Algorithmic Trading Strategy: Statistical Arbitrage & Cointegration

Author : Cousquer Maxime
Date : 02/10/2026
Context : Personal project
Stack : Pyhton, Pandas, Numpy, Statsmodel, Seaborn, Yfinance

Development of a **Market Neutral** strategy based on Pairs Trading (Mean Reversion) applied to the DJ30 and CAC40 indices (2020-2024).
Unlike simple correlation, this project utilizes the **Engle-Granger Two-Step approach** to identify stationary spreads.

Key Achievements:

1. **Market Scanning:** Automated testing of over 1,000 pairs to map market efficiency.
2. **Data Cleaning:** Identification and exclusion of spurious correlations (data artifacts).
3. **Pair Selection:** Optimization of capital allocation by selecting **Chevron (CVX) vs Exxon Mobil (XOM)** based on Half-Life mean reversion, despite other pairs having better pure statistical scores.
4. **Risk Management:** Backtest revealing a **94% Hit Rate**, robust to standard volatility but highlighting vulnerabilities during structural breaks (COVID-19 crash).

1. Theoretical Framework: Why Cointegration?

Traditional correlation measures short-term directional movements. However, a high correlation does not imply a long-term equilibrium. To ensure the spread reverts to the mean, we test for **Cointegration**.

We model the relationship between asset Y and asset X as: $$Y_t = \alpha + \beta X_t + \epsilon_t$$
Where β is the Hedge ratio derivated from the OLS regression. The spread ϵ is then tested for stationarity using the **Augmented Dickey-Fuller test (ADF)**
$$\Delta \epsilon_t = \gamma \epsilon_{t-1} + \sum_{i=1}^{p} \delta_i \Delta \epsilon_{t-i} + u_t$$

-Null hypothesis : The spread as a unit root (Non-Stationary/Random walk)
-Alternative : The spread is Stationary (Mean reverting)

2. Data Integrity & Anomaly Detection

Before running the cointegration algorithm across the entire market, a strict data quality check was performed. Initial scans returned several "false positive" cointegrations with near-zero P-Values.

Visual diagnostics revealed these were **Data Artifacts** (e.g., API missing values, illiquidity, or stock splits causing flat price lines). These flat or broken lines artificially trick the Augmented Dickey-Fuller test into detecting perfect stationarity.

Strategic Decision: Anomalous tickers exhibiting these corrupt data patterns were strictly removed from the investment universe (Cleaned Universe) to prevent model poisoning.

In [ ]:

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import statsmodels.api as sm
from statsmodels.tsa.stattools import adfuller
import yfinance as yf

# --- 1. LA FONCTION DE SCANNER ---
def scanner_marche(nom_indice, liste_tickers):
    print(f"\n--- DÉMARRAGE DU SCANNER : {nom_indice} ---")

    # Téléchargement
    print(f"Téléchargement via Yahoo Finance...")
    try:
        data = yf.download(liste_tickers, start="2020-01-01", end="
        # Gestion du format MultiIndex de Yahoo (Close vs Adj Close
        if 'Adj Close' in data.columns:
            data = data['Adj Close']
        elif 'Close' in data.columns:
            data = data['Close']
        # Si c'est déjà un DataFrame simple
        elif 'Adj Close' in data.columns.get_level_values(0):
             data = data['Adj Close']

        data = data.dropna(axis=1, how='all').dropna() # Nettoyage
    except Exception as e:
        print(f"Erreur de téléchargement : {e}")
        return None

    tickers = data.columns
    n = len(tickers)
    print(f" Données prêtes : {n} actions à analyser.")
    # Matrice vide
    pvalue_matrix = pd.DataFrame(index=tickers, columns=tickers, dt

    # Boucle de calcul
    print("Calcul des cointégrations en cours...")
    for i in range(n):
        for j in range(i+1, n):
            stock_a = data.iloc[:, i]
            stock_b = data.iloc[:, j]

            try:
                model = sm.OLS(stock_a, sm.add_constant(stock_b))
                res = model.fit()
                beta = res.params.iloc[1]
                spread = stock_a - beta * stock_b
                p_val = adfuller(spread)[1]

                pvalue_matrix.iloc[i, j] = p_val
                pvalue_matrix.iloc[j, i] = p_val
            except:
                pvalue_matrix.iloc[i, j] = 1.0
                pvalue_matrix.iloc[j, i] = 1.0

    np.fill_diagonal(pvalue_matrix.values, 1.0)
    print(f"Analyse terminée pour {nom_indice}.")
    return pvalue_matrix

# --- 2. LES LISTES PROPRES  ---

# DOW JONES (Cleaned Universe)
tickers_dow_clean = [
    "AAPL", "MSFT", "JPM", "V", "PG", "WMT", "DIS", "HD", "JNJ", "K
    "MRK", "MCD", "CSCO", "VZ", "CRM", "NKE", "AXP", "INTC", "IBM",
    "HON", "CAT", "AMGN", "CVX", "MMM", "TRV", "UNH", "DOW", "AMZN"
]

# CAC 40 (Cleaned Universe)
tickers_cac_clean = [
    "AI.PA", "AIR.PA", "ALO.PA", "CS.PA", "BNP.PA", "EN.PA", "CAP.P
    "CA.PA", "ACA.PA", "BN.PA", "DSY.PA", "EDEN.PA", "EL.PA", "ERF.
    "KER.PA", "OR.PA", "LR.PA", "MC.PA", "ML.PA", "ORA.PA", "RI.PA"
    "RNO.PA", "SAF.PA", "SGO.PA", "SAN.PA", "SU.PA", "GLE.PA", "TTE
]

# --- 3. LANCEMENT DES DEUX SCANNERS ---

matrice_dow = scanner_marche("DOW JONES (Clean)", tickers_dow_clean
matrice_cac = scanner_marche("CAC 40 (Clean)", tickers_cac_clean)

# --- 4. VISUALISATION ---
if matrice_dow is not None and matrice_cac is not None:
    fig, ax = plt.subplots(1, 2, figsize=(24, 12))

    # Heatmap US
    sns.heatmap(matrice_dow, ax=ax[0], cmap="RdYlGn_r", vmin=0, vma
    ax[0].set_title("DOW JONES (US) - Nettoyé")
```

```
# Heatmap FR
sns.heatmap(matrice_cac, ax=ax[1], cmap="RdYlGn_r", vmin=0, vma
ax[1].set_title("CAC 40 (FRANCE) - Nettoyé")

plt.suptitle("Comparaison Finale : Structure des marchés US vs
plt.tight_layout()
plt.show()

# Stats
nb_pairs_dow = (matrice_dow < 0.05).sum().sum() / 2
nb_pairs_cac = (matrice_cac < 0.05).sum().sum() / 2
print(f"\ RÉSULTATS FINAUX :")
print(f" Dow Jones : {int(nb_pairs_dow)} paires valides.")
print(f" CAC 40    : {int(nb_pairs_cac)} paires valides.")
```

Liste des tickers

- DOW JONES (Sans Boeing BA, sans WBA) tickers_dow_clean = ["AAPL", "MSFT", "JPM", "V", "PG", "WMT", "DIS", "HD", "JNJ", "KO", "MRK", "MCD", "CSCO", "VZ", "CRM", "NKE", "AXP", "INTC", "IBM", "GS", "HON", "CAT", "AMGN", "CVX", "MMM", "TRV", "UNH", "DOW", "AMZN"]
- CAC 40 (Sans Thales HO.PA, ni Stellantis/Arcelor/STMicro et TPE/PA) tickers_cac_clean = ["AI.PA", "AIR.PA", "ALO.PA", "CS.PA", "BNP.PA", "EN.PA", "CAP.PA", "CA.PA", "ACA.PA", "BN.PA", "DSY.PA", "EDEN.PA", "EL.PA", "ERF.PA", "RMS.PA", "KER.PA", "OR.PA", "LR.PA", "MC.PA", "ML.PA", "ORA.PA", "RI.PA", "PUB.PA", "RNO.PA", "SAF.PA", "SGO.PA", "SAN.PA", "SU.PA", "GLE.PA", "TTE.PA", "URW.PA", "VIE.PA", "DG.PA", "VIV.PA", "WLN.PA"]

Data is fetched via Yahoo Finance API.
Pre-processing includes handling missing values and adjusting for stock splits.

3. Structural Market Analysis: US vs France (Macro View)

We computed the cointegration p-values for all pairs. The results are mapped below (Green = Cointegrated, $p < 0.05$).

Quantitative Observations:

- **US Market (Dow Jones):** High density of opportunities (**12.56%** of pairs are cointegrated). The US index exhibits deep sectoral homogeneity (e.g., multiple Oil giants, Tech giants), naturally fostering statistical arbitrage.

- **French Market (CAC 40):** Low density (**7.73%**). The index is highly fragmented. Finding pairs often requires cross-sector matching (e.g., Axa/Michelin), which carries a higher fundamental risk of correlation breakdown.

4. Pair Selection & Strategy Implementation

We compared two candidates: **Pepsi/Coca-Cola (PEP/KO)** and **Chevron/Exxon (CVX/XOM)**.

Metric	PEP / KO	CVX / XOM	Decision
Correlation	94.5%	**97.1%**	CVX/XOM is fundamentally cleaner (Pure Energy play).
Cointegration (P-Value)	**0.018**	0.033	PEP/KO is statistically stronger.
Half-Life (Mean Reversion)	~35 days	**~24 days**	CVX/XOM reverts faster.
Max Holding Period	110 days	**89 days**	CVX/XOM offers better capital efficiency.

Selected Pair: Chevron (CVX) vs Exxon Mobil (XOM). **Rationale:** We prioritize capital rotation and fundamental sector homogeneity over pure statistical significance.

In [ ]:

```
# --- SPREAD ANALYSIS: CHEVRON VS EXXON ---
pair_data = yf.download(["CVX", "XOM"], start="2020-01-01", end="20
stock_a = pair_data['CVX']
stock_b = pair_data['XOM']

# OLS Regression for Hedge Ratio
model = sm.OLS(stock_a, sm.add_constant(stock_b)).fit()
hedge_ratio = model.params.iloc[1]
spread = stock_a - hedge_ratio * stock_b

# Z-Score Calculation (Rolling window of 30 days for dynamic mean)
rolling_mean = spread.rolling(window=30).mean()
rolling_std = spread.rolling(window=30).std()
zscore = (spread - rolling_mean) / rolling_std

# Plot
plt.figure(figsize=(14, 5))
plt.plot(zscore.index, zscore, label="Z-Score (CVX/XOM)", color='bl
plt.axhline(2.0, color='red', linestyle='--', label="Short Spread (
plt.axhline(-2.0, color='green', linestyle='--', label="Long Spread
plt.axhline(0, color='black', label="Mean")
plt.title("Z-Score Dynamics: Mean Reversion Engine")
plt.legend()
plt.show()
```

5. Backtesting & Risk Management

Entry Rule : Z-Score > 2.0 (Short Spread) or Z-Score < -2.0 (Long Spread).
Exit Rule : Z-Score reverts to 0.
Stop Loss : Z-Score > 4.0 (Model Break).

Results:

- The strategy generated **34 signals** over 5 years.
- **Drawdown Analysis :** The only major divergence occurred in **March 2020 (COVID-19)**. This highlights the risk of "Fat Tail" events where historical correlations break down due to systemic exogenous shocks.

In [ ]:

```
import pandas as pd
import matplotlib.pyplot as plt

def analyser_stats_marche(matrice, nom_indice):
    # On ne regarde que le triangle supérieur pour ne pas compter l
    # et on exclut la diagonale
    mask = np.triu(np.ones(matrice.shape), k=1).astype(bool)
    valeurs = matrice.where(mask).stack() # On met tout dans une se

    total_pairs = len(valeurs)

    # 1. Paires Cointégrées (Standard) : P < 0.05
    nb_coint = (valeurs < 0.05).sum()
    pct_coint = (nb_coint / total_pairs) * 100

    # 2. Paires "Pépites" (Très fortes) : P < 0.01
    nb_strong = (valeurs < 0.01).sum()
    pct_strong = (nb_strong / total_pairs) * 100

    # 3. Paires "Espoir" (Presque bonnes) : 0.05 < P < 0.10
    nb_weak = ((valeurs >= 0.05) & (valeurs < 0.10)).sum()
    pct_weak = (nb_weak / total_pairs) * 100

    return {
        "Indice": nom_indice,
        "Total Paires Testées": total_pairs,
        "Paires Validées (<0.05)": nb_coint,
        "Taux de Succès (%)": round(pct_coint, 2),
        "Qualité 'Or' (<0.01)": nb_strong,
        "Ratio Pépites (%)": round(pct_strong, 2),
        "Potentiel Latent (0.05-0.10)": nb_weak
    }

# --- CALCUL DES STATS ---
stats_dow = analyser_stats_marche(matrice_dow, "Dow Jones (US)")
stats_cac = analyser_stats_marche(matrice_cac, "CAC 40 (FR)")
```

```
# Création du Tableau Comparatif
df_stats = pd.DataFrame([stats_dow, stats_cac])
df_stats = df_stats.set_index("Indice")

print("TABLEAU COMPARATIF DES MARCHÉS :")
display(df_stats)

# --- VISUALISATION GRAPHIQUE POUR LE RAPPORT ---
# On va tracer le % de réussite côte à côte
ax = df_stats[["Taux de Succès (%)", "Ratio Pépites (%)"]].plot(kin

plt.title("Densité d'Opportunités d'Arbitrage : US vs France")
plt.ylabel("Pourcentage de Paires Cointégrées")
plt.xticks(rotation=0)
plt.grid(axis='y', linestyle='--', alpha=0.7)
plt.legend(["Cointégration Standard (P<0.05)", "Cointégration Forte

# Ajout des étiquettes de valeur sur les barres
for p in ax.patches:
    ax.annotate(f"{p.get_height()}%", (p.get_x() + p.get_width() /
                ha='center', va='center', xytext=(0, 9), textcoords

plt.show()
```

6. Conclusion

This empirical study demonstrates the power of the Engle-Granger methodology to build Market Neutral strategies. However, it also highlights three critical lessons for risk management:

1. **Capital Efficiency vs Statistics:** The "best" mathematical pair is not always the most profitable to trade if it locks up capital for too long (e.g, PEP/KO vs CVX/XOM).
2. **Model Risk & Black Swans:** The strategy achieved a high theoretical win rate (94%), but suffered its only major drawdown during the **March 2020 COVID-19 crash**. Systemic exogenous shocks can violently break historical cointegration.
3. **Execution Reality:** While theoretical PnL is positive, real-world deployment would require modeling transaction costs, margin requirements for holding periods up to 89 days, and execution slippage.