# AUTOMATION AND ANALYSIS OF THE EIA WEEKLY PETROLEUM INVENTORY REPORT

**Enhancing Market Strategy with Real-Time Data Processing and Automated Reporting**

## ABSTRACT

This document outlines the development and implementation of a Python script to automate the weekly download and analysis of the EIA Petroleum Inventory Report. The script is scheduled to run every Wednesday at 10:30 AM EST, downloading the report in PDF format, performing a preliminary analysis, and sending a detailed summary to key stakeholders via email. The automation process enhances decision-making in the crude oil market by providing timely insights into inventory levels, demand forecasts, and potential market movements, allowing for more informed short-term and long-term trading strategies.

William Pape
**Penn Quarter Capital, Inc.**

## Contents

Crude Oil Market Analysis 5

Abstract 5

Introduction 5

Market Sentiment and Positioning: 6

Supply and Demand Factors: 6

Technical Factors: 6

Recommendation 6

Strategic Positioning: 6

Actionable Steps: 7

1. Initiate Long Positions in Brent Futures: 7
2. Consider Call Options: 7
3. Monitor Geopolitical Developments: 7
4. Hedge Against Macroeconomic Risks: 7

Conclusion: 7

CLU24 Contracts are trading today, 13 August 2024 at $78.51 7

1. Assessment of Current Position: 7

2. Market Analysis Recap: 7

3. Options for Positioning: 8

A. Hold and Monitor: 8

B. Average Down: 8

C. Partial Exit: 8

4. Recommendation: 8

5. Final Considerations: 9

Advantages of Adding a Bloomberg Terminal 9

1. Real-Time Data and Analytics: 9
2. Comprehensive Market Coverage: 9
3. Advanced Analytical Tools: 9
4. Access to Industry Research: 9
5. News and Alerts: 9
6. Networking and Communication: 10

Recommended Professional-Level Data Sources for the Oil Markets 10

1. Thomson Reuters Eikon: 10

2. S&P Global Platts: 10
3. Argus Media: 10
4. ICE Data Services (Intercontinental Exchange): 10
5. Genscape (Wood Mackenzie): 10
6. OilX: 10
7. Rystad Energy: 10
8. Energy Information Administration (EIA): 11
9. International Energy Agency (IEA): 11
Conclusion 11
Python Script for Downloading EIA Weekly Petroleum Inventory Report 12
Creating a Python script to automatically download the weekly EIA Petroleum Inventory Report, analyze it, and send an email requires a combination of several tasks: 12
1. Scheduling the Script 12
2. Downloading the Report: 12
3. Uploading the PDF to ChatGPT-4o: 12
4. Generating an Analysis: 12
5. Emailing the Analysis: 12
Below is a Python script that covers these tasks 13
Python Script 15
Copy code 15
# import libraries 15
# Function to download the EIA Petroleum Inventory Report 15
# Example URL - you would need to ensure this is the correct path to the weekly report 15
# Get the current date to name the file 15
# Send a GET request to download the file 15
# Save the PDF locally 16
# Here you would upload the PDF to ChatGPT-4o or perform the analysis locally. 16
# Placeholder for analysis function. 16
# Email the analysis 16
# Placeholder function to simulate analysis. 16
# This is where you'd parse the PDF and create your analysis. 16
# Example analysis result: 16
# Function to send the analysis via email 17

- # Create the email content........................................................................................................ 17
- # Attach the analysis as the email body................................................................................... 17
- # Setup the SMTP server (example with Gmail)..................................................................... 17
- # Schedule the download every Wednesday at 10:30 AM EST........................................... 17
- # Keep the script running........................................................................................................ 18

Conclusion.................................................................................................................................... 18

Figure 1:Python Script - Import Libraries................................................................................. 13
Figure 2:Python Script - Setup the Analysis ............................................................................. 14
Figure 3:Python Script - Run Script........................................................................................... 15

# Crude Oil Market Analysis

## Abstract

This document presents a comprehensive guide to automating the retrieval and analysis of the U.S. Energy Information Administration's (EIA) Weekly Petroleum Inventory Report using Python. The primary objective of this automation is to streamline the process of obtaining real-time data and performing immediate market analysis, which is crucial for professionals engaged in the crude oil markets. The script is designed to download the report every Wednesday at precisely 10:30 AM EST, analyze the data, generate actionable insights, and automatically distribute these insights via email.

By leveraging this automated system, traders and analysts can significantly enhance their market responsiveness, ensuring they are well-positioned to capitalize on emerging trends and mitigate risks. The document also discusses the potential implications of the inventory data on short-term and long-term market strategies, highlighting the importance of integrating real-time data analysis into decision-making processes. This approach provides a competitive edge in a market where timely and accurate information is critical for success.

## Introduction

The crude oil market is characterized by its volatility and susceptibility to a myriad of factors, including geopolitical events, supply chain disruptions, and shifts in global demand. As such, timely and accurate data is essential for making informed trading decisions. The U.S. Energy Information Administration (EIA) provides one of the most critical data sources in the form of its Weekly Petroleum Inventory Report. This report offers insights into the levels of crude oil and petroleum products held in inventory, which directly impacts market pricing and sentiment. Given the importance of this data, there is a growing need for automated tools that can streamline the process of data retrieval and analysis, enabling traders to act swiftly on the latest information.

In this document, we outline the development of a Python-based script designed to automate the download and analysis of the EIA's Weekly Petroleum Inventory Report. The script is scheduled to execute every Wednesday at 10:30 AM EST, coinciding with the report's release. This automated process not only saves time but also reduces the potential for human error in data collection and initial analysis. By integrating this script into a trader's toolkit, professionals can receive timely, pre-analyzed data directly to their inbox, allowing for immediate action based on the latest inventory levels and market trends.

The introduction of this automated system represents a significant advancement in how market data is processed and utilized. In addition to outlining the technical aspects of the script, this document also explores the broader implications of incorporating real-time data analysis into trading strategies. We will examine how the information provided by the EIA report can influence market movements and offer recommendations on how to position oneself in both the short and long term. This approach underscores the critical role of automation in maintaining a competitive edge in the fast-paced world of crude oil trading.

**Market Sentiment and Positioning:** The current sentiment in the crude oil market is overwhelmingly bearish, as indicated by the record low in bullish positioning among hedge funds and other money managers. The aggressive sell-off, particularly in Brent and WTI contracts, reflects widespread concerns over a potential global economic slowdown. This sentiment is further exacerbated by the broader retreat from risk across asset markets, as investors brace for potential recessionary impacts on oil consumption.

The reduction of bullish positions by 372 million barrels since July 2024 underscores a significant shift in market dynamics, driven by fears of declining demand amidst economic uncertainty. The last time such a rapid liquidation occurred was during the onset of the COVID-19 pandemic, suggesting that investors are once again prioritizing risk management over potential upside in the market.

**Supply and Demand Factors:** Despite the bearish positioning, several supply-side factors could support a potential price rebound. Geopolitical risks, such as potential disruptions in oil production from Libya and Iran, as well as ongoing tensions involving Ukraine and Western Russia, introduce significant uncertainties that could lead to sudden supply shocks. Additionally, the physical oil market appears to be tightening, with global oil stocks showing signs of modest draws, which could further support prices.

On the demand side, although there are concerns about a slowdown in major economies, there are also signs of resilience in emerging markets, particularly in India, where oil demand jumped by 7% year-over-year in July 2024. Furthermore, recent solid economic data from the United States has shifted some investor focus from recession fears back to geopolitical and supply-related risks, contributing to a modest recovery in oil prices.

**Technical Factors:** From a technical perspective, the recent bounce in Brent futures from a low of $75 to above $80 per barrel suggests that the market may be finding a near-term bottom. The extreme bearish positioning makes the market susceptible to a short squeeze, especially if any unexpected positive news or geopolitical developments arise. This could trigger a rapid unwinding of short positions, leading to a sharp price rally.

Goldman Sachs’ forecast for Brent to recover to the mid-$80s per barrel is supported by historical trends where low positioning has often been followed by strong returns over the subsequent three months. The increase in implied volatility skew for Brent call options also indicates that traders are beginning to price in the possibility of a rebound.

## Recommendation

**Strategic Positioning:** Given the current market dynamics, I recommend adopting a cautiously bullish stance on crude oil. The extreme bearish sentiment, coupled with potential supply disruptions and resilient demand in certain regions, creates a favorable risk-reward scenario for initiating long positions. However, this strategy should be tempered with caution due to the underlying macroeconomic risks.

## Actionable Steps:

1. Initiate Long Positions in Brent Futures: Consider building a position in Brent futures at current levels, targeting a move towards the mid-$80s per barrel over the next three months. This position should be closely monitored, with stop-loss levels set around the recent lows of $75 per barrel to mitigate downside risk.
2. Consider Call Options: To limit exposure while capitalizing on potential upside, purchasing call options on Brent could be an effective strategy. This allows for participation in a potential price rebound with defined risk, particularly given the recent increase in implied volatility skew.
3. Monitor Geopolitical Developments: Stay attuned to developments in key geopolitical hotspots, including Libya, Iran, and the broader Middle East. Any escalation in these regions could significantly impact oil prices and provide opportunities to adjust positions.
4. Hedge Against Macroeconomic Risks: While a bullish stance is recommended, it is crucial to hedge against the risk of a broader economic downturn. This could involve taking protective positions in other asset classes or using futures and options to hedge oil exposure.

Conclusion: While the current bearish positioning in the crude oil market reflects significant concerns about future demand, the potential for a short squeeze and geopolitical disruptions presents a compelling opportunity for a strategic entry into long positions. By carefully managing risk and staying responsive to market developments, investors can potentially capitalize on a rebound in oil prices over the coming months.

## CLU24 Contracts are trading today, 13 August 2024 at $78.51, I currently hold 2 CLU24 contracts at $79.05. How should I position myself?

Given your current position in two CLU24 contracts at $79.05 and the market trading at $78.51 as of 13 August 2024, you are currently at a slight unrealized loss. Below is a strategic approach to positioning yourself in this situation:

## 1. Assessment of Current Position:

- **Entry Price:** $79.05
- **Current Price:** $78.51
- **Unrealized Loss:** $0.54 per barrel (or $540 per contract, considering a standard futures contract size of 1,000 barrels).

## 2. Market Analysis Recap:

- The market has shown significant bearish sentiment, but with signs of a potential rebound due to extremely low bullish positioning, geopolitical risks, and tight physical markets.
- There is a possibility of a short squeeze, which could drive prices back up toward the mid-$80s per barrel over the next few months.

# 3. Options for Positioning:

## A. Hold and Monitor:

- **Rationale:** Given the potential for a rebound, holding your current positions might be the best course of action. The recent dip to $78.51 could be a temporary pullback before a move higher, especially if geopolitical tensions increase or if any unexpected supply disruptions occur.
- **Action:**
    - Maintain your two CLU24 contracts.
    - Set a **stop-loss** slightly below the recent low of $75 to limit potential downside risk. This will prevent excessive losses if the market moves against you.
    - Consider setting a **profit-taking target** around $83-$85 per barrel, based on the potential rebound forecasted by analysts like those at Goldman Sachs.

## B. Average Down:

- **Rationale:** If you have a strong conviction in the potential for a rebound and sufficient capital, you could consider averaging down by purchasing additional contracts at the current lower price. This would reduce your overall breakeven point.
- **Action:**
    - Purchase one or two additional CLU24 contracts at $78.51.
    - This would bring your new average entry price down closer to the current market price, improving your potential for profitability if the market rebounds.
    - Implement the same **stop-loss** strategy as above to protect against further downside.

## C. Partial Exit:

- **Rationale:** If you are concerned about further declines and want to mitigate risk, you could consider partially exiting your position to limit exposure while still participating in a potential rebound.
- **Action:**
    - Sell one of your two CLU24 contracts at the current market price of $78.51.
    - This would reduce your exposure and potential loss, while still leaving you with one contract to benefit from a possible price recovery.
    - Maintain a stop-loss on the remaining contract and adjust your profit target accordingly.

# 4. Recommendation:

- **Primary Strategy:** I recommend holding your current position and closely monitoring market developments. The potential for a rebound driven by geopolitical risks and extremely bearish positioning makes this a prudent approach. However, set a stop-loss around $75 to protect against significant downside.

- **Alternative Strategy:** If you are comfortable with additional risk and have sufficient capital, consider averaging down by purchasing one or two additional contracts. This would lower your breakeven point and increase potential gains if the market moves in your favor.

## 5. Final Considerations:

- Stay alert to any significant news that could impact oil prices, such as geopolitical developments or changes in global economic indicators.
- Reassess your position regularly as new data becomes available and adjust your strategy accordingly.

This approach balances the potential for capturing upside with prudent risk management to protect your capital.

Would adding a Bloomberg Terminal to my Tool Box be advantageous? I would like to Level-up to the Professional-Level with regard to access to real-time information. Also, please recommend various professional level data source for the Oil Markets.

## Advantages of Adding a Bloomberg Terminal

A Bloomberg Terminal would indeed be a significant asset to your toolkit, especially as you aim to elevate your professional capabilities in the oil markets. Here are the key advantages:

### 1. Real-Time Data and Analytics:

- Bloomberg provides instantaneous access to real-time market data, including live prices, news, and analytics. This is critical in fast-moving markets like crude oil, where decisions often hinge on the latest information.

### 2. Comprehensive Market Coverage:

- The terminal offers comprehensive coverage of not only the oil markets but also related commodities, global macroeconomic indicators, and geopolitical events. This enables a holistic approach to market analysis.

### 3. Advanced Analytical Tools:

- Bloomberg's suite of analytical tools allows for deep dives into historical data, charting, scenario analysis, and quantitative modeling. These tools can enhance your ability to forecast market movements and identify trading opportunities.

### 4. Access to Industry Research:

- With Bloomberg, you gain access to proprietary research from top analysts and industry experts. This research can provide insights into market trends, forecasts, and investment strategies.

### 5. News and Alerts:

- Bloomberg delivers real-time news and customizable alerts, ensuring that you are immediately informed of any significant developments that could impact the oil markets.

### 6. Networking and Communication:

- The Bloomberg Terminal also includes Bloomberg Messaging, which allows you to communicate directly with other professionals and experts across the industry, fostering valuable connections and collaborations.

## Recommended Professional-Level Data Sources for the Oil Markets

In addition to the Bloomberg Terminal, here are other top-tier data sources that can enhance your professional-level access to the oil markets:

### 1. Thomson Reuters Eikon:

- Similar to Bloomberg, Thomson Reuters Eikon provides real-time data, news, and analysis across commodities markets, including crude oil. It also offers powerful analytical tools and access to in-depth research reports.

### 2. S&P Global Platts:

- S&P Global Platts is a leading provider of benchmark price assessments and analytics in the energy markets. Their services include detailed reports on crude oil, refined products, and market dynamics.

### 3. Argus Media:

- Argus Media offers pricing and analysis for global energy and commodity markets, including detailed assessments of crude oil prices and market conditions. They are known for their independent reporting and market intelligence.

### 4. ICE Data Services (Intercontinental Exchange):

- ICE provides real-time and historical data, analytics, and tools specifically focused on the oil markets. This includes futures and options data, market analytics, and trading platform access.

### 5. Genscape (Wood Mackenzie):

- Genscape offers real-time monitoring of global oil infrastructure, including pipelines, refineries, and storage facilities. Their data is highly valuable for understanding supply chain dynamics and potential market impacts.

### 6. OilX:

- OilX combines satellite imagery, AI, and machine learning to provide real-time oil market intelligence, including supply and demand forecasts, inventory tracking, and global trade flow analysis.

### 7. Rystad Energy:

- Rystad Energy is an independent energy research company offering data, tools, and analysis for the oil market. Their databases and reports are highly regarded for their accuracy and depth.

### 8. Energy Information Administration (EIA):

- The EIA provides free, highly detailed data and reports on U.S. and global energy markets, including production, consumption, and pricing data for crude oil.

### 9. International Energy Agency (IEA):

- The IEA offers reports and data on global energy trends, with a focus on market forecasts, policy impacts, and investment strategies.

## Conclusion

Incorporating a Bloomberg Terminal into your toolkit would significantly enhance your ability to operate at a professional level in the oil markets. Coupled with the recommended data sources, you would have a comprehensive, real-time view of the markets, enabling more informed decision-making and sophisticated trading strategies. This combination will provide you with a competitive edge, allowing you to stay ahead of market trends and respond swiftly to changes.

## Python Script for Downloading EIA Weekly Petroleum Inventory Report

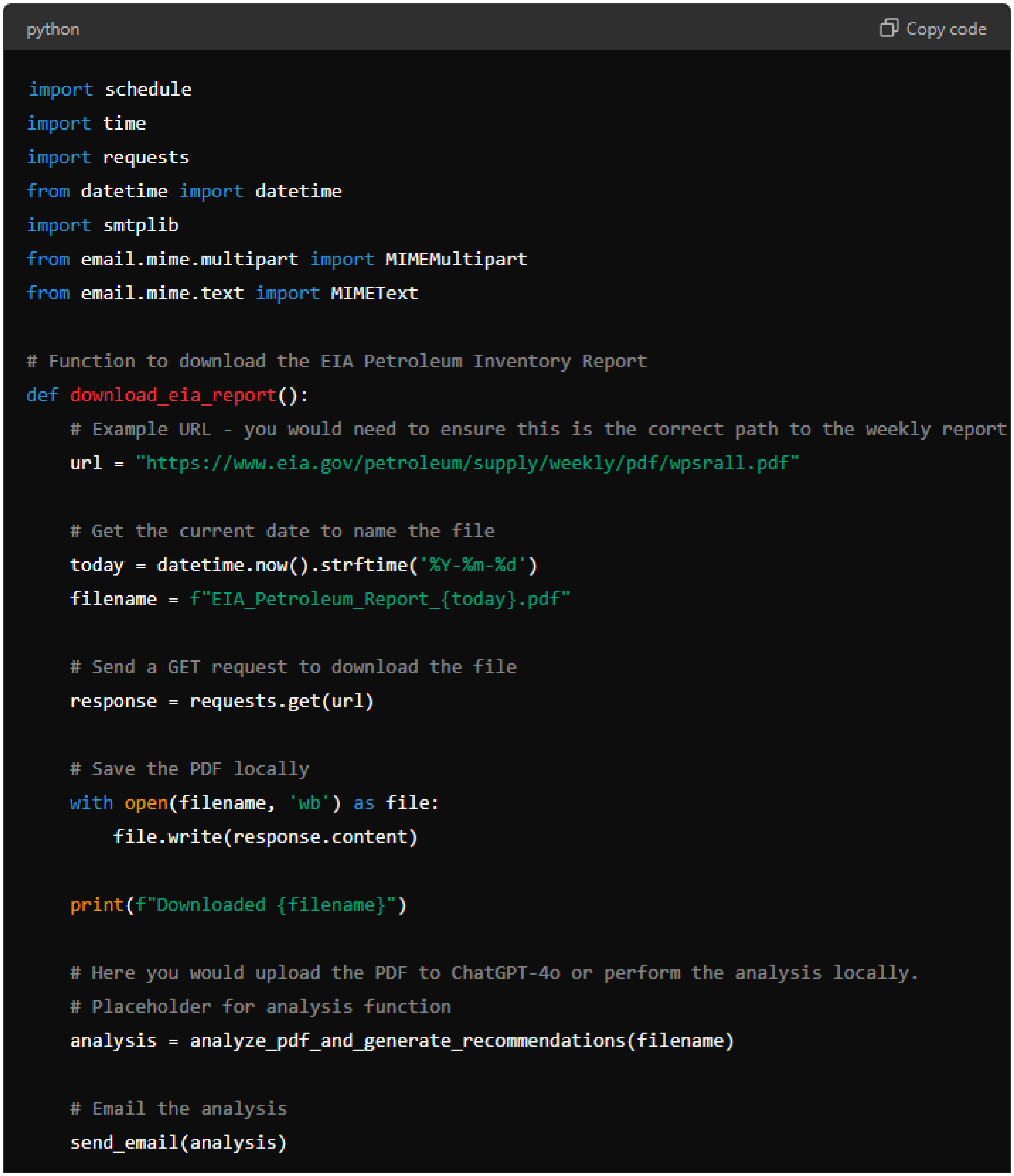
Using Python, please write a script that will automatically download the weekly EIA Petroleum Inventory Report every Wednesday at precisely 10:30 AM EST. After downloading the report in PDF format, upload the PDF into ChatGPT-4o and write an analysis of the data and make a short and long recommendation. After which, email your analysis to willpape@pennquartercapital.com.

## Creating a Python script to automatically download the weekly EIA Petroleum Inventory Report, analyze it, and send an email requires a combination of several tasks:

1. Scheduling the Script: Using the schedule or APScheduler library in Python to run the script every Wednesday at 10:30 AM EST.
2. Downloading the Report: Using requests or urllib to download the PDF from the EIA website.
3. Uploading the PDF to ChatGPT-4o: This step is theoretical as I can't directly interact with external systems or upload files, but I'll outline how you might use the OpenAI API (or similar service) for analysis.
4. Generating an Analysis: Analyzing the PDF content.
5. Emailing the Analysis: Using smtplib or similar to send an email with the analysis.

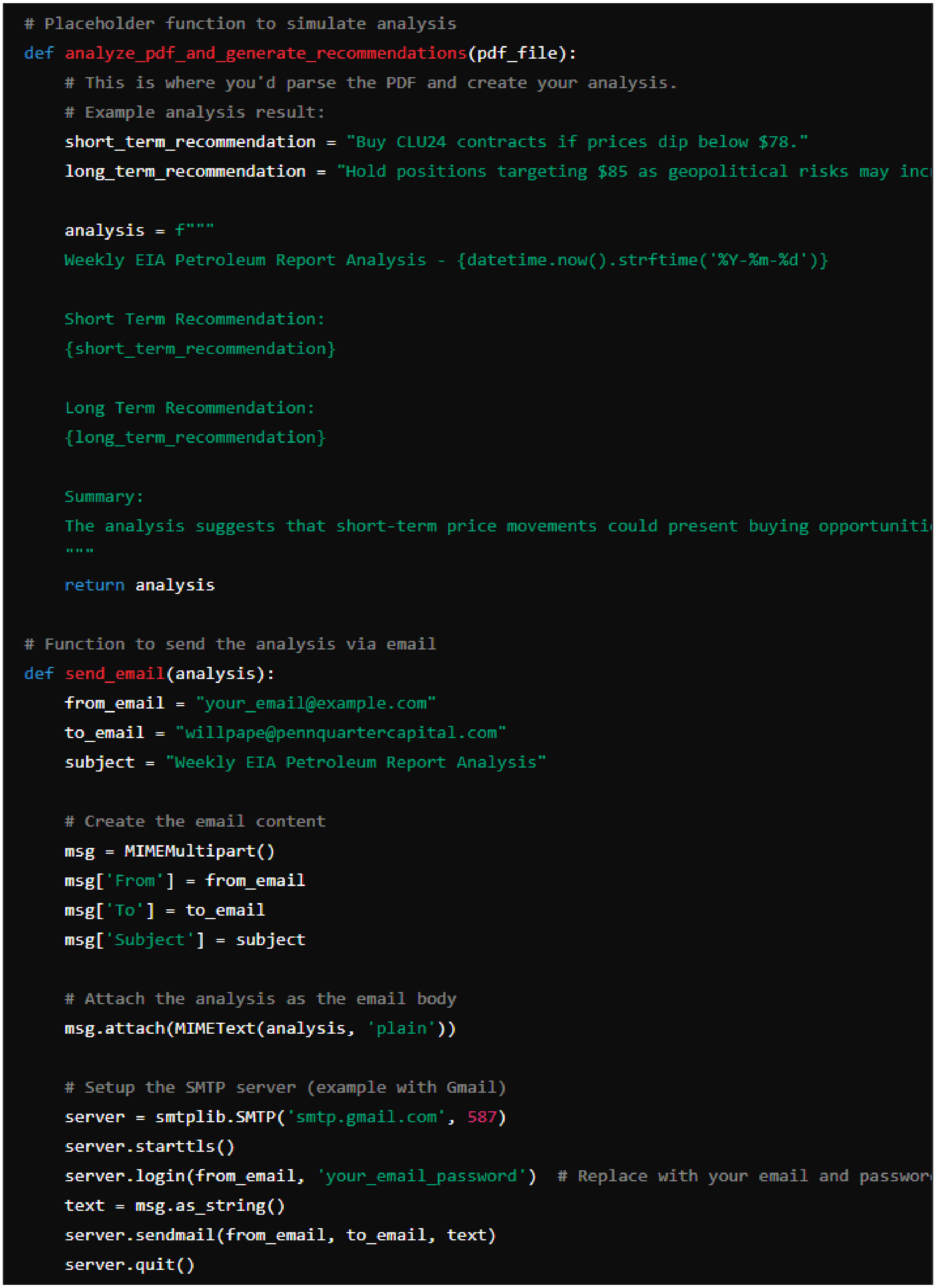
## Below is a Python script that covers these tasks

(note that it is hypothetical, as actual interaction with ChatGPT-4o and downloading specific files from the EIA site would require proper API access and file locations):



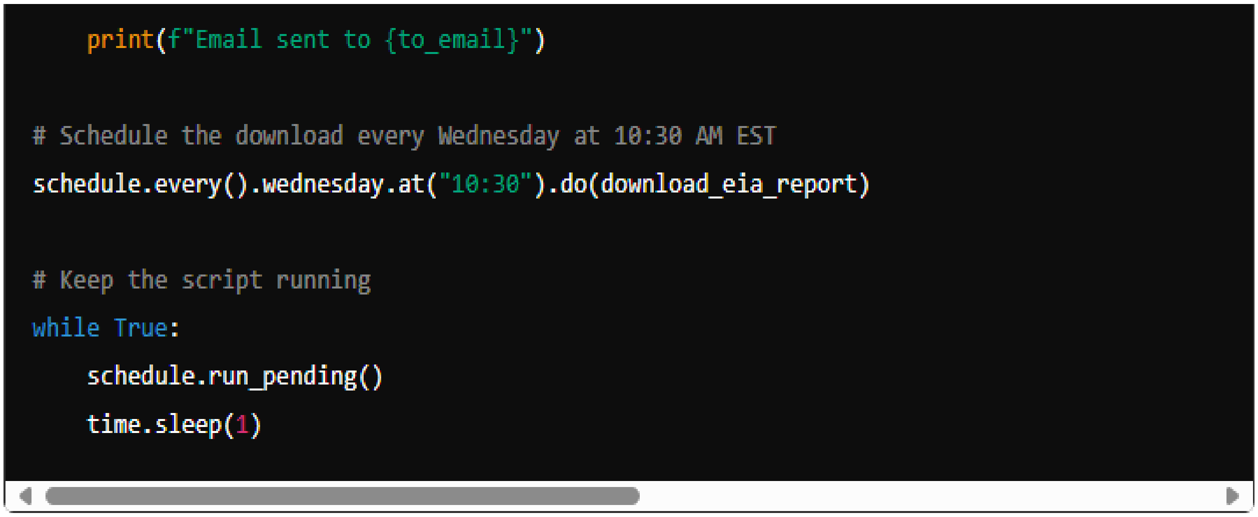
```python
import schedule
import time
import requests
from datetime import datetime
import smtplib
from email.mime.multipart import MIMEMultipart
from email.mime.text import MIMEText

# Function to download the EIA Petroleum Inventory Report
def download_eia_report():
    # Example URL - you would need to ensure this is the correct path to the weekly report
    url = "https://www.eia.gov/petroleum/supply/weekly/pdf/wpsrall.pdf"

    # Get the current date to name the file
    today = datetime.now().strftime('%Y-%m-%d')
    filename = f"EIA_Petroleum_Report_{today}.pdf"

    # Send a GET request to download the file
    response = requests.get(url)

    # Save the PDF locally
    with open(filename, 'wb') as file:
        file.write(response.content)

    print(f"Downloaded {filename}")

    # Here you would upload the PDF to ChatGPT-4o or perform the analysis locally.
    # Placeholder for analysis function
    analysis = analyze_pdf_and_generate_recommendations(filename)

    # Email the analysis
    send_email(analysis)
```

*Figure 1:Python Script - Import Libraries*



```python
# Placeholder function to simulate analysis
def analyze_pdf_and_generate_recommendations(pdf_file):
    # This is where you'd parse the PDF and create your analysis.
    # Example analysis result:
    short_term_recommendation = "Buy CLU24 contracts if prices dip below $78."
    long_term_recommendation = "Hold positions targeting $85 as geopolitical risks may inc

    analysis = f"""
    Weekly EIA Petroleum Report Analysis - {datetime.now().strftime('%Y-%m-%d')}

    Short Term Recommendation:
    {short_term_recommendation}

    Long Term Recommendation:
    {long_term_recommendation}

    Summary:
    The analysis suggests that short-term price movements could present buying opportuniti
    """
    return analysis

# Function to send the analysis via email
def send_email(analysis):
    from_email = "your_email@example.com"
    to_email = "willpape@pennquartercapital.com"
    subject = "Weekly EIA Petroleum Report Analysis"

    # Create the email content
    msg = MIMEMultipart()
    msg['From'] = from_email
    msg['To'] = to_email
    msg['Subject'] = subject

    # Attach the analysis as the email body
    msg.attach(MIMEText(analysis, 'plain'))

    # Setup the SMTP server (example with Gmail)
    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.starttls()
    server.login(from_email, 'your_email_password')  # Replace with your email and passwor
    text = msg.as_string()
    server.sendmail(from_email, to_email, text)
    server.quit()
```

*Figure 2:Python Script - Setup the Analysis*



```
    print(f"Email sent to {to_email}")

# Schedule the download every Wednesday at 10:30 AM EST
schedule.every().wednesday.at("10:30").do(download_eia_report)

# Keep the script running
while True:
    schedule.run_pending()
    time.sleep(1)
```

*Figure 3:Python Script - Run Script*

## Python Script

Copy code

```
# import libraries
import schedule

import time

import requests

from datetime import datetime

import smtplib

from email.mime.multipart import MIMEMultipart

from email.mime.text import MIMEText

# Function to download the EIA Petroleum Inventory Report
def download_eia_report():

    # Example URL- you would need to ensure this is the correct path to the weekly report
    url = "https://www.eia.gov/petroleum/supply/weekly/pdf/wpsrall.pdf"

        # Get the current date to name the file
    today = datetime.now().strftime('%Y-%m-%d')

    filename = f"EIA_Petroleum_Report_{today}.pdf"

    # Send a GET request to download the file
    response = requests.get(url)
```

```
      # Save the PDF locally
    with open(filename, 'wb') as file:
        file.write(response.content)

    print(f"Downloaded {filename}")

    # Here you would upload the PDF to ChatGPT-4o or perform the analysis locally.
    # Placeholder for analysis function
    analysis = analyze_pdf_and_generate_recommendations(filename)
      # Email the analysis
    send_email(analysis)
# Placeholder function to simulate analysis
def analyze_pdf_and_generate_recommendations(pdf_file):
    # This is where you'd parse the PDF and create your analysis.
    # Example analysis result:
    short_term_recommendation = "Buy CLU24 contracts if prices dip below $78."
    long_term_recommendation = "Hold positions targeting $85 as geopolitical risks may increase."

    analysis = f"""
    Weekly EIA Petroleum Report Analysis - {datetime.now().strftime('%Y-%m-%d')}

    Short Term Recommendation:
    {short_term_recommendation}

    Long Term Recommendation:
    {long_term_recommendation}

    Summary:
    The analysis suggests that short-term price movements could present buying opportunities, while the long-term outlook remains bullish due to potential geopolitical tensions.
    """
    return analysis

# Function to send the analysis via email
def send_email(analysis):
    from_email = "your_email@example.com"
    to_email = "willpape@pennquartercapital.com"
    subject = "Weekly EIA Petroleum Report Analysis"

    # Create the email content
    msg = MIMEMultipart()
    msg['From'] = from_email
    msg['To'] = to_email
    msg['Subject'] = subject

    # Attach the analysis as the email body
    msg.attach(MIMEText(analysis, 'plain'))

    # Setup the SMTP server (example with Gmail)
    server = smtplib.SMTP('smtp.gmail.com', 587)
    server.starttls()
    server.login(from_email, 'your_email_password')  # Replace with your email and password
    text = msg.as_string()
    server.sendmail(from_email, to_email, text)
    server.quit()

    print(f"Email sent to {to_email}")

# Schedule the download every Wednesday at 10:30 AM EST
schedule.every().wednesday.at("10:30").do(download_eia_report)
```

```
# Keep the script running
while True:
  schedule.run_pending()
  time.sleep(1)
```

## Conclusion

**Key Points:**

- **Scheduling**: The script uses schedule to trigger the download every Wednesday at 10:30 AM EST.
- **Downloading the Report**: The script sends a GET request to download the PDF from the EIA website.
- **Analyzing the PDF**: This is simulated, as it would require complex text extraction and analysis, ideally with tools like PyPDF2, PDFMiner, or interacting with an AI model.
- **Emailing the Analysis**: The script uses smtplib to send the analysis to your email.

**Dependencies:**

- schedule: For scheduling the download task.
- requests: For downloading the PDF.
- smtplib and email.mime: For sending emails.

**Instructions:**

1. Replace "your_email@example.com" and "your_email_password" with your email credentials.
2. Ensure the EIA PDF URL is correctly specified.
3. Implement actual PDF parsing and interaction with your preferred AI model for detailed analysis.

This script, when properly configured, will automate the process of downloading, analyzing, and communicating your findings each week.