

```
In [1]: import pandas as pd
import numpy as np
import os
from datetime import datetime

In [2]: # DEFINE PARAMETERS

# States to analyze
states = ['Edo', 'Cross River', 'Ondo', 'Akwa Ibom', 'Delta', 'Imo', 'Rivers']

# Base parameters
EXTRACTION_RATE = 0.21 # 21% CPO from FFB
TAX_RATE = 0.30 # 30%
DISCOUNT_RATE = 0.12
TARGET_PROFIT = 1_000_000_000 # ₦1 billion

In [3]: # State-specific data
state_info = {
    'Edo': {
        'FFB_Yield_Base': 19.0,
        'Land_Cost': 60000,
        'Rainfall_mm': 2000,
        'Suitability': 9.0
    },
    'Cross River': {
        'FFB_Yield_Base': 18.0,
        'Land_Cost': 300000,
        'Rainfall_mm': 2400,
        'Suitability': 9.5
    },
    'Ondo': {
        'FFB_Yield_Base': 16.0,
        'Land_Cost': 550000,
        'Rainfall_mm': 2200,
        'Suitability': 8.5
    },
    'Akwa Ibom': {
        'FFB_Yield_Base': 18.0,
        'Land_Cost': 450000,
        'Rainfall_mm': 2500,
        'Suitability': 9.5
    },
    'Delta': {
        'FFB_Yield_Base': 15.0,
        'Land_Cost': 55000,
        'Rainfall_mm': 2100,
        'Suitability': 8.0
    },
    'Imo': {
        'FFB_Yield_Base': 15.0,
        'Land_Cost': 600000,
        'Rainfall_mm': 2000,
        'Suitability': 7.5
    },
    'Rivers': {
        'FFB_Yield_Base': 15.0,
        'Land_Cost': 450000,
        'Rainfall_mm': 2500,
        'Suitability': 8.5
    }
}

In [4]: # Scenario definitions
scenarios = [
    'Conservative': {
        'Yield_Factor': 0.67, # 67% of base yield
        'CPO_Price': 1_200_000,
        'OPEX_per_ha': 805_000
    },
    'Base': {
        'Yield_Factor': 1.00, # 100% of base yield
        'CPO_Price': 1_500_000,
        'OPEX_per_ha': 700_000
    },
    'Optimistic': {
        'Yield_Factor': 1.28, # 128% of base yield
        'CPO_Price': 2_000_000,
        'OPEX_per_ha': 595_000
    }
]

In [5]: # Fixed costs
CAPEX_per_ha = 826_000 # Land prep + seedlings + planting
MILL_CAPEX = 447_000_000 # Processing mill

In [6]: # BUILD THE MASTER DATA TABLE
master_data = []

for state, info in state_info.items():
    for scenario_name, scenario_params in scenarios.items():
        # Calculate yield for this scenario
        ffb_yield = info['FFB_Yield_Base'] * scenario_params['Yield_Factor']
        cpo_yield = ffb_yield * EXTRACTION_RATE

        # Calculate financials per hectare
        revenue_per_ha = cpo_yield * scenario_params['CPO_Price']
        opex_per_ha = scenario_params['OPEX_per_ha']
        gross_profit_per_ha = revenue_per_ha - opex_per_ha
        tax_per_ha = max(0, gross_profit_per_ha * TAX_RATE)
        net_profit_per_ha = gross_profit_per_ha - tax_per_ha

        # Calculate required scale for ₦1 billion
        if net_profit_per_ha > 0:
            hectares_needed = TARGET_PROFIT / net_profit_per_ha
        else:
            hectares_needed = 999999 # Not feasible

        # Calculate total investment
        land_cost_total = info['Land_Cost'] * hectares_needed
        capex_total = CAPEX_per_ha * hectares_needed
        total_investment = land_cost_total + capex_total + MILL_CAPEX

        # Calculate 15-year metrics
        # Assume 3 years no production, then ramp to peak by year 8
        production_years = 12 # Years 4-15 (with ramping)
        avg_production_factor = 0.85 # Average 85% of peak over 15 years

        total_15yr_profit = net_profit_per_ha * hectares_needed * production_years * avg_production_factor

        # ROI calculation
        roi = ((total_15yr_profit - total_investment) / total_investment * 100) if total_investment > 0 else 0

        # Payback period (simplified)
        annual_profit_avg = net_profit_per_ha * hectares_needed * avg_production_factor
        if annual_profit_avg > 0:
            payback_years = (total_investment / annual_profit_avg) + 3 # Add 3 years for maturity
        else:
            payback_years = 99

        # NPV calculation (simplified)
        npv = 0
        for year in range(16):
            if year <= 3:
                cash_flow = 0
            elif year <= 7:
                cash_flow = net_profit_per_ha * hectares_needed * 0.25
            elif year <= 7:
                ramp_factor = 0.45 + (year - 3) * 0.14 # Ramp from 45% to 95%
                cash_flow = net_profit_per_ha * hectares_needed * ramp_factor
            else:
                cash_flow = net_profit_per_ha * hectares_needed

            npv += cash_flow / ((1 + DISCOUNT_RATE) ** year)

        npv = npv - total_investment # Subtract initial investment

        # IRR approximation (simplified)
        if total_investment > 0 and total_15yr_profit > total_investment:
            irr = ((total_15yr_profit / total_investment) ** (1/15) - 1) * 100
        else:
            irr = 0

        # Profit margin
        profit_margin = (net_profit_per_ha / revenue_per_ha * 100) if revenue_per_ha > 0 else 0

        # Store all data
        master_data.append({
            'State': state,
            'Scenario': scenario_name,
            'Plantation_Size_ha': round(hectares_needed, 0),
            'FFB_Yield_tonnes_per_ha': round(ffb_yield, 1),
            'CPO_Yield_tonnes_per_ha': round(cpo_yield, 2),
            'CPO_Price_NGN_per_tonne': scenario_params['CPO_Price'],
            'Revenue_per_ha_NGN': round(revenue_per_ha, 0),
            'OPEX_per_ha_NGN': round(opex_per_ha, 0),
            'CAPEX_per_ha_NGN': round(capex_total, 0),
            'Gross_Profit_per_ha_NGN': round(gross_profit_per_ha, 0),
            'Net_Profit_per_ha_NGN': round(net_profit_per_ha, 0),
            'Profit_Margin_%': round(profit_margin, 1),
            'Land_Cost_NGN': round(land_cost_total, 0),
            'Plantation_CAPEX_NGN': round(capex_total, 0),
            'MILL_CAPEX_NGN': MILL_CAPEX,
            'Total_Investment_NGN': round(total_investment, 0),
            'Total_Investment_Billions': round(total_investment / 1e9, 2),
            'Annual_Target_Profit_NGN': TARGET_PROFIT,
            'Total_15yr_Profit_NGN': round(total_15yr_profit, 0),
            'NPV_NGN': round(npv, 0),
            'NPV_Billions': round(npv / 1e9, 2),
            'IRR_%': round(irr, 1),
            'ROI_%': round(roi, 1),
            'Payback_Years': round(payback_years, 1),
            'Break_even_ha': round(hectares_needed, 0),
            'Rainfall_mm': info['Rainfall_mm'],
            'Suitability_Score': info['Suitability'],
            'Land_Cost_per_ha_NGN': info['Land_Cost']
        })

In [7]: # Create DataFrame
df_master = pd.DataFrame(master_data)

In [8]: # ADD RANKINGS

# Rank by profitability per hectare within each scenario
for scenario in ['Conservative', 'Base', 'Optimistic']:
    scenario_data = df_master[df_master['Scenario'] == scenario].copy()
    scenario_data['Profit_Rank'] = scenario_data['Net_Profit_per_ha_NGN'].rank(ascending=False, astype(int))
    df_master.loc[df_master['Scenario'] == scenario, 'Profit_Rank'] = scenario_data['Profit_Rank'].values

In [9]: # REORDER COLUMNS FOR CLARITY

column_order = [
    'State',
    'Scenario',
    'Plantation_Size_ha',
    'FFB_Yield_tonnes_per_ha',
    'CPO_Yield_tonnes_per_ha',
    'CPO_Price_NGN_per_tonne',
    'Revenue_per_ha_NGN',
    'OPEX_per_ha_NGN',
    'CAPEX_per_ha_NGN',
    'Gross_Profit_per_ha_NGN',
    'Net_Profit_per_ha_NGN',
    'Profit_Margin_%',
    'Total_Investment_NGN',
    'Total_Investment_Billions',
    'Annual_Target_Profit_NGN',
    'Total_15yr_Profit_NGN',
    'NPV_NGN',
    'NPV_Billions',
    'IRR_%',
    'ROI_%',
    'Payback_Years',
    'Break_even_ha',
    'Rainfall_mm',
    'Suitability_Score',
    'Land_Cost_per_ha_NGN',
    'Plantation_CAPEX_NGN',
    'MILL_CAPEX_NGN',
    'Profit_Rank'
]

df_master = df_master[column_order]

# Sort by State and Scenario
df_master = df_master.sort_values(['State', 'Scenario'])

In [10]: # SAVE TO EXCEL AND CSV (WITH TIMESTAMP)

# Generate timestamp (e.g., 2025-11-11_19-45)
timestamp = datetime.now().strftime("%Y-%m-%d_%H-%M")

# File names with timestamp
output_excel = fr'C:\Users\hp\Documents\MASTER_PALM_OIL_ANALYSIS-{timestamp}.xlsx'
output_csv = fr'C:\Users\hp\Documents\MASTER_PALM_OIL_ANALYSIS-{timestamp}.csv'

# Ensure your DataFrame (df_master) and states list exist
# Example: df_master = your_final_dataset.DataFrame
# Example: states = ["Edo", "Ondo", "Cross River"]

# Create folder if not existing
os.makedirs(os.path.dirname(output_excel), exist_ok=True)

# 1. Save as CSV (recommended for Power BI)
df_master.to_csv(output_csv, index=False, encoding='utf-8-sig')
print(f"CSV file created successfully at: {output_csv}")

# 2. Save as Excel with formatting
try:
    with pd.ExcelWriter(output_excel, engine='openpyxl') as writer:
        # Write main data
        df_master.to_excel(writer, sheet_name='Complete_Analysis', index=False)

        # Get workbook and worksheet
        workbook = writer.book
        worksheet = writer.sheets['Complete_Analysis']

        # Auto-adjust column widths safely
        for idx, col in enumerate(df_master.columns):
            max_length = max(
                df_master[col].astype(str).apply(len).max(),
                len(col)
            )
            # Handle column names beyond 'Z' properly
            if idx < 26:
                col_letter = chr(65 + idx)
            else:
                first = chr(64 + (idx // 26))
                second = chr(65 + (idx % 26))
                col_letter = f"{first}{second}"

            worksheet.column_dimensions[col_letter].width = min(max_length + 2, 50)

        # Freeze first row
        worksheet.freeze_panes = 'A2'

        print(f"Excel file created successfully at: {output_excel}")
        print(f"Total rows: {len(df_master)}")
        print(f"States: {len(states)}")
        print(f"Scenarios per state: 3 (Conservative/Base/Optimistic)")
        print(f"Total combinations: {len(df_master)}")
except Exception as e:
    print(f"Error saving Excel file: {e}")

CSV file created successfully at: C:\Users\hp\Documents\MASTER_PALM_OIL_ANALYSIS_2025-12-21_18-39.csv
Excel file created successfully at: C:\Users\hp\Documents\MASTER_PALM_OIL_ANALYSIS_2025-12-21_18-39.xlsx
Total rows: 21
States: 7
Scenarios per state: 3 (Conservative/Base/Optimistic)
Total combinations: 21

In [11]: # DISPLAY SUMMARY

print("SUMMARY: TOP RECOMMENDATIONS")

# Get Base scenario only for recommendations
base_scenario = df_master[df_master['Scenario'] == 'Base'].copy()
base_scenario = base_scenario.sort_values('Net_Profit_per_ha_NGN', ascending=False)

print(f"TOP 3 STATES (BASE CASE SCENARIO):")
for idx, row in base_scenario.head(3).iterrows():
    print(f"State: {row['State']}")
    print(f"Plantation Size: {row['Plantation_Size_ha']:.0f} hectares")
    print(f"Investment: ₦{row['Total_Investment_Billions']:.2f} billion")
    print(f"Profit per hectare: ₦{row['Net_Profit_per_ha_NGN']:.0f}")
    print(f"NPV: ₦{row['NPV_Billions']:.2f} billion")
    print(f"IRR: {row['IRR_%']:.1f}%")
    print(f"ROI: {row['ROI_%']:.1f}%")
    print(f"Payback: {row['Payback_Years']:.1f} years")

print("\n" + "-" * 80)
print("SCENARIO COMPARISON (Best State - Edo)")
print("-" * 80)

edo_scenarios = df_master[df_master['State'] == 'Edo'].copy()
print(edo_scenarios[['Scenario', 'Plantation_Size_ha', 'Total_Investment_Billions',
    'NPV_Billions', 'IRR_%', 'ROI_%', 'Payback_Years']].to_string(index=False))

print("\n" + "-" * 80)
print("CREATE THESE VISUALS:")
print("1. Map: States colored by profitability")
print("2. Bar Chart: Net Profit per hectare by State")
print("3. Scatter: Investment vs NPV")
print("4. Column Chart: Scenarios comparison (Conservative/Base/Optimistic)")
print("5. Table: Top 3 states with all metrics")
print("6. Line Chart: Payback period by state")
print("7. Scatter: Yield vs Rainfall")
print("8. Cards: Total Investment, Hectares, NPV, ROI")

best_state = base_scenario.iloc[0]
print(f"At what scale and in which states will palm oil generate ₦1B profit?")
print(f"SCALE: {best_state['Plantation_Size_ha']:.0f} hectares")
print(f"INVESTMENT: ₦{best_state['Total_Investment_Billions']:.2f} billion")
print(f"EXPECTED RETURN: {best_state['IRR_%']:.1f}% IRR, {best_state['ROI_%']:.1f}% ROI")
print(f>PAYBACK: {best_state['Payback_Years']:.1f} years")

SUMMARY: TOP RECOMMENDATIONS

TOP 3 STATES (BASE CASE SCENARIO):

1.0. Edo
Plantation Size: 270 hectares
Investment: ₦0.83 billion
Profit per hectare: ₦5,699,500
NPV: ₦3.28 billion
IRR: 18.2%
ROI: 1125.3%
Payback: 4.0 years

2.0. Akwa Ibom
Plantation Size: 287 hectares
Investment: ₦0.81 billion
Profit per hectare: ₦5,479,000
NPV: ₦3.30 billion
IRR: 18.4%
ROI: 1153.4%
Payback: 4.0 years

2.0. Cross River
Plantation Size: 287 hectares
Investment: ₦0.77 billion
Profit per hectare: ₦5,479,000
NPV: ₦3.34 billion
IRR: 18.8%
ROI: 1223.5%
Payback: 3.9 years

=====
SCENARIO COMPARISON (Best State - Edo)
=====

Scenario Plantation_Size_ha Total_Investment_Billions NPV_Billions IRR_% ROI_% Payback_Years
Base 270.0 0.83 3.28 18.2 1125.3 4.0
Conservative 595.0 1.29 2.82 14.8 687.8 4.5
Optimistic 149.0 0.66 3.45 20.0 1448.3 3.8

=====
CREATE THESE VISUALS:
1. Map: States colored by profitability
2. Bar Chart: Net Profit per hectare by State
3. Scatter: Investment vs NPV
4. Column Chart: Scenarios comparison (Conservative/Base/Optimistic)
5. Table: Top 3 states with all metrics
6. Line Chart: Payback period by state
7. Scatter: Yield vs Rainfall
8. Cards: Total Investment, Hectares, NPV, ROI
'At what scale and in which states will palm oil generate ₦1B profit?'

STATE: Edo
```

