**GROUP WORK PROJECT #** 1 MScFE 660: RISK MANAGEMENT
**GROUP NUMBER:** 3986

| FULL LEGAL NAME | LOCATION (COUNTRY) | EMAIL ADDRESS | MARK X FOR ANY NON-CONTRIBUTING MEMBER |
|---|---|---|---|
| Harshil Sumra | India | harshilsumra19972gmail.com | |
| Gaurav Srivastava | India | gauravsriitk@gmail.com | |
| Sunil Kumar Sharma | USA | sunilksh+mscfe@outlook.com | |

**Statement of integrity:** By typing the names of all group members in the text boxes below, you confirm that the assignment submitted is original work produced by the group (excluding any non-contributing members identified with an "X" above).

| | |
|---|---|
| **Team member 1** | **Harshil Sumra** |
| **Team member 2** | **Gaurav Srivastava** |
| **Team member 3** | **Sunil Kumar Sharma** |

Use the box below to explain any attempts to reach out to a non-contributing member. Type (N/A) if all members contributed.
**Note:** You may be required to provide proof of your outreach to non-contributing members upon request.

## 0. Answer guide

| **Step No** | **Location in the paper** |
|---|---|
| **1** | **Done individually** |
| **2** | **Introduction and Theoretical Set-up** |
| **3** | **In code file and Data description** |
| **4** | **Data description** |
| **5** | **Logic of Data Cleaning** |
| **6** | **Logic of Data Cleaning** |
| **7** | **Introduction and Theoretical Set-up** |
| **8** | **Conclusions** |
| **9** | **Introduction and Theoretical Set-up** |
| **10** | **Pseudo-code for inferred Causality** |

## 1. Introduction and Theoretical Set-up

This report is first of the three parts of the mini capstone project as part M.SC FE 660 risk management course. In this submission, we aim to understand the nuances of the suggested reading "Application of Probabilistic Graphical Models in Forecasting Crude Oil Price" by Danish A. Alvi [2018].

In this paper, the author is trying to utilize probabilistic graphical models (PGMs) in order define the structure and then use that to do crude oil price forecasting. More precisely, the model would encompass the entire energy market dynamics. There are several key challenges that are overcome by using PGM's which are addressed in this paper, namely:

a. It allows us to bypass the underlying assumptions of GARCH models.
b. It allows the author to bypass the hassle of understanding the vast data and doing precise data preprocessing. The PGM's allow for a much more unconstrained approach even though giving highly erroneous prediction. This issue was bypassed by following a causal scheme to prioritize linkages/impacts of different variables. The thesis proposes using structural learning algorithms like hill climbing search to learn the network structure directly from the data.
c. Utilization of hidden Markov models to identify different regimes of the time series data allowed for better PGM's being generated.

To understand the thesis better we need to understand the basics of PGM's, and the pros and cons associated with such a model. PGM's are basically noncyclic graphical representation of decision-making problem statements which allows study the relationship between different factors and how they impact the primary decision in focus in a structured manner. In this case, we'll be focusing on "Bayesian decision networks.", Their defining characteristic being utilization of basic conditional probability theory in well directed noncyclic graphs which is quite helpful in case of cause-and-effect relationship studies, which is the focus of the paper. This approach can also be quite helpful in applications where we want to do an economy's linkage analysis between it's different sectors.

Pros-

a. Modularity: Variables are represented as nodes, making it easier to modularize and update models.
b. Efficient Inference: PGMs allow for efficient computation of probabilistic inferences, such as computing the probability distribution of a set of variables given evidence.

c. Structured Representation: The graph structure reflects the relationships between variables, making it easier to understand and interpret complex dependencies.
d. Handling Uncertainty: PGMs are well-suited for representing and reasoning about uncertainty, which is common in real-world problems.

Cons-

a. Complex models which are not easy to infer, especially with vast number of variables as input.
b. Continuous Variables: are required to be converted to discrete variables.
c. Model/Structure selection- could end up being the key to do effective modeling which does not a well-defined approach.

Due to the solution/research utilizing large amount of data sets, the pros dominate the cons making this model being a good fit here as it would allow us to understand causal relationship opening the way towards more narrow studies based on the importance of variables.

Important Concepts:

a. Probabilistic graphical models (PGM) are frameworks for representing and reasoning about uncertainty in complex systems. They combine principles from probability theory and graph theory to model the relationships/dependencies among a set of random variables.

   In a probabilistic graphical model: Nodes represent random variables, which can take on various values with associated probabilities. Edges between nodes indicate probabilistic dependencies or relationships. These edges express how the variables influence each other. The graph structure defines the joint probability distribution over all variables and thus enables efficient inference and reasoning. PGMs can be classified into two main types: Bayesian Networks (Belief Networks) use directed acyclic graphs to represent causal relationships, while Markov Networks use undirected graphs to represent pairwise dependencies.

| **Criteria** | **Belief Networks (Bayesian Networks)** | **Markov Networks (Markov Random Fields)** |
|---|---|---|
| **Graph Structure** | Directed Acyclic Graph (DAG) | Undirected Graph (No specific order) |
| **Node Interpretation** | Nodes represent random variables | Nodes represent factors or cliques |
| **Graph Structure** | Contains directed edges | Contains undirected edges |

| **Criteria** | **Belief Networks (Bayesian Networks)** | **Markov Networks (Markov Random Fields)** |
|---|---|---|
| **Edge Semantics** | Uses conditional probabilities | Uses potential functions |
| **Propagation** | Often used for inference | Often used for modeling |
| **Cycle** | Must be acyclic | Cycles are allowed |
| **Dependency Type** | Represent conditional dependencies | Represent pairwise dependencies |
| **Edge Interpretation** | Represents direct causality | Represents statistical dependence |
| **Causality Representation** | Good for modeling causality | Focuses more on pairwise dependencies |
| **Global vs. Local** | Captures both global and local dependencies | Primarily captures local dependencies |
| **Inference** | Efficient for exact inference (e.g., Variable Elimination) | Generally, requires approximate methods (e.g., Markov Chain Monte Carlo) |
| **Parameterization** | Requires specifying conditional probabilities | Requires specifying factors and potentials |

b. Parameter Learning – also known as parameter estimation, involves determining the numerical values of the parameters (conditional probability distributions) associated with the nodes in a graphical model. These parameters represent the probabilities of various outcomes given the values of the parent nodes in Bayesian networks. Parameter learning is done based on observed data to fit the model to the actual distribution of the data.
For example, consider a Bayesian network with two variables: "Rain" and "good crop yield." The conditional probability distribution for " good crop yield " might depend on whether it's raining or not (parent variable "Rain") in case of state like Rajasthan in India. Parameter learning involves estimating the probabilities of " good crop yield " occurring given the different values of "Rain".

| Parameter learning VS Structured learning | | |
|---|---|---|
| | Parameter learning | Structured learning |
| Purpose | Fitting the model's parameters to the observed data to match the distribution of the data. | Discovering the relationships and dependencies between variables to establish the graphical structure of the model. |
| Focus | Estimating the probabilities associated with each node's conditional probability distribution. | Determining the presence or absence of edges (dependencies) between nodes. |

| | | |
|---|---|---|
| Task | Assigning specific values to existing parameters | Determining the graphical connections between nodes |
| Data usage | Uses observed data to estimate the probabilities for existing relationships. | Uses observed data, but its primary goal is to uncover the causal or dependency relationships. |

c. Markov chains and Markov blankets

c.1. Markov Chains:
A Markov chain is a mathematical concept used to model systems that transition from one state to another in a probabilistic manner. The fundamental property of a Markov chain is that the future state of the system depends solely on its current state and not on its previous history. This property is known as the Markov property or memory lessness.

A Markov chain consists of a set of states and a transition matrix that defines the probabilities of moving from one state to another. The transition probabilities are usually represented in a square matrix, where each row corresponds to a current state, and the values in each row represent the probabilities of transitioning to other states.

Markov chains are widely used in various fields, including physics, economics, biology, and computer science, to model processes that exhibit randomness and transitions over time. They are particularly useful for predicting future states of a system based on its current state and understanding the long-term behavior of the system.

c.2. Markov Blankets:
In the context of Bayesian networks and probabilistic graphical models, the Markov blanket of a node is the set of nodes that are considered to be its immediate neighbors, providing all the information needed to make the node conditionally independent of the rest of the nodes in the network. The Markov blanket of a node shields it from the influence of other nodes.

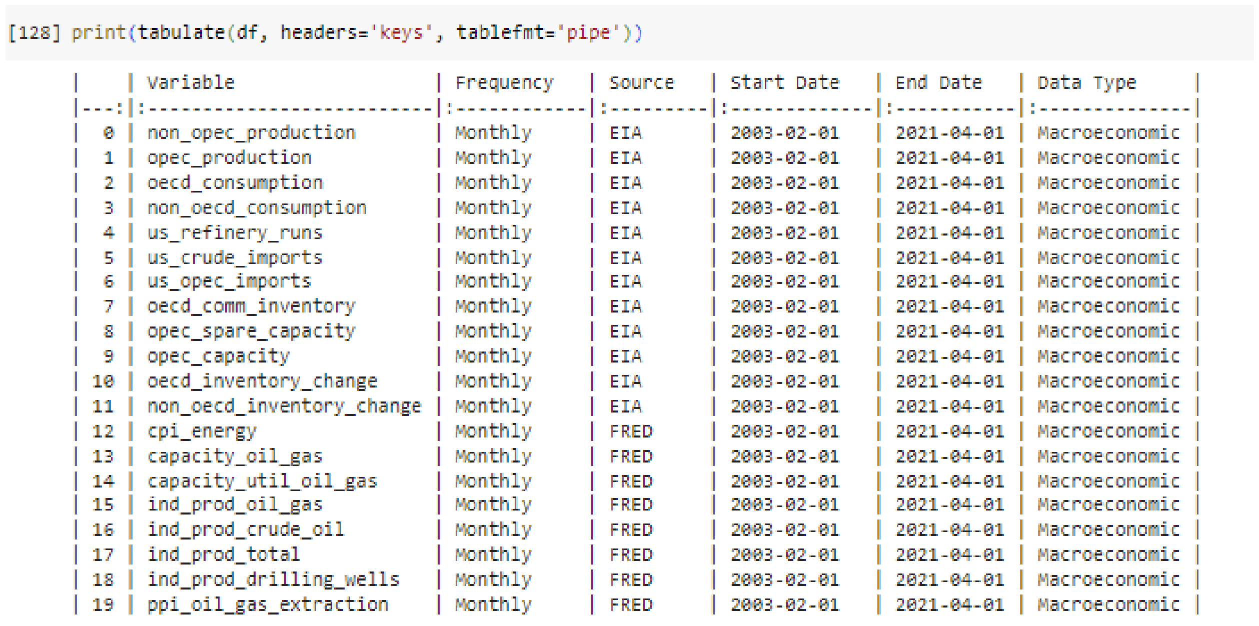
Mathematically, the Markov blanket of a node A consists of three sets:

Parents of A: Nodes that directly influence node A.
Children of A: Nodes that are directly influenced by node A.
Parents of Children of A: Nodes that are parents of nodes influenced by node A.
The Markov blanket concept is important in probabilistic graphical models because it helps simplify calculations of conditional probabilities and influences the structure of the graphical model. Nodes outside the Markov blanket of a node are irrelevant for making predictions or inferences about that node, given the nodes inside the Markov blanket.

Markov blankets play a crucial role in determining conditional independence relationships within a Bayesian network and guide the construction of efficient inference algorithms, such as belief propagation and variable elimination. They help identify the minimal set of variables required for reasoning about a specific variable of interest.

**Data description**

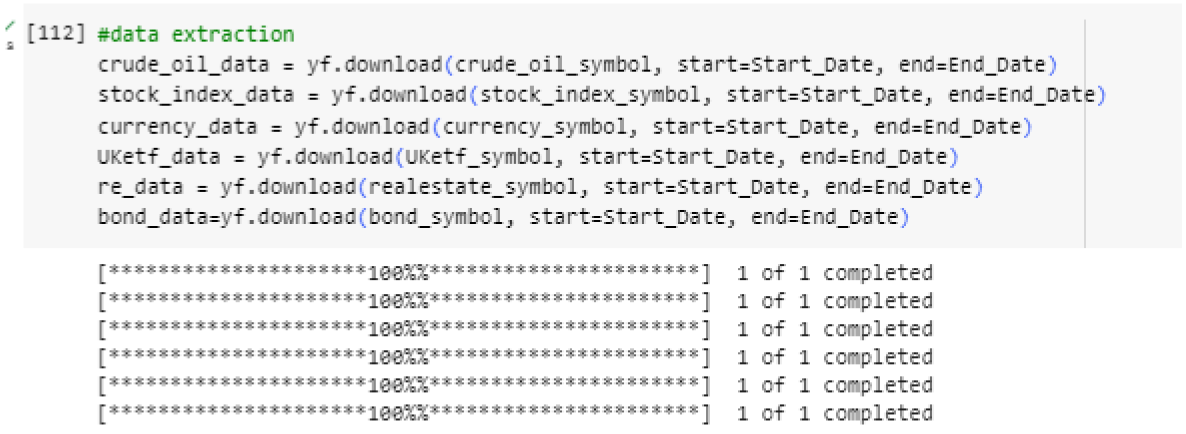
**a. Macroeconomic data (from – FRED and EIA)**



```
[128] print(tabulate(df, headers='keys', tablefmt='pipe'))
```

```
|    | Variable                   | Frequency   | Source   | Start Date   | End Date   | Data Type     |
|---:|:---------------------------|:------------|:---------|:-------------|:-----------|:--------------|
|  0 | non_opec_production        | Monthly     | EIA      | 2003-02-01   | 2021-04-01 | Macroeconomic |
|  1 | opec_production            | Monthly     | EIA      | 2003-02-01   | 2021-04-01 | Macroeconomic |
|  2 | oecd_consumption           | Monthly     | EIA      | 2003-02-01   | 2021-04-01 | Macroeconomic |
|  3 | non_oecd_consumption       | Monthly     | EIA      | 2003-02-01   | 2021-04-01 | Macroeconomic |
|  4 | us_refinery_runs           | Monthly     | EIA      | 2003-02-01   | 2021-04-01 | Macroeconomic |
|  5 | us_crude_imports           | Monthly     | EIA      | 2003-02-01   | 2021-04-01 | Macroeconomic |
|  6 | us_opec_imports            | Monthly     | EIA      | 2003-02-01   | 2021-04-01 | Macroeconomic |
|  7 | oecd_comm_inventory        | Monthly     | EIA      | 2003-02-01   | 2021-04-01 | Macroeconomic |
|  8 | opec_spare_capacity        | Monthly     | EIA      | 2003-02-01   | 2021-04-01 | Macroeconomic |
|  9 | opec_capacity              | Monthly     | EIA      | 2003-02-01   | 2021-04-01 | Macroeconomic |
| 10 | oecd_inventory_change      | Monthly     | EIA      | 2003-02-01   | 2021-04-01 | Macroeconomic |
| 11 | non_oecd_inventory_change  | Monthly     | EIA      | 2003-02-01   | 2021-04-01 | Macroeconomic |
| 12 | cpi_energy                 | Monthly     | FRED     | 2003-02-01   | 2021-04-01 | Macroeconomic |
| 13 | capacity_oil_gas           | Monthly     | FRED     | 2003-02-01   | 2021-04-01 | Macroeconomic |
| 14 | capacity_util_oil_gas      | Monthly     | FRED     | 2003-02-01   | 2021-04-01 | Macroeconomic |
| 15 | ind_prod_oil_gas           | Monthly     | FRED     | 2003-02-01   | 2021-04-01 | Macroeconomic |
| 16 | ind_prod_crude_oil         | Monthly     | FRED     | 2003-02-01   | 2021-04-01 | Macroeconomic |
| 17 | ind_prod_total             | Monthly     | FRED     | 2003-02-01   | 2021-04-01 | Macroeconomic |
| 18 | ind_prod_drilling_wells    | Monthly     | FRED     | 2003-02-01   | 2021-04-01 | Macroeconomic |
| 19 | ppi_oil_gas_extraction     | Monthly     | FRED     | 2003-02-01   | 2021-04-01 | Macroeconomic |
```

**b. Microeconomic data (from yahoo finance daily frequency)**



```
[112] #data extraction
      crude_oil_data = yf.download(crude_oil_symbol, start=Start_Date, end=End_Date)
      stock_index_data = yf.download(stock_index_symbol, start=Start_Date, end=End_Date)
      currency_data = yf.download(currency_symbol, start=Start_Date, end=End_Date)
      UKetf_data = yf.download(UKetf_symbol, start=Start_Date, end=End_Date)
      re_data = yf.download(realestate_symbol, start=Start_Date, end=End_Date)
      bond_data=yf.download(bond_symbol, start=Start_Date, end=End_Date)
```

```
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
```

**c. Financial context data (from FRED)**
**For financial data selection, we have opted for the following key indicators:**

**Non-Farm Payroll (US) - Represented by the code PAYEMS, these metric gauges the total number of non-farm workers in the United States. It is measured monthly and can be accessed at the following link: Non-Farm Payroll Data.**

**US Unemployment Rate - Tracked by the code UNRATE, the US Unemployment Rate is a monthly series compiled by the Bureau of Labor Statistics (BLS). It is calculated as the ratio of unemployed individuals to the total labor force in the United States. You can find this data here: US Unemployment Rate Data.**

**US Core CPI (Excluding Food/Energy) - Identified as USACPICORMINMEI, this data series is provided by the OECD and represents the Consumer Price Index (CPI) for all items in the United States, excluding food and energy components. It is reported on a monthly basis and can be accessed via the following link: US Core CPI Data.**

**These indicators are valuable for assessing the economic health and stability of the United States, providing insights into labor market dynamics and inflation trends.**

**Logic of Data Cleaning**

| Data issue | Macroeconomic data | Microeconomic data | Financial context data |
|---|---|---|---|
| Extreme Outlier | z-score approach with threshold at 3 and replace with median value. | z-score approach with threshold at 3 and replace with mean value. | z-score approach with threshold at 3 and replace with mean value. |
| Bad data | Not required as the API helped us deal with these issues. | Not required as the API helped us deal with these issues. | Not required as the API helped us deal with these issues. |
| Missing value | Drop the rows with missing values | Drop the rows with missing values | Drop the rows with missing values |

**Looking at the outlier data, the outliers are sporadic across different columns (and only 18 rows out of 219 entries) and are especially concentrated in particular years (e.g., 2003, 2008, 2020).**

**[All the graphs can be found in the code file]**
**Pseudo-code for inferred Causality**

Next intriguing question being, how exactly does Structured learning and Markov networks used to infer causality to determine the graphical relationship? There are many approaches which can be used to quantify. The causal relationship between two variables, like Engle Granger Causality. Here is an algorithm which can help do the same, called "Inferred Causality" algorithm.

**Logic: -**

**Inputs:**

1. dataset: A dataset containing variables Xi, where i ranges from 1 to m.

**Outputs:**

1. G: A partially directed graphical model.

**Phase 1: Learning Markov Blankets**

1. Initialize an empty dictionary Markov blanket to store the Markov blankets of each variable.
2. For each variable Xi in the dataset, do the following:
    - Learn its Markov blanket B(Xi) (a set of variables that are probabilistically dependent on Xi).
    - Store B(Xi) in the Markov blankets dictionary with Xi as the key.
3. Check if the Markov blankets are symmetric, meaning if Xi is in the Markov blanket of Xj, then Xj should also be in the Markov blanket of Xi. Remove any asymmetric relationships as false positives.
4. Return the symmetric Markov blankets.

**Phase 2: Learning Neighbors**

1. Initialize an empty dictionary partial_dag to represent the partially directed acyclic graph (DAG).
2. For each pair of variables Xi and Xj in the symmetric Markov blankets:
   - Attempt to find a set SXiXj such that Xi is conditionally independent of Xj given SXiXj, and neither Xi nor Xj are in SXiXj.
   - If SXiXj is not found, place an undirected arc between Xi and Xj in partial_dag.
3. Check if the neighbors in partial_dag are symmetric and correct any asymmetries.

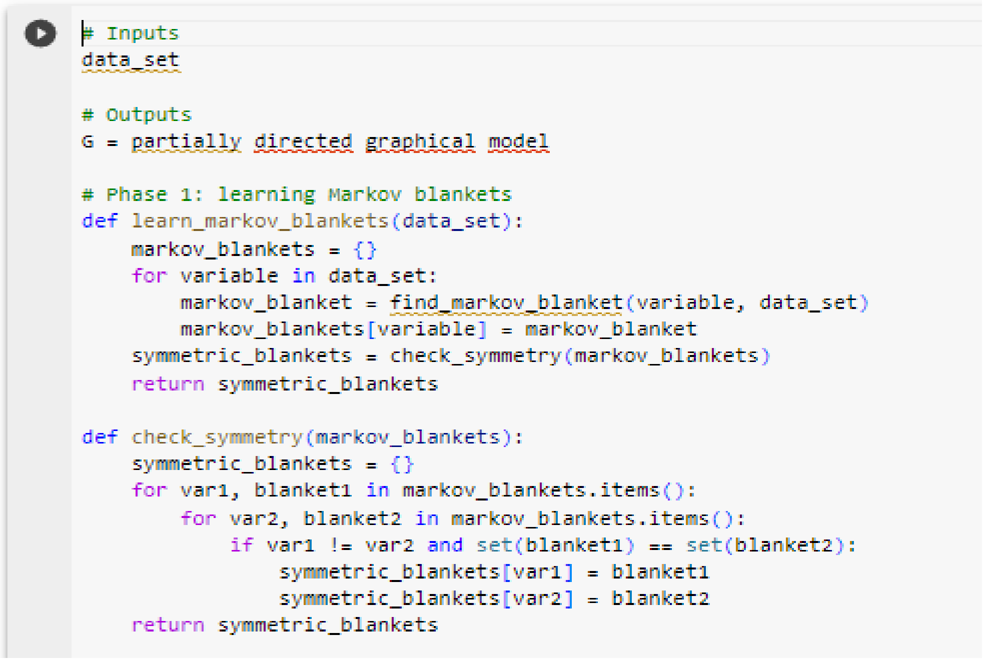
**Phase 3: Learning Arc Directions**

1. Create a copy of the symmetric DAG as completed_dag.
2. For each pair of non-adjacent variables Xi and Xj with a common neighbor Xk that is not in SXiXj: a. Set the directions of the arcs Xi → Xk and Xk → Xj to obtain a V-structure (Xi → Xk ← Xj).
3. Set the direction of the remaining undirected arcs using the following rules recursively:
   - If there exists a strictly directed path from Xi to Xj, set the direction as Xi → Xj.
   - If Xi and Xj are not adjacent and Xi → Xk and Xk → Xj, then set Xk → Xj.

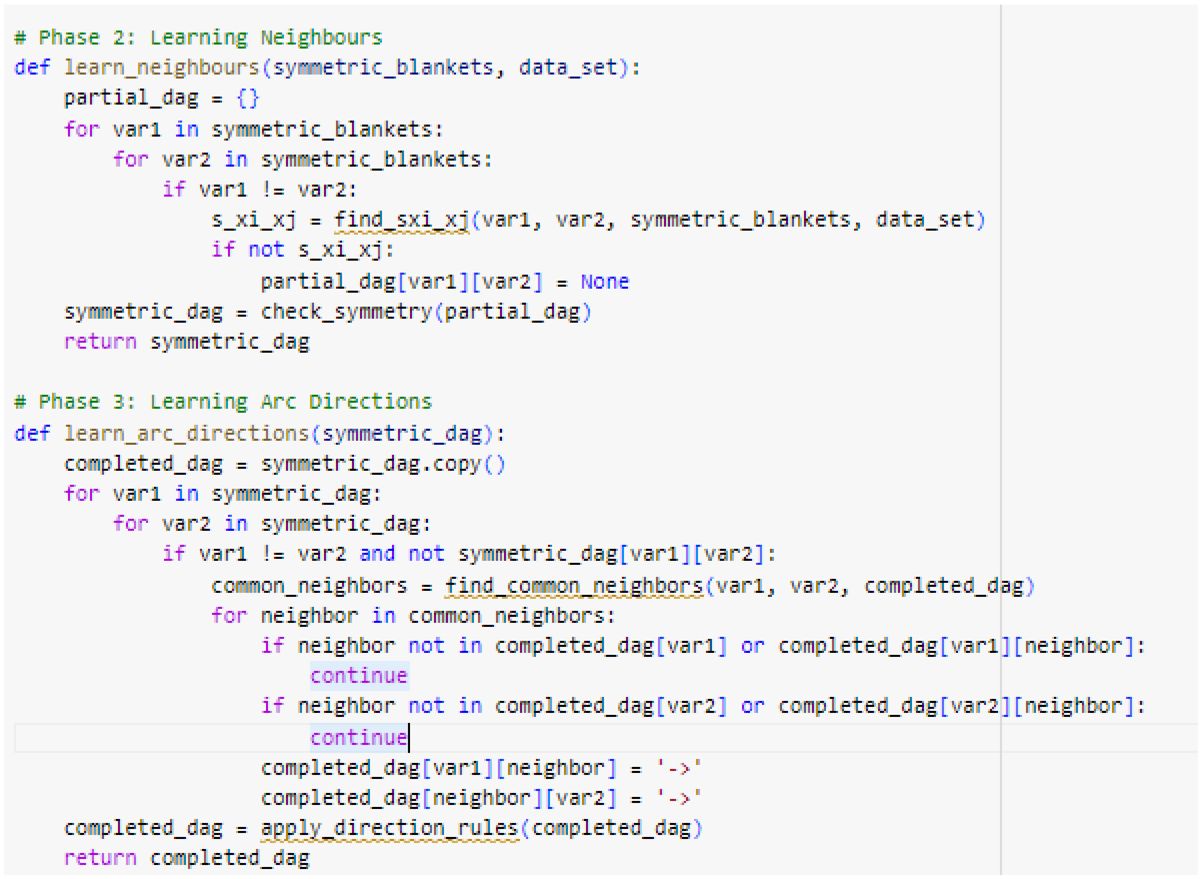
**Main Algorithm**

1. Execute the following steps in sequence:
   - Learn the symmetric Markov blankets for each variable.
   - Learn the neighbors based on the Markov blankets.
   - Learn the arc directions within the graph.
2. Return the completed partially directed graphical model G.

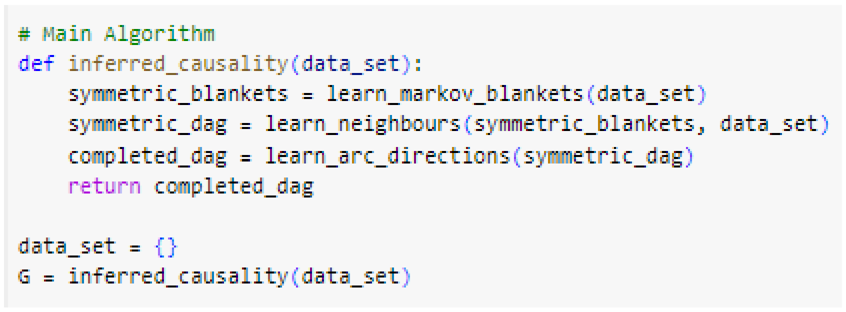
Pseudocode: -



```
# Inputs
data_set

# Outputs
G = partially directed graphical model

# Phase 1: learning Markov blankets
def learn_markov_blankets(data_set):
    markov_blankets = {}
    for variable in data_set:
        markov_blanket = find_markov_blanket(variable, data_set)
        markov_blankets[variable] = markov_blanket
    symmetric_blankets = check_symmetry(markov_blankets)
    return symmetric_blankets

def check_symmetry(markov_blankets):
    symmetric_blankets = {}
    for var1, blanket1 in markov_blankets.items():
        for var2, blanket2 in markov_blankets.items():
            if var1 != var2 and set(blanket1) == set(blanket2):
                symmetric_blankets[var1] = blanket1
                symmetric_blankets[var2] = blanket2
    return symmetric_blankets
```



```
# Phase 2: Learning Neighbours
def learn_neighbours(symmetric_blankets, data_set):
    partial_dag = {}
    for var1 in symmetric_blankets:
        for var2 in symmetric_blankets:
            if var1 != var2:
                s_xi_xj = find_sxi_xj(var1, var2, symmetric_blankets, data_set)
                if not s_xi_xj:
                    partial_dag[var1][var2] = None
    symmetric_dag = check_symmetry(partial_dag)
    return symmetric_dag

# Phase 3: Learning Arc Directions
def learn_arc_directions(symmetric_dag):
    completed_dag = symmetric_dag.copy()
    for var1 in symmetric_dag:
        for var2 in symmetric_dag:
            if var1 != var2 and not symmetric_dag[var1][var2]:
                common_neighbors = find_common_neighbors(var1, var2, completed_dag)
                for neighbor in common_neighbors:
                    if neighbor not in completed_dag[var1] or completed_dag[var1][neighbor]:
                        continue
                    if neighbor not in completed_dag[var2] or completed_dag[var2][neighbor]:
                        continue
                    completed_dag[var1][neighbor] = '->'
                    completed_dag[neighbor][var2] = '->'
    completed_dag = apply_direction_rules(completed_dag)
    return completed_dag
```



```
# Main Algorithm
def inferred_causality(data_set):
    symmetric_blankets = learn_markov_blankets(data_set)
    symmetric_dag = learn_neighbours(symmetric_blankets, data_set)
    completed_dag = learn_arc_directions(symmetric_dag)
    return completed_dag

data_set = {}
G = inferred_causality(data_set)
```

**Conclusions:**

- What makes oil prices look different from other asset prices? E.g., spikes, clustered volatility, seasonality, etc.

Oil prices exhibit several characteristics that distinguish them from other asset prices:

a) Sensitivity to Geopolitical Events: Oil prices are highly sensitive to geopolitical events, such as conflicts in oil-producing regions(Russia-Ukraine conflict), supply disruptions(due to weather constraints of local regulations), and changes in political regimes. These events can lead to sudden and significant price spikes.

b) OPEC Influence: The Organization of the Petroleum Exporting Countries (OPEC) plays a significant role in influencing oil prices through production quotas and decisions. OPEC's actions can lead to price fluctuations and volatility.

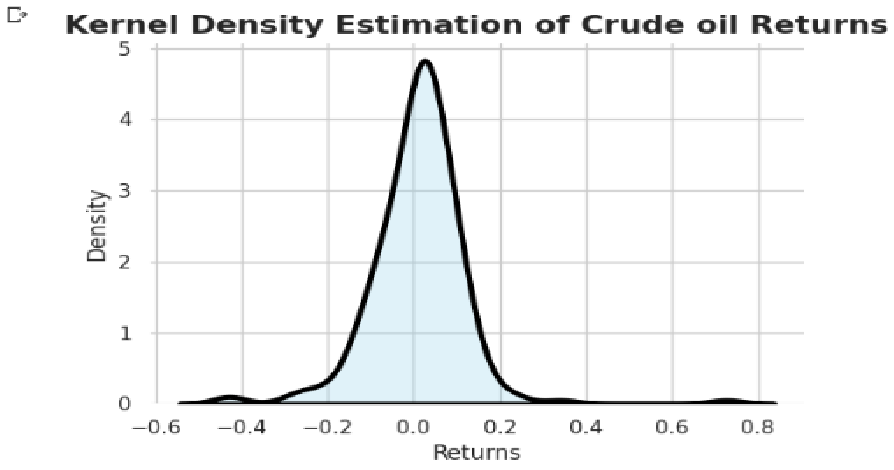
c) Seasonality: Oil prices often exhibit seasonality due to factors like weather conditions affecting demand for heating oil and gasoline. For example, oil prices tend to rise during the winter months when heating demand increases.

d) Volatility Clustering: Oil prices can experience periods of clustered volatility, where periods of high volatility are followed by relatively calmer periods. This clustering can be attributed to sudden market shocks and changing supply and demand dynamics.

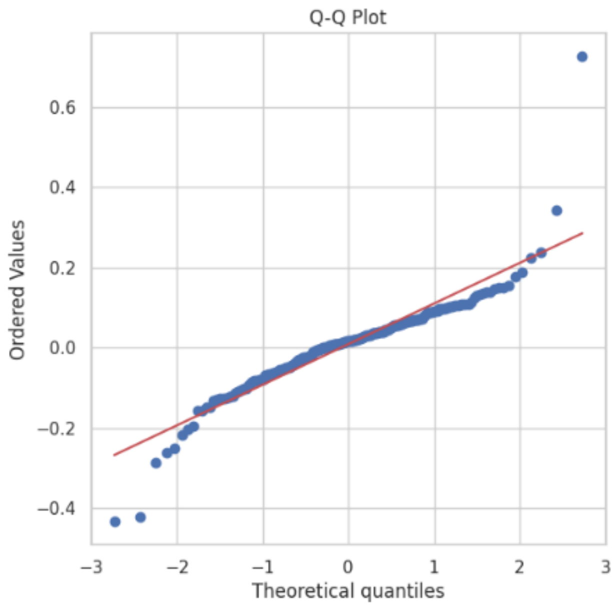
e) Currency Exchange Rates: Oil prices are typically quoted in U.S. dollars, so fluctuations in currency exchange rates can affect the purchasing power of oil-importing countries and, consequently, oil demand and prices.

f) Speculative Bubbles: Oil prices can experience speculative bubbles, where prices rise sharply and then collapse, often fueled by market sentiment and excessive speculation.

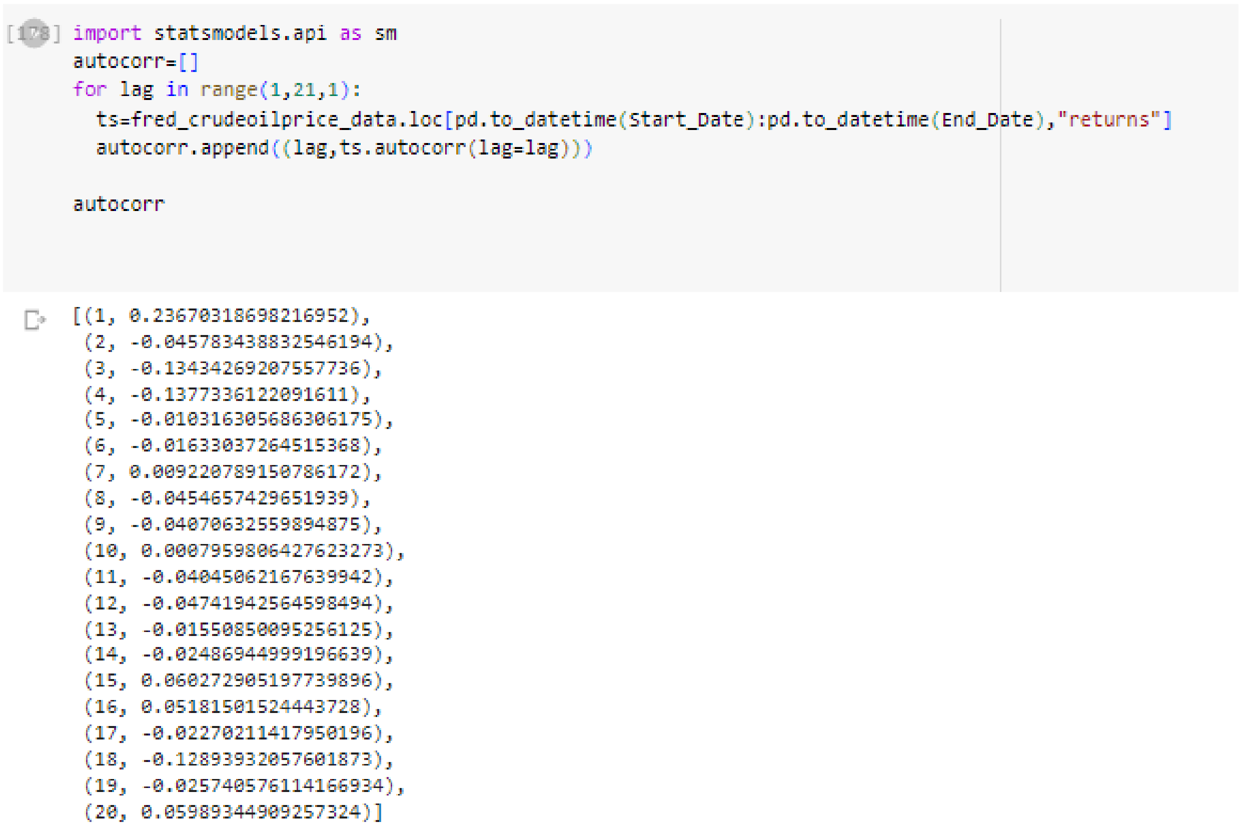
- What types of distributions do oil returns have?
  Based on the q-q plot and kernel density plot, we observe that oil returns are close to normal.



In Q-Q plot, we observe that the deviation con into effect past two standard deviations. So, I think that we can go ahead and use 2 standard deviation threshold in the next phase of the GWP project.



- What type of autocorrelation do the oil returns have?



```
[178] import statsmodels.api as sm
      autocorr=[]
      for lag in range(1,21,1):
        ts=fred_crudeoilprice_data.loc[pd.to_datetime(Start_Date):pd.to_datetime(End_Date),"returns"]
        autocorr.append((lag,ts.autocorr(lag=lag)))

      autocorr
```

```
[(1, 0.23670318698216952),
 (2, -0.045783438832546194),
 (3, -0.13434269207557736),
 (4, -0.1377336122091611),
 (5, -0.010316305686306175),
 (6, -0.01633037264515368),
 (7, 0.009220789150786172),
 (8, -0.0454657429651939),
 (9, -0.04070632559894875),
 (10, 0.0007959806427623273),
 (11, -0.04045062167639942),
 (12, -0.04741942564598494),
 (13, -0.01550850095256125),
 (14, -0.02486944999196639),
 (15, 0.06027290519773989б),
 (16, 0.05181501524443728),
 (17, -0.02270211417950196),
 (18, -0.12893932057601873),
 (19, -0.025740576114166934),
 (20, 0.05989344909257324)]
```

Therefore, maximum autocorrelation is for lag 1.

- What other stylized facts can you say about oil prices?

a) Correlation of ind_prod_drilling_wells with Crude oil prices: 0.6718779734679279 -> high correlation.

b) Correlation of ppi_oil_gas_extraction with Crude oil prices: 0.7447840118864673 -> high correlation.

**References:**

- **Alvi, Danish A. Application of Probabilistic Graphical Models in Forecasting Crude. Bachelors Thesis. London: University College London, 2018. Document. <https://arxiv.org/abs/1804.10869>.**
- **Groen, Jan J. Weaker Oil Prices: Demand, Supply or Neither. Presentation. New York: US Federal Reserve Bank of New York, 2016. <https://www.eia.gov/finance/markets/reports_presentations/2016JanGroen.pdf>**
- **Badel, Alejandro and Joseph McGillicuddy. Oil Prices and Inflation Expectations. St Louis: EIA, 2015. <https://www.stlouisfed.org/-/media/project/frbstl/stlouisfed/Publications/Regional-Economist/2015/July/Oil.pdf>.**