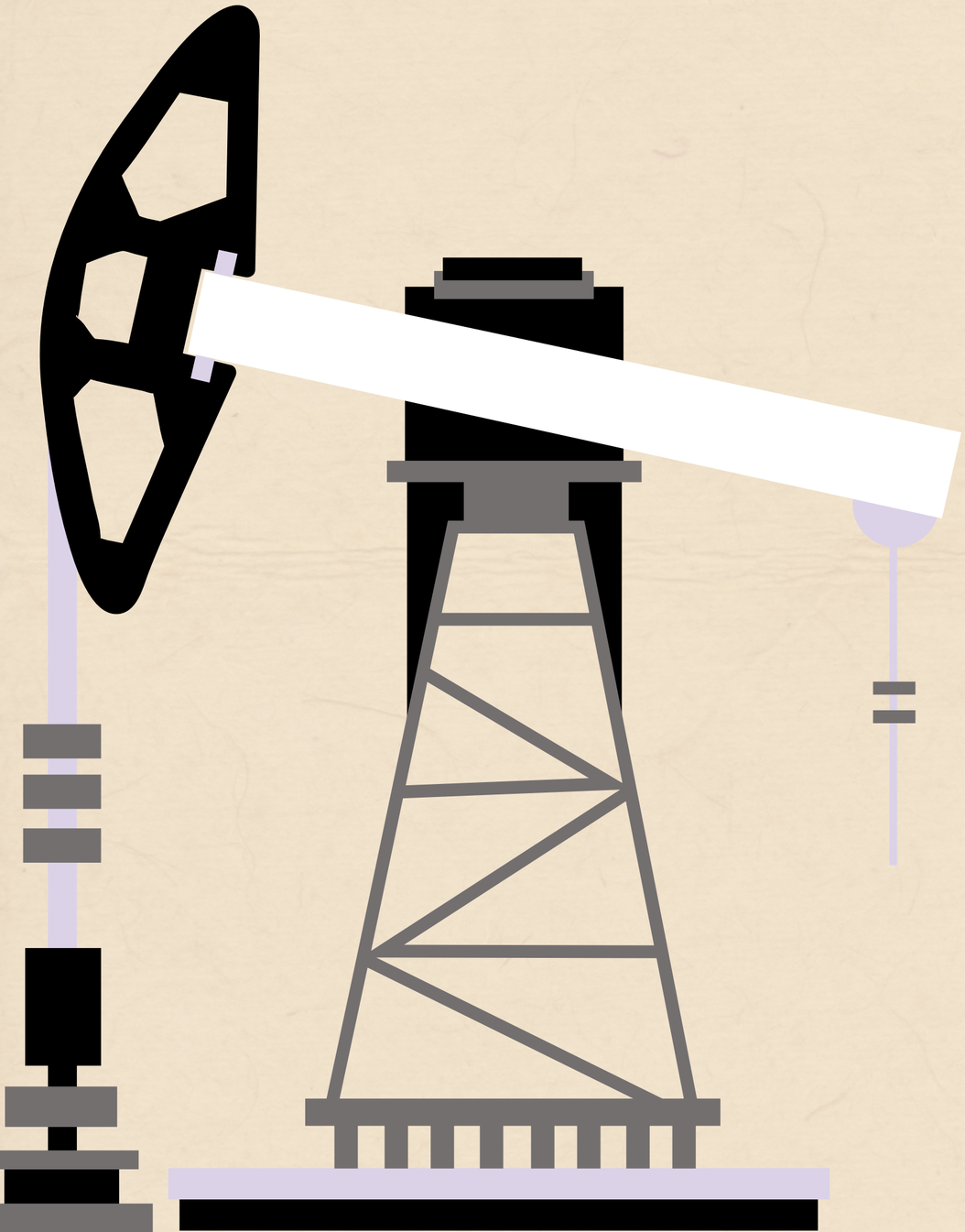


ASSIGNMENT 3
" Oil Production Optimization With Gurobi "

Group member

Pattarapon	Buathong	62070504012
Suchada	Panyawiraphat	62070504016
Pawornrat	Supavit Pranayanuntana	62070504030

Problem

Assume that you are working in a Thai's well-known crude oil company. They ask you to develop a five-year operating plan for the gulf of Thailand which includes four offshore oil rigs in it. **They are limited to operating a maximum of three rigs in this gulf each year**. Even if a rig does not operate in a given year, the company is still required to pay royalties on it if there is a reasonable expectation that it will operate in the future. Otherwise, it may be permanently closed with no further royalties due. The following table summarizes the annual royalties due on each active rig (whether operating or not):

Rig IDs	Royalties (Baht)
Rig A	13,085,600
Rig B	16,357,000
Rig C	13,085,600
Rig D	16,357,000

Each Rig is limited to drilling a certain amount of crude oil per year and the quality of it (measured in API gravity) is different. These restrictions include the following:

Rig IDs	Max Production (barrel)	API Gravity
Rig A	1.9 Million	21
Rig B	1.5 Million	30
Rig C	1.0 Million	45
Rig D	2.3 Million	15

Problem

Each year, the crude oil drill from each operating rate must be combined to produce a specified grade of crude oil. The combined crude oil's annual requirements are as follows:

Year	Expected API Gravity
1	27
2	24
3	36
4	18
5	30

The combined crude oil is sold at a price 75 baht per barrel.

Due to the financial economic crisis, the company's revenue from crude **oil sales is forecast to be discounted at a rate of 5% per annum. The cost is assumed to be increased at the rate of 2% per annum due to inflation.**

You are assigned to give a plan according to the requirements that which rigs should be operated annually. **How much crude oil should be drilled from each rig? How much the expected revenue gain from your plan?:**

Linear Programming Model

Decision Variables

$$x_i = Oil\ Rig\ Usage\ ;\ 0 = Not\ being\ used,\ 1 = being\ used$$

$$y_i = Barrels\ Produced$$

Objective Function

$$Maximize:\ x_1y_1 + x_2y_2 + x_3y_3 + x_4y_4$$

Constraints

$$x_1 + x_2 + x_3 + x_4 \leq 3$$

$$x_1y_1 \leq 1,900,000$$

$$x_2y_2 \leq 1,500,000$$

$$x_3y_3 \leq 1,000,000$$

$$x_4y_4 \leq 2,300,000$$

$$\frac{(21)x_1y_1 + (30)x_2y_2 + (45)x_3y_3 + (15)x_4y_4}{\Sigma x_iy_i} \leq 27$$

$$\frac{(21)x_1y_1 + (30)x_2y_2 + (45)x_3y_3 + (15)x_4y_4}{\Sigma x_iy_i} \leq 24$$

$$\frac{(21)x_1y_1 + (30)x_2y_2 + (45)x_3y_3 + (15)x_4y_4}{\Sigma x_iy_i} \leq 36$$

$$\frac{(21)x_1y_1 + (30)x_2y_2 + (45)x_3y_3 + (15)x_4y_4}{\Sigma x_iy_i} \leq 18$$

$$\frac{(21)x_1y_1 + (30)x_2y_2 + (45)x_3y_3 + (15)x_4y_4}{\Sigma x_iy_i} \leq 30$$

Raw Revenue

The raw revenue is calculated by multiplying the barrels of oil produced in that year by the price of the oil per barrel (75 Baht).

$$Raw\ Revenue\ =\ (Oil\ barrels\ produced)(75)$$

Adjusted Revenue

The adjusted revenue is calculated by

$$Adjusted\ Revenue\ =\ [(Raw\ Revenue)(1\ -\ 0.05)^{year-1}]\ -\ (Royalties)(1\ +\ 0.02)^{year-1}$$

Optimization Result using Gurobi

Best Objective

```
Solution count 5: 1.99417e+07 1.935e+07 1.90167e+07 ... -0

Optimal solution found (tolerance 1.00e-04)
Best objective 1.994166666667e+07, best bound 1.994166666667e+07, gap 0.0000%
Total produced = 19941666.666666664 barrels
```

It means that the optimized **summation of five years production is equal to" roughly 1994166.67 barrels."**

Oil Production

In the first year, the expected **API gravity is 27**. The total oil production in this year is **4,375,000 barrels.**

Crude Oil Production[2015/01/01,RigA]	0
Crude Oil Production[2015/01/01,RigB]	1500000
Crude Oil Production[2015/01/01,RigC]	1000000
Crude Oil Production[2015/01/01,RigD]	1875000

In the second year, the expected **API gravity is 24**. The total oil production in this year is **4,600,000 barrels.**

Crude Oil Production[2016/01/01,RigA]	1900000
Crude Oil Production[2016/01/01,RigB]	0
Crude Oil Production[2016/01/01,RigC]	1000000
Crude Oil Production[2016/01/01,RigD]	1700000

Crude Oil Production[2017/01/01,RigA]	0
Crude Oil Production[2017/01/01,RigB]	1500000
Crude Oil Production[2017/01/01,RigC]	1000000
Crude Oil Production[2017/01/01,RigD]	0

In the third year, the expected **API gravity is 36**. The total oil production in this year is **2,500,000 barrels.**

Crude Oil Production[2018/01/01,RigA]	1900000
Crude Oil Production[2018/01/01,RigB]	100000
Crude Oil Production[2018/01/01,RigC]	0
Crude Oil Production[2018/01/01,RigD]	2300000

In the fourth year, the expected **API gravity is 18**. The total oil production in this year is **4,300,000 barrels.**

Oil Production

Crude Oil Production[2019/01/01,RigA]	1666667
Crude Oil Production[2019/01/01,RigB]	1500000
Crude Oil Production[2019/01/01,RigC]	1000000
Crude Oil Production[2019/01/01,RigD]	0

In the fifth year, the expected **API gravity is 30**. The total oil production in this year is **4,166,666.67 barrels.**

Royalties, Raw Revenue, Pre-adjusted Revenue, and Adjusted Revenue in DataFrame.

	Date	Royalties	Raw Rev	Pre adj Rev	Adjusted Rev
0	2015/01/01	45799600	328125000.0	3.281250e+08	2.823254e+08
1	2016/01/01	43378764	345000000.0	3.277500e+08	2.843712e+08
2	2017/01/01	30632080	187500000.0	1.692188e+08	1.385867e+08
3	2018/01/01	48602901	322500000.0	2.765034e+08	2.279005e+08
4	2019/01/01	46033890	312500000.0	2.545332e+08	2.084993e+08

Raw Revenue

The raw revenue is calculated by multiplying the barrels of oil produced in that year by the price of the oil per barrel (75 Baht).

$$Raw\ Revenue\ =\ (Oil\ barrels\ produced)(75)$$

First year: 328,125,000 Baht
Second year: 345,000,000 Baht
Third year: 187,500,000 Baht
Fourth year: 322,500,000 Baht
Fifth year: 312,500,000 Baht

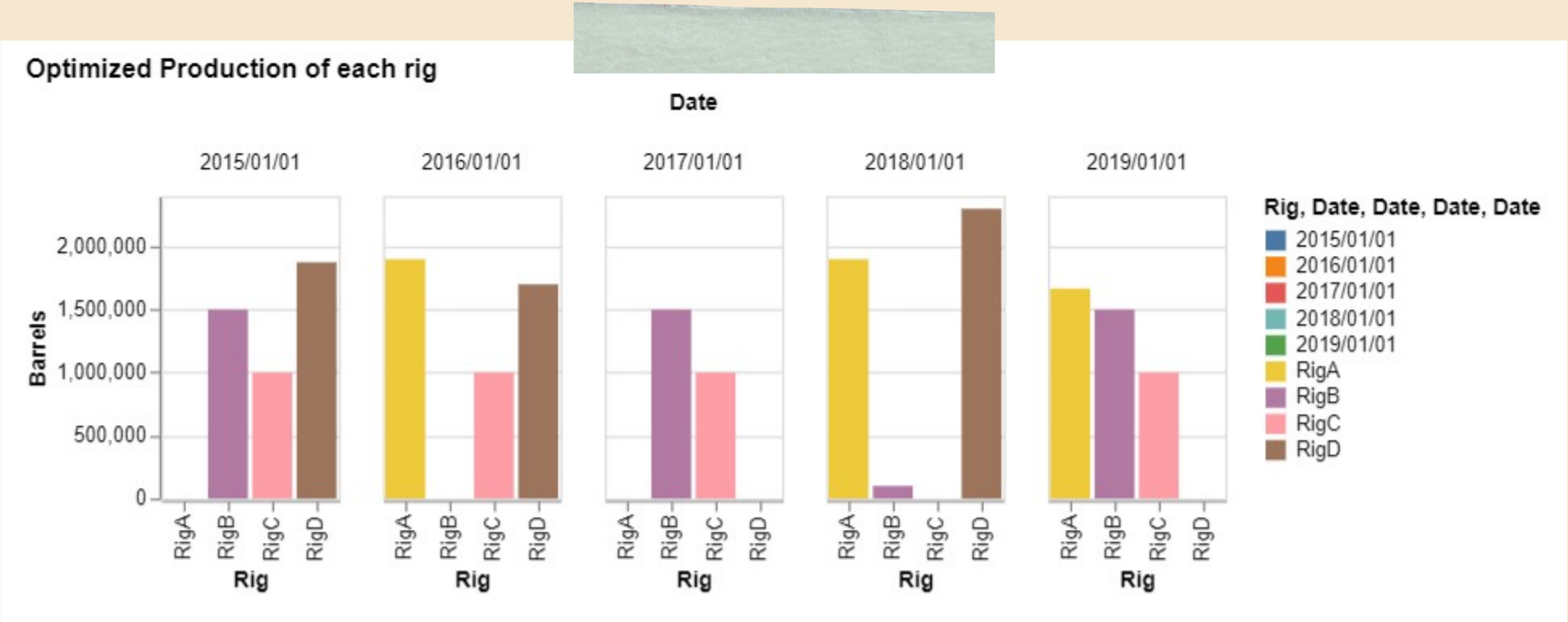
Adjusted Revenue

The adjusted revenue is calculated by

$$Adjusted\ Revenue\ =\ [(Raw\ Revenue)(1\ -\ 0.05)^{year-1}]\ -\ (Royalties)(1\ +\ 0.02)^{year-1}$$

Year 1: 282,325,400 Baht
Year 2: 284,371,200 Baht
Year 3: 138,586,700 Baht
Year 4: 227,900,500 Baht
Year 5: 208,499,300 Baht

Datapane Graphs



The first graph **shows the production planning for each rig every year**, if we follow this plan, we will produce the maximum oil barrel according to the optimizer by Gurobi. Analyzing the graph, it is guaranteed we could reduce the royalties cost by shutting down RigD in year 4 as it is not used in year 5 (the last year) according to the optimized plan.

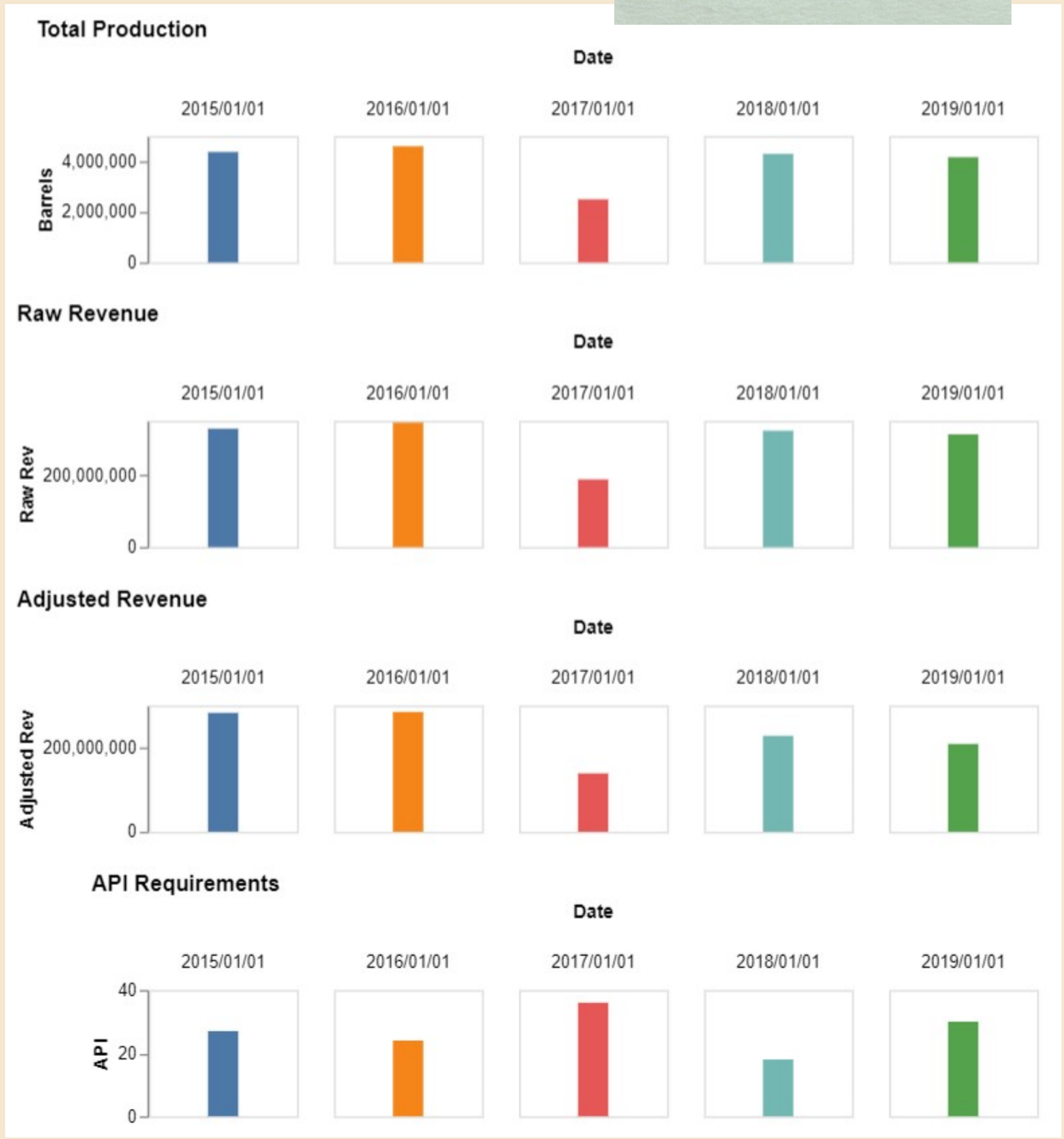
Datapane Graphs

The second graph shows **total oil production for each year.**

The third graph shows **raw revenue for each year.**

The fourth graph shows **the adjusted revenue calculated by adjusting the raw revenue to the inflation (5%) and subtracting the cost (royalties) that rise by 2% every year.**

The fifth graph shows **the total API Gravity of the oil barrels each year.**



Code

```
import gurobipy
import numpy as np
import pandas as pd
import altair as alt
import datapane as dp
from typing import List, Dict
def optimize_planning(timeline: List[str],
                      rigs: List[str],
                      yearly_requirements: Dict[str, int],
                      production_per_rig: Dict[str, int],
                      API_cap: Dict[str, int]) -> pd.DataFrame:

    model = gurobipy.Model("Optimize production planning")
```

```
# DEFINE VARIABLES
line_opening = model.addVars(
    timeline, rigs, vtype=gurobipy.GRB.BINARY, name="Use status"
)
```

$$x_1 + x_2 + x_3 + x_4 \leq 3$$

```
oil_produced = model.addVars(
    timeline,
    rigs,
    vtype=gurobipy.GRB.CONTINUOUS,
    name="Crude Oil Production",
)

sum_of_oil_produced = model.addVars(
    timeline,
    vtype=gurobipy.GRB.CONTINUOUS,
    name="Crude Oil Divide",
)

API_Mul_Oil = model.addVars(
    timeline,
    rigs,
    vtype=gurobipy.GRB.CONTINUOUS,
    name="API Mul Oil Produced",
)
```

$$\frac{(21)x_1y_1 + (30)x_2y_2 + (45)x_3y_3 + (15)x_4y_4}{\Sigma x_i y_i} \leq 27$$

Code

```
    # CONSTRAINTS
    model.addConstrs(
        (
            oil_produced[(date, rig)] <= production_per_rig[rig] *
line_opening[(date, rig)]
            for date in timeline
            for rig in rigs
        ),
        name="Oil produced - Barrels per year",
    )
```

$$x_1y_1 + x_2y_2 + x_3y_3 + x_4y_4$$

```
    model.addConstrs(
        (
            line_opening.sum(date, "*") <= 3
            for date in timeline
        ),
        name="Restricted to 3 rigs per year",
    )
```

$$x_1 + x_2 + x_3 + x_4 \leq 3$$

```
    model.addConstrs(
        (
            oil_produced.sum(date, '*') == sum_of_oil_produced[(date)]
            for date in timeline
        ),
        name="Divide of oil produced",
    )
```

$$\Sigma x_i y_i$$

```
    model.addConstrs(
        (
            API_Mul_Oil[(date, rig)] == API_cap[rig]*oil_produced[(date, rig)]
            for date in timeline
            for rig in rigs
        ),
        name="API Mul Oil",
    )
```

$$\frac{(21)x_1y_1 + (30)x_2y_2 + (45)x_3y_3 + (15)x_4y_4}{\Sigma x_i y_i} \leq 27$$

```
    model.addConstrs(
        (
            gurobipy.quicksum(API_Mul_Oil[(date, rig)] for rig in rigs)
            == yearly_requirements[date] * sum_of_oil_produced[(date)]
            for date in timeline

        ),
        name="Oil produced API - Requirement",
    )
```

```
# DEFINE MODEL
# Objective : maximize a function
model.ModelSense = gurobipy.GRB.MAXIMIZE

# Function to maximize [maximize the sum]
optimization_var = gurobipy.quicksum(
    oil_produced[(date, rig)] for date in timeline for rig in rigs
)
objective = 0
objective += optimization_var
```

```
    # SOLVE MODEL
    model.setObjective(objective)
    model.optimize()
    sol = pd.DataFrame(data={"Solution": model.X}, index=model.VarName)

    print("Total produced = " + str(model.ObjVal) + " barrels")

    return sol, model
```

Code

```python
def plot_planning(planning, timeline):

    # Plot graph - Oil each rig produced
    source = planning.filter(like="Crude Oil Production",axis=0).copy()
    source["Date"] = list(source.index.values)
    source = source.rename(columns={"Solution": "Barrels"}).reset_index()
    source[["Date", "Rig"]] = source["Date"].str.split(",", expand=True)
    source["Date"] = source["Date"].str.split("[").str[1]
    source["Rig"] = source["Rig"].str.split("]").str[0]
    bars = (
        alt.Chart(source)
        .mark_bar()
        .encode(
            x="Rig:N",
            y="Barrels:Q",
            column=alt.Column("Date:N"),
            color="Rig:N",
            tooltip=["Date", "Rig", "Barrels"],
        )
        .interactive()
        .properties(
            width=550 / len(timeline) - 22,
            height=150,
            title="Optimized Production of each rig"
        )
    )
```

defining source

	index	Barrels	Date	Rig
0	Crude Oil Production[2015/01/01,RigA]	0.000000e+00	2015/01/01	RigA
1	Crude Oil Production[2015/01/01,RigB]	1.500000e+06	2015/01/01	RigB
2	Crude Oil Production[2015/01/01,RigC]	1.000000e+06	2015/01/01	RigC
3	Crude Oil Production[2015/01/01,RigD]	1.875000e+06	2015/01/01	RigD
4	Crude Oil Production[2016/01/01,RigA]	1.900000e+06	2016/01/01	RigA
5	Crude Oil Production[2016/01/01,RigB]	0.000000e+00	2016/01/01	RigB
6	Crude Oil Production[2016/01/01,RigC]	1.000000e+06	2016/01/01	RigC
7	Crude Oil Production[2016/01/01,RigD]	1.700000e+06	2016/01/01	RigD
8	Crude Oil Production[2017/01/01,RigA]	0.000000e+00	2017/01/01	RigA
9	Crude Oil Production[2017/01/01,RigB]	1.500000e+06	2017/01/01	RigB
10	Crude Oil Production[2017/01/01,RigC]	1.000000e+06	2017/01/01	RigC
11	Crude Oil Production[2017/01/01,RigD]	0.000000e+00	2017/01/01	RigD
12	Crude Oil Production[2018/01/01,RigA]	1.900000e+06	2018/01/01	RigA
13	Crude Oil Production[2018/01/01,RigB]	1.000000e+05	2018/01/01	RigB
14	Crude Oil Production[2018/01/01,RigC]	0.000000e+00	2018/01/01	RigC
15	Crude Oil Production[2018/01/01,RigD]	2.300000e+06	2018/01/01	RigD
16	Crude Oil Production[2019/01/01,RigA]	1.666667e+06	2019/01/01	RigA
17	Crude Oil Production[2019/01/01,RigB]	1.500000e+06	2019/01/01	RigB
18	Crude Oil Production[2019/01/01,RigC]	1.000000e+06	2019/01/01	RigC
19	Crude Oil Production[2019/01/01,RigD]	0.000000e+00	2019/01/01	RigD

print(source)

Code

```
# Plot graph - Total oil produced
    source2 = source.groupby("Date")["Barrels"].sum().reset_index()
    bars_sum = (
        alt.Chart(source2)
        .mark_bar()
        .encode(
            y=alt.Y("Barrels", axis=alt.Axis(grid=False)),
            column=alt.Column("Date:N"),
            color="Date:N",
            tooltip=["Date", "Barrels"],
        )
        .interactive()
        .properties(
            width=550 / len(timeline) - 22,
            height=75,
            title="Total Production"
        )
    )
```

defining source2

	Date	Barrels
0	2015/01/01	4.375000e+06
1	2016/01/01	4.600000e+06
2	2017/01/01	2.500000e+06
3	2018/01/01	4.300000e+06
4	2019/01/01	4.166667e+06

print(source2)

Code

```
# Plot graph - Raw revenue
source3 = source.groupby("Date")["Barrels"].sum().reset_index()
source3["Raw Rev"] = source3["Barrels"]*75

bars_rawrev = (
    alt.Chart(source3)
    .mark_bar()
    .encode(
        y=alt.Y("Raw Rev", axis=alt.Axis(grid=False)),
        column=alt.Column("Date:N"),
        color="Date:N",
        tooltip=["Date", "Raw Rev"],
    )
    .interactive()
    .properties(
        width=550 / len(timeline) - 22,
        height=75,
        title="Raw Revenue"
    )
)
```

defining source3

	Date	Barrels	Raw Rev
0	2015/01/01	4.375000e+06	328125000.0
1	2016/01/01	4.600000e+06	345000000.0
2	2017/01/01	2.500000e+06	187500000.0
3	2018/01/01	4.300000e+06	322500000.0
4	2019/01/01	4.166667e+06	312500000.0

print(source3)

Code

```
# Plot graph - Adjusted revenue
# raw_rev*(1-0.05)^year-1 - royalties*(1+0.02)^year-1
source4 = planning.filter(like="Use status",axis=0).copy()
source4["Date"] = list(source4.index.values)
source4 = source4.rename(columns={"Solution": "Usage"}).reset_index()
source4[["Date", "Rig"]] = source4["Date"].str.split(",", expand=True)
source4["Date"] = source4["Date"].str.split("[").str[1]
source4["Rig"] = source4["Rig"].str.split("]").str[0]
source4["Year"] = source4["Date"].str.split("/").str[0]
source4["Year"] = pd.to_numeric(source4["Year"])
source4["Royalties"] = source4["Year"] # Dummy
```

defining source4

```
    # Calculate royalities by rig and year
    for index, row in source4.iterrows():
        if(source4["Rig"][index] == "RigA"):
            source4["Royalties"][index] =
source4["Usage"][index]*(13085600)*((1+0.02)**(source4["Year"][index]-2015))
        elif(source4["Rig"][index] == "RigB"):
            source4["Royalties"][index] =
source4["Usage"][index]*(16357000)*((1+0.02)**(source4["Year"][index]-2015))
        elif(source4["Rig"][index] == "RigC"):
            source4["Royalties"][index] =
source4["Usage"][index]*(13085600)*((1+0.02)**(source4["Year"][index]-2015))
        elif(source4["Rig"][index] == "RigD"):
            source4["Royalties"][index] =
source4["Usage"][index]*(16357000)*((1+0.02)**(source4["Year"][index]-2015))

    source4 = source4.groupby("Date")["Royalties"].sum().reset_index()
    source4["Raw Rev"] = source3["Raw Rev"]
    source4["Pre adj Rev"] = source4["Raw Rev"] # Dummy

    for index, row in source4.iterrows():
        source4["Pre adj Rev"][index] *= (1-0.05)**index

    source4["Adjusted Rev"] = source4["Pre adj Rev"]-source4["Royalties"]
    print(source4)


    bars_adjustrev = (
        alt.Chart(source4)
        .mark_bar()
        .encode(
            y=alt.Y("Adjusted Rev", axis=alt.Axis(grid=False)),
            column=alt.Column("Date:N"),
            color="Date:N",
            tooltip=["Date", "Adjusted Rev"],
        )
        .interactive()
        .properties(
            width=550 / len(timeline) - 22,
            height=75,
            title="Adjusted Revenue"
        )
    )
```

	Date	Royalties	Raw Rev	Pre adj Rev	Adjusted Rev
0	2015/01/01	45799600	328125000.0	3.281250e+08	2.823254e+08
1	2016/01/01	43378764	345000000.0	3.277500e+08	2.843712e+08
2	2017/01/01	30632080	187500000.0	1.692188e+08	1.385867e+08
3	2018/01/01	48602901	322500000.0	2.765034e+08	2.279005e+08
4	2019/01/01	46033890	312500000.0	2.545332e+08	2.084993e+08

print(source4)

Code

```
    # Plot graph - API requirements
    source5 = source2.copy()
    API_dummy = [27, 24, 36, 18, 30]
    source5['API'] = API_dummy
    source5.drop('Barrels', inplace=True, axis=1)
#     print(source5)

    bars_api = (
        alt.Chart(source5)
        .mark_bar()
        .encode(
            y=alt.Y("API", axis=alt.Axis(grid=False)),
            column=alt.Column("Date:N"),
            color="Date:N",
            tooltip=["Date", "API"],
        )
        .interactive()
        .properties(
            width=550 / len(timeline) - 22,
            height=75,
            title="API Requirements"
        )
    )
```

defining source5

Code

```
# Vertically Concatenate all graphs
    chart = alt.vconcat(bars, bars_sum, bars_rawrev, bars_adjustrev)
    chart.save("Oil_plan_model_4.html")

    dp.Report(dp.Plot(chart, caption="Oil production model 4")).upload(
        name="Oil production model 4",
        description="Oil production model 4 by Samkok",
        open=True,
        visibily="PUBLIC",
    )
```

```
    yearly_requirements: Dict[str, int] = {
        "2015/01/01": 27,
        "2016/01/01": 24,
        "2017/01/01": 36,
        "2018/01/01": 18,
        "2019/01/01": 30
    }

    yearly_requirements_df = pd.DataFrame.from_dict(yearly_requirements, orient="index")

    calendar: List[str] = list(yearly_requirements.keys())

    production_per_rig = {"RigA": 1900000, "RigB": 1500000, "RigC": 1000000, "RigD":
    2300000}

    rigs: List[str] = list(production_per_rig.keys())

    API_cap = {"RigA": 21, "RigB": 30, "RigC": 45, "RigD": 15}

    solution, model = optimize_planning(
        calendar,
        rigs,
        yearly_requirements,
        production_per_rig,
        API_cap
    )
    # print(solution)
    # solution.to_csv('Optimized_test.csv', encoding='utf-8')
    plot_planning(solution, calendar)
    # print(model.ObjVal)
```

Summary for Production Planning

1st year:

- Rig A: 0 barrels
- Rig B: 1,500,000 barrels
- RigC: 1,000,000 barrels
- Rig D: 1,875,000 barrels.
- Total oil production: 4,375,000 barrels

3rd year:

- Rig A: 0 barrels
- Rig B: 1,500,000 barrels
- Rig C: 1,000,000 barrels
- Rig D: 0 barrels
- Total oil production: 2,500,000 barrels.

5th year:

- Rig A: 1,666,666.67 barrels
- Rig B: 1,500,000 barrels
- Rig C: 1,000,000 barrels
- Rig D: 0 barrels
- Total oil production: 4,166,666.67 barrels.

2nd year:

- Rig A: 1,900,000 barrels
- Rig B: 0 barrels
- Rig C: 1,000,000 barrels
- RigD: 1,700,000 barrels.
- Total oil production: 4,600,000 barrels.

4th year:

- Rig A: 1,900,000 barrels
- Rig B: 100,000 barrels
- Rig C: 0 barrels
- Rig D: 2,300,000 barrels
- Total oil production: 4,300,000 barrels

In which we use these optimized results to calculate the raw revenue and adjusted revenue as well.