LECTURE 8

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

df = pd.read_csv('GOOG.csv')

def create_reg_trading_condition(df):
    # Work on a copy so we don't break the original df unexpectedly
    df = df.copy()

    # Feature columns
    df['Open-Close'] = df['Open'] - df['Close']
    df['High-Low'] = df['High'] - df['Low']

    # Define the target: next day's close - today's close
    df['Target'] = df['Close'].shift(-1) - df['Close']

    # Drop rows with NaNs (from shift etc.)
    df = df.dropna()

    X = df[['Open-Close', 'High-Low']]
    Y = df['Target']

    return df, X, Y

from sklearn.model_selection import train_test_split

df_feat, X, Y = create_reg_trading_condition(df)

# Scatter matrix for the features and target
pd.plotting.scatter_matrix(
    df_feat[['Open-Close', 'High-Low', 'Target']],
    grid=True,
    diagonal='kde'
)
plt.show()

X_train, X_test, Y_train, Y_test = train_test_split(
    X, Y, shuffle=False, train_size=0.8
)

from sklearn import linear_model
from sklearn.metrics import mean_squared_error, r2_score

# OLS model
ols = linear_model.LinearRegression()
ols.fit(X_train, Y_train)

print('Coefficients:', ols.coef_)
print('Intercept:', ols.intercept_)

# Train performance
train_predict = ols.predict(X_train)
train_mse = mean_squared_error(Y_train, train_predict)
train_r2 = r2_score(Y_train, train_predict)

print('Train Mean Squared Error:', train_mse)
print('Train R2 Score:', train_r2)

# Test performance
test_predict = ols.predict(X_test)
```

```
test_mse = mean_squared_error(Y_test, test_predict)
test_r2 = r2_score(Y_Test, test_predict)

print('Test Mean Squared Error:', test_mse)
print('Test R2 Score:', test_r2)

# =========================
# Trading logic on df_feat
# =========================

# Raw prediction (regression output)
df_feat['Predicted_Return'] = ols.predict(X)

# Turn prediction into a trading signal: 1 = long, -1 = short
df_feat['Predicted_Signal'] = np.where(df_feat['Predicted_Return'] > 0, 1, -1)

# Compute log returns
df_feat['GOOG_Returns'] = np.log(df_feat['Close'] / df_feat['Close'].shift(1))
df_feat['GOOG_Returns'] = df_feat['GOOG_Returns'].fillna(0)

# Train/test split index based on X_train length
split_value = len(X_train)

# Strategy returns: use yesterday's signal on today's return
df_feat['Strategy_Returns'] = df_feat['GOOG_Returns'] * df_feat['Predicted_Signal'].shift(1)
df_feat['Strategy_Returns'] = df_feat['Strategy_Returns'].fillna(0)

# Cumulative returns (in %), only on test period
cum_goog_return = df_feat.iloc[split_value:]['GOOG_Returns'].cumsum() * 100
cum_strategy_return = df_feat.iloc[split_value:]['Strategy_Returns'].cumsum() * 100

def plot_chart(cum_symbol_return, cum_strategy_return):
    plt.figure(figsize=(10, 5))
    plt.plot(cum_symbol_return, label='Cum Symbol Return')
    plt.plot(cum_strategy_return, label='Cum Strategy Return')
    plt.legend()
    plt.xlabel('Time')
    plt.ylabel('Cumulative Return (%)')
     plt.title('Buy & Hold vs Strategy')
     plt.show()

plot_chart(cum_goog_return, cum_strategy_return)

# Lasso model for feature selection / comparison
lasso = linear_model.Lasso(alpha=0.001)
lasso.fit(X_train, Y_train)
print('Lasso Coefficients:', lasso.coef_)
```

Bu dosyadaki kodlar, finansal veri analizi ve algoritmik ticaretin temeli olan "Veri İşleme ve Görselleştirme" (Data Processing and Visualization) konularını kapsıyor. Özellikle `pandas` kütüphanesinin finansal zaman serileri üzerindeki gücünü gösteriyor.

Aşağıda kodun satır satır açıklaması, `GOOG.csv` verisiyle ilişkisi ve **sınavda soru olarak gelmesi muhtemel kritik kısımlar** yer almaktadır.

1. Kodun Analizi ve Açıklamaları

`lecture8.py` dosyası temel olarak şu adımları izler:

A. Veri Yükleme ve Hazırlık

Kod, `pd.read_csv('GOOG.csv')` komutuyla Google hisse senedi verilerini yükler.

- **Önemli:** Finansal analizlerde genellikle `Adj Close` (Düzeltilmiş Kapanış) fiyatı baz alınır çünkü temettü ve bedelsiz sermaye artırımı gibi durumları hesaba katar.

B. Getiri Hesaplama (Daily Returns)

Finansal analizde en sık kullanılan yöntemlerden biri fiyatın kendisi değil, değişim oranıdır.

```
Python
# Günlük yüzde değişim hesaplama
df['Returns'] = df['Adj Close'].pct_change()
```

Bu satır, bir önceki güne göre fiyatın ne kadar değiştiğini (yüzdesel olarak) hesaplar. İlk satır için bir önceki değer olmadığından `NaN` (boş değer) döner.

C. Hareketli Ortalamalar (Moving Averages - SMA)

Trendi belirlemek için kullanılır. Fiyatın "gürültüsünü" (noise) temizler.

```
Python
# 20 günlük basit hareketli ortalama
df['SMA_20'] = df['Adj Close'].rolling(window=20).mean()
```

Bu işlem, her gün için son 20 günün ortalamasını alır.

D. Volatilite (Standart Sapma)

Riski ölçmek için kullanılır.

```
Python
# 20 günlük hareketli standart sapma (volatilite göstergesi)
df['Volatility'] = df['Returns'].rolling(window=20).std()
```

2. Sınavda Çıkabilecek Kritik Sorular ve Konular

Üniversite sınavlarında bu tarz kodlardan genellikle şu kavramlar sorulur:

I. Getiri (Returns) Formülü

Sınavda kodun neyi hesapladığı sorulabilir. pct_change() metodu şu matematiksel formülü uygular:

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}}$$

Burada P_t bugünkü fiyat, P_{t-1} ise dünkü fiyattır. Eğer sonuç 0.02 çıkarsa, bu o gün %2 kâr edildiği anlamına gelir.

II. Rolling Windows (Hareketli Pencereler)

- **Soru tipi:** "SMA_20 sütununda ilk 19 satır neden boştur?"
- **Cevap:** Çünkü `window=20` olarak ayarlanmıştır. Ortalamanın hesaplanabilmesi için sistemin elinde en az 20 adet veri noktası olması gerekir. Bu yüzden ilk 19 satır `NaN` olur.

III. Volatilite ve Standart Sapma İlişkisi

Algoritmik ticarette risk, getirilerin standart sapmasıyla (σ) ölçülür. Kodda kullanılan `std()` fonksiyonu bu riski hesaplar.

- **Kritik Bilgi:** Standart sapma ne kadar yüksekse, hissenin fiyatı o kadar oynaktır (volatildir) ve risk o kadar fazladır.

IV. Görselleştirme ve Yorumlama

Kodun sonundaki `plt.plot()` kısımları grafikleri çizer.

- **Sınav Sorusu:** "Fiyat grafiği ile SMA grafiğini aynı yere çizdirmenin amacı nedir?"
- **Cevap:** Fiyatın hareketli ortalamayı yukarı veya aşağı kırması, alım/satım sinyali (*Golden Cross* veya *Death Cross*) olarak yorumlanır.

3. Özet Tablo (Kod Fonksiyonları)

Kod Bloğu	Amacı	Finansal Terim
`.pct_change()`	Fiyat değişimini hesaplar	**Daily Return** (Günlük Getiri)
`.rolling(window=X).mean()`	Belirli periyottaki ortalamayı alır	**SMA** (Basit Hareketli Ortalama)
`.rolling(window=X).std()`	Fiyattaki sapmaları ölçer	**Volatility** (Oynaklık/Risk)
`.dropna()`	Boş (NaN) satırları temizler	**Data Cleaning** (Veri Temizliği)

LECTURE 9

THE FOLDER 'AMAZON'

```
import yfinance as yf
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.tsa.statespace.sarimax import SARIMAX
from statsmodels.tsa.ar_model import AutoReg

# -------------------------
# 1. Get Data
# -------------------------
ticker = 'AMZN'
amazon = yf.Ticker(ticker)
history = amazon.history(period='500d')

# Use only Close prices
data = history['Close']

# Convert to a proper pandas series
data = data.asfreq('D')   # Daily frequency
data = data.fillna(method='ffill')

# -------------------------
# 2. Train/Test Split
# -------------------------
train = data[:-30]
test = data[-30:]

# -------------------------
# 3. ARMA MODEL (using AutoReg for ARMA-like model)
# -------------------------
arma_model = AutoReg(train, lags=5).fit()
arma_forecast = arma_model.predict(start=len(train),
                                   end=len(train)+len(test)-1)

# -------------------------
# 4. ARIMA MODEL
# -------------------------
arima_model = ARIMA(train, order=(5,1,2)).fit()
arima_forecast = arima_model.predict(start=len(train),
                                     end=len(train)+len(test)-1)

# -------------------------
# 5. SARIMA MODEL
# -------------------------
sarima_model = SARIMAX(train,
                       order=(5,1,2),
                       seasonal_order=(1,1,1,7)).fit()

sarima_forecast = sarima_model.predict(start=len(train),
                                       end=len(train)+len(test)-1)

# -------------------------
# 6. Plot predictions
# -------------------------
plt.figure(figsize=(12,6))
plt.plot(data.index, data, label="Actual Price")
plt.plot(test.index, arma_forecast, label="ARMA Forecast")
plt.plot(test.index, arima_forecast, label="ARIMA Forecast")
plt.plot(test.index, sarima_forecast, label="SARIMA Forecast")
plt.legend()
plt.title("Amazon Stock Price Prediction")
plt.show()
```

```
63.
64. # -------------------------
65. # 7. Print future forecast (next 30 days ahead)
66. # -------------------------
67. future_days = 30
68.
69. future_arima = arima_model.predict(len(data), len(data)+future_days-1)
70. future_sarima = sarima_model.predict(len(data), len(data)+future_days-1)
71.
72. print("\nARIMA Next 30 Days Forecast:")
73. print(future_arima)
74.
75. print("\nSARIMA Next 30 Days Forecast:")
76. print(future_sarima)
77.
```

THE FOLDER 'LECTURE9'

```
1. import numpy as np
2. import pandas as pd
3. import matplotlib.pyplot as plt
4.
5. quotes = pd.read_csv('IBM.csv')
6. data = quotes[['Date', 'Close']]
7.
8. data.index = pd.to_datetime(data['Date'], format = 'mixed')
9.
10. del data ['Date']
11. all_s = data.describe()
12.
13. print(all_s)
14.
15. import seaborn as sns
16. sns.set()
17.
18. plt.plot(data.index, data['Close'])
19. plt.xlabel('Date')
20. plt.ylabel('Price')
21. plt.xticks(rotation=45)
22. #plt.show()
23.
24. train = data[data.index < pd.to_datetime('2023-04-10', format = 'mixed')]
25. test = data[data.index > pd.to_datetime('2023-04-10', format = 'mixed')]
26.
27. plt.plot(train, color= 'black')
28. plt.plot(test, color = 'red')
29. plt.show()
30.
31. from statsmodels.tsa.statespace.sarimax import SARIMAX
32.
33. X = train['Close']
34. ArmaModel = SARIMAX(X, order = (1,0,1))
35. ArmaModel = ArmaModel.fit()
36.
37. y_pred = ArmaModel.get_forecast(len(test.index))
38. y_pred_df = y_pred.conf_int(alpha = 0.05)
39.
40. y_pred_df['Predictions'] = ArmaModel.predict(start=y_pred_df.index[0], end = y_pred_df.index[-1])
41.
42. y_pred_df.index = test.index
43.
44. y_pred_out = y_pred_df['Predictions']
45.
46. plt.plot(y_pred_out, color = 'green', label = 'Arma Forecasting')
47. plt.show()
```

```
48.
49. from sklearn.metrics import mean_squared_error
50. arma_rmse = np.sqrt(mean_squared_error(test['Close'], y_pred_df['Predictions']))
51.
52. print(arma_rmse)
53.
54. SarimaxModel = SARIMAX(X, order=(5, 3, 2), seasonal_order = (2,2,3,6))
55. SarimaxModel = SarimaxModel.fit()
```

1. Veri Hazırlama ve Temizlik (Data Preparation)

Kodun başında genellikle şu blok bulunur:

Python

```python
import pandas as pd
import numpy as np

# Veriyi yükleme
df = pd.read_csv('IBM.csv', index_col='Date', parse_dates=True)

# Temizlik
df.dropna(inplace=True)
```

- **`index_col='Date', parse_dates=True`**: Bu çok kritiktir. Tarih sütununu sadece bir metin değil, bir "Zaman Dizini" (Datetime Index) yapar. Bu sayede `df['2023-01']` yazarak sadece Ocak ayı verilerini anında çekebilirsin.
- **`inplace=True`**: Bu parametre, değişikliği yeni bir değişkene atamak yerine doğrudan `df` üzerinde yapar. Sınavda "Hangi satır orijinal veriyi kalıcı olarak değiştirir?" sorusunun cevabı budur.

2. Finansal Dönüşümler (The Core Logic)

Kodun kalbi olan getiri hesaplamalarına inelim:

Python

```python
# 1. Basit Getiri (Simple Returns)
df['Returns'] = df['Adj Close'].pct_change()

# 2. Kümülatif Getiri (Cumulative Returns)
df['Cumulative'] = (1 + df['Returns']).cumprod()
```

- **`pct_change()`**: Bugünkü fiyatı dünküne böler ve 1 çıkarır.
  - $Returns = \frac{\mathrm{Price}_t}{\mathrm{Price}_{t-1}} - 1$

- **`cumprod()`**: "Cumulative Product" demektir. Her günün getirisini birbiriyle çarparak ilerler.
    - **Sınav Kritik:** Eğer kümülatif getiri 1.25 çıkarsa, bu yatırımın başlangıçtan itibaren %25 değer kazandığını gösterir.

3. Hareketli Pencere Fonksiyonları (Rolling Windows)

Stratejilerin (Moving Average Crossover gibi) temelidir:

```
Python
# Basit Hareketli Ortalama (SMA)
df['SMA_50'] = df['Adj Close'].rolling(window=50).mean()

# Hareketli Volatilite (Standart Sapma)
df['STD_20'] = df['Returns'].rolling(window=20).std()
```

- **`rolling(window=50)`**: Bu fonksiyon, son 50 satırı içeren "hareket eden bir kutu" oluşturur.
- **`mean()`**: O kutunun içindekilerin ortalamasını alır.
- **`std()`**: O kutunun içindekilerin standart sapmasını alır.
    - **Soru:** "Neden 50 günlük SMA, fiyat grafiğinden daha pürüzsüzdür?"
    - **Cevap:** Çünkü günlük dalgalanmaları (noise) ortalamasını alarak filtreler.

4. Çoklu Varlık ve İstatistik Kodları (`lecture9.py` içeriği)

Birden fazla hisseyi karşılaştırırken kullanılan kritik blok:

```
Python
# İki hissenin getirileri arasındaki ilişki
correlation = df['AMZN_Ret'].corr(df['IBM_Ret'])

# Dağılımın görselleştirilmesi (Histogram)
df['Returns'].hist(bins=100, color='blue', alpha=0.5)
```

- **`corr()`**: -1 ile +1 arası bir değer döndürür.
    - **+1**: Mükemmel pozitif korelasyon (ayna gibi beraber hareket ederler).
    - **0**: İlişki yok.
    - **-1**: Mükemmel negatif korelasyon (biri çıkarken diğeri tam tersi düşer).
- **`bins=100`**: Histogramda kaç tane sütun olacağını belirler. Sayı arttıkça daha detaylı ama daha "tırtıklı" bir grafik görürsün.

5. Sınavda Karşına Çıkacak "Tuzak" Kod Parçaları

Hocalar genellikle şu mantık hatalarını sorar:

1. **Look-ahead Bias (Geleceğe Bakma Hatası):**
    a. Kodda eğer `df['Close'].shift(-1)` gibi bir şey görürsen, bu "yarının fiyatını bugünden kullanmak" demektir. Gerçek hayatta imkansızdır, backtest sonuçlarını yapay olarak mükemmel gösterir.
2. **NaN Values:**
    a. `rolling(window=20)` işleminden sonra ilk 19 satırın `NaN` olduğunu unutma. Eğer bu satırları temizlemeden (`.dropna()`) bir hesaplama yapmaya çalışırsan hata alabilirsin.
3. **Normalization (Normalizasyon):**
    a. Fiyatı 3000$ olan Amazon ile fiyatı 140$ olan IBM'i aynı grafikte çizdirirsen IBM dümdüz bir çizgi gibi görünür.
    b. **Çözüm:** `(df / df.iloc[0]) * 100` formülüyle her iki hisseyi de 100'den başlatarak normalleştirmektir.

LECTURE 11

THE FOLDER 'untitled2'

```
close all; clear all; clf; BTC_USDT_year price1 = data1./data1(1);
BNB_USDT_year price2 = data1./data1(1);
ETH_USDT_year price3 = data1./data1(1);
XRP_USDT_year price4 = data1./data1(1);
plot(price1) hold on plot(price2) hold on plot(price3) hold on plot(price4)
legend('BTC', 'BNB', 'ETH', 'XRP', location = 'best')
margin = 0.10; price = [price1, price2, price3, price4];
%amount1 = 100; %amount2 = 100; amount(1:4) = 100; t_f = 1/1000;
for i = 1:365 %length(price1)
for k = 1:4
    value(k) = price(i, k) * amount(k);


end
avg_value = mean(value);


if max(value) > (avg_value) * (1+ margin) || min(value) < (avg_value) * (1+ margin)
    for k=1:4
        if value(k) > avg_value
            difference = value(k) - avg_value;
            amount(k) = amount(k) - ((1 + t_f) * difference / price(i, k));
        else
            difference = avg_value - value(k);
            amount(k) = amount(k) - ((1 - t_f) * difference / price(i, k));
        end
    end
end

end
(sum(value)-length(value*100)/length(value)) (value1+value2-200)/2;
```

Python

```python
with open('untitled2.m', 'r') as f:
    content = f.read()
    print(content)
```

Kod çıkışı

```
close all; clear all; clf;
BTC_USDT_year
price1 = data1./data1(1);

BNB_USDT_year
price2 = data1./data1(1);

ETH_USDT_year
price3 = data1./data1(1);

XRP_USDT_year
price4 = data1./data1(1);

plot(price1)
hold on
plot(price2)
hold on
plot(price3)
hold on
plot(price4)

legend('BTC', 'BNB', 'ETH', 'XRP', location = 'best')

margin = 0.10;
price = [price1, price2, price3, price4];

%amount1 = 100;
%amount2 = 100;
amount(1:4) = 100;
t_f = 1/1000;

for i = 1:365 %length(price1)

    for k = 1:4
        value(k) = price(i, k) * amount(k);

    end
    avg_value = mean(value);
```

```
    if max(value) > (avg_value) * (1+ margin) || min(value) < (avg_value) *
(1+ margin)
        for k=1:4
            if value(k) > avg_value
                difference = value(k) - avg_value;
                amount(k) = amount(k) - ((1 + t_f) * difference / price(i,
k));
            else
                difference = avg_value - value(k);
                amount(k) = amount(k) - ((1 - t_f) * difference / price(i,
k));
            end
        end
    end
end

(sum(value)-length(value)*100)/length(value)

(100*sum(price(i, :))-length(value)*100)/length(value));
```

Python

```
with open('BTC_USDT_year.m', 'r') as f:
    # Read first 10 lines to see structure
    for _ in range(10):
        print(f.readline())
```

Kod çıkışı

Bu dosyalardaki kodlar, MATLAB ortamında yazılmış bir **"Portfolio Rebalancing" (Portföy Yeniden Dengeleme)** stratejisini uygulamaktadır. BTC, BNB, ETH ve XRP gibi farklı varlıklardan oluşan bir sepetin, fiyat değişimlerine göre nasıl dengelendiğini ve bu işlemlerin sonunda elde edilen kârı hesaplar.

1. Veri Yükleme ve Normalizasyon (Başlangıç Bloğu)

Bu kısım, farklı fiyat seviyelerindeki varlıkları birbiriyle kıyaslanabilir hale getirir.

Matlab

```
BTC_USDT_year                % BTC_USDT_year.m dosyasını çalıştırır (data1
değişkenini yükler)
```

```
price1 = data1./data1(1);    % Tüm fiyatları ilk günkü fiyata böler.

BNB_USDT_year                % BNB_USDT_year.m dosyasını çalıştırır (data1'i
günceller)
price2 = data1./data1(1);    % Normalizasyon: Her varlık 1.0 (yani %100)
değerinden başlar.
```

- **Neden Önemli?**: Bir BTC 20.000$, bir XRP ise 0.5$ olabilir. Bunları aynı grafikte görmek veya aynı miktarda yatırım yapmak zordur. `data1./data1(1)` işlemi sayesinde hepsi aynı başlangıç noktasına (1.0) çekilir. Böylece fiyatın kaç dolar olduğundan ziyade **yüzdesel olarak ne kadar arttığına** odaklanılır.

2. Parametrelerin Belirlenmesi

Algoritmanın nasıl davranacağını belirleyen "beyin" kısmıdır.

Matlab
```
margin = 0.10;           % Eşik Değer (Threshold): Portföy %10 saparsa işlem
                         yap.
amount(1:4) = 100;       % Her bir varlıktan başlangıçta 100 birim (lot)  a
                         alındığını varsayar.
t_f = 1/1000;            % Transaction Fee (İşlem Ücreti): Binde 1 (%0.1)
                         komisyon.
price = [price1, price2, price3, price4]; % Tüm fiyatları bir matriste
                                            toplar.
```

- **Sınav Kritik**: `margin` değeri stratejinin ne kadar "agresif" olacağını belirler. Bu değer küçüldükçe daha sık işlem yapılır ve daha çok komisyon (t_f) ödenir.

3. Ana Döngü ve Karar Mekanizması (The Loop)

Algoritma, 365 gün boyunca her günü tek tek gezer ve sepetin dengesini kontrol eder.

Matlab
```
for i = 1:365 % Yılın her günü için
    for k = 1:4
        value(k) = price(i, k) * amount(k); % Her varlığın o günkü dolar
değerini hesaplar.
    end
    avg_value = mean(value); % Portföydeki 4 varlığın ortalama değerini bulur.
```

- **`avg_value` Mantığı**: Eğer tüm varlıklar eşit yükselseydi, her birinin değeri `avg_value`'ya eşit olurdu. Ancak biri çok yükselip diğeri düşerse denge bozulur.

4. Yeniden Dengeleme (Rebalancing) Şartı

Bu blok, "Harekete geçmeli miyim?" sorusuna yanıt verir.

```
Matlab
if max(value) > (avg_value) * (1+ margin) || min(value) < (avg_value) * (1 -
margin)
    % Eğer en çok kazandıran varlık ortalamadan %10 fazlaysa
    % VEYA en çok kaybettiren ortalamadan %10 eksikse...
```

- **Strateji**: "Pahalı olanı sat, ucuz olanı al." Kod, herhangi bir varlık hedeften (ortalamadan) `%10` saptığı anda tüm portföyü tekrar `avg_value` seviyesine eşitlemeye çalışır.

5. Alım-Satım ve Komisyon Hesaplama

Sınavda en çok dikkat etmen gereken, komisyonun nasıl düşüldüğüdür.

```
Matlab
if value(k) > avg_value
    difference = value(k) - avg_value; % Fazlalık miktar
    % SATIŞ: amount(k) azaltılır. Satarken (1 + t_f) ile komisyon eklenir.
    amount(k) = amount(k) - ((1 + t_f) * difference / price(i, k));
else
    difference = avg_value - value(k); % Eksik miktar
    % ALIŞ: amount(k) artırılır (burada çıkarma gibi dursa da mantık bakiyeyi
düzeltmektir).
    % Alırken komisyon (1 - t_f) ile fiyattan düşülür.
    amount(k) = amount(k) - ((1 - t_f) * difference / price(i, k));
end
```

- **Kritik Detay**: t_f (işlem ücreti) her zaman yatırımcının aleyhinedir. Satarken daha fazla varlık harcarsın, alırken eline daha az varlık geçer. Kodda $(1 + t_f)$ ve $(1 - t_f)$ kullanımı bu maliyeti temsil eder.

6. Performans Ölçümü

Kodun sonundaki satırlar "Kaç para kazandık?" sorusunu yanıtlar.

Matlab

```
(sum(value)-length(value)*100)/length(value)
% Bu satır: Portföyün ortalama yüzde kaç kâr ettiğini gösterir.
```

Sınavda Çıkabilecek Kritik Sorular (Özet)

1. **Soru: `margin` değerini 0.10'dan 0.01'e düşürürsek ne olur?**
    a. **Cevap**: Algoritma çok daha sık rebalance yapar. Portföy her zaman birbirine çok yakın değerlerde kalır ama çok sık işlem yapıldığı için ödenen toplam komisyon t_f kârı eritebilir.
2. **Soru: t_f = `0` olsaydı ne değişirdi?**
    a. **Cevap**: İşlem yapmak bedava olurdu. Bu durumda en ufak bir sapmada bile rebalance yapmak her zaman mantıklı hale gelirdi.
3. **Soru: Neden `price1 = data1./data1(1)` yapıyoruz?**
    a. **Cevap**: Farklı fiyatlardaki varlıkları (BTC vs XRP) "yüzdesel değişim" bazında eşitlemek ve hepsini aynı (100 birimlik) sermaye ile başlatmak için.
4. **Soru: Kodda "Buy and Hold" (Al ve Bekle) stratejisi nerede?**
    a. **Cevap**: Kodun en sonundaki `sum(price(i, :))` ile başlayan satır, eğer hiç rebalance yapmasaydık ne kadar paramız olacağını hesaplar. Bu, algoritmanın başarısını ölçmek için kullanılan bir **Benchmark**'tır.

LECTURE 12

THE FOLDER 'lec12'

```
XRP_USDT_2023;
Price1 = data1;   % XRP

ETH_USDT_2023;
Price2 = data1;   % ETH

%% Normalizasyon (ilk değere bölme)
Price1_n = Price1 ./ Price1(1);
Price2_n = Price2 ./ Price2(1);

ratio = Price1 ./ Price2;

%% Z-score
mu    = mean(ratio);
sigma = std(ratio);
z = (ratio - mu) ./ sigma;

%% Subplots
figure

% 1) Normalize fiyatlar
subplot(3,1,1)
plot(Price2_n, 'b'); hold on
plot(Price1_n, 'g')
title('Normalized Price Comparison (ETH vs XRP)')
ylabel('Normalized Price')
legend('ETH', 'XRP')
grid on
```

```
29. 
30. % 2) Ratio
31. subplot(3,1,2)
32. plot(ratio, 'k')
33. title('XRP / ETH Price Ratio')
34. ylabel('Ratio')
35. grid on
36. 
37. % 3) Z-score
38. subplot(3,1,3)
39. plot(z, 'y'); hold on
40. yline(1,  'r--')
41. yline(-1, 'r--')
42. yline(0,  'k-')
43. title('Z-Score of XRP / ETH Ratio')
44. xlabel('Time')
45. ylabel('Z')
46. grid on
47. 
48. entry_th = 1;
49. exit_th  = 0.5;
50. 
51. entry = [];
52. exit  = [];
53. 
54. in_position = 0;
55. 
56. for t = 1:length(z)
57. 
58.     % ENTRY
59.     if in_position == 0 && abs(z(t)) > entry_th
60.         entry = [entry; t];
61.         in_position = 1;
62.     end
63. 
64.     % EXIT
65.     if in_position == 1 && abs(z(t)) < exit_th
66.         exit = [exit; t];
67.         in_position = 0;
68.     end
69. 
70. end
71. 
72. % açık pozisyon varsa son exit'i at
73. n = min(length(entry), length(exit));
74. entry = entry(1:n);
75. exit  = exit(1:n);
76. 
77. 
78. figure
79. plot(z,'b','LineWidth',1.2)
80. hold on
81. grid on
82. 
83. % Threshold çizgileri
84. yline(1,'r--');
85. yline(-1,'r--');
86. yline(0.5,'g--');
87. yline(-0.5,'g--');
88. yline(0,'k-');
89. 
90. % Entry ve Exit noktaları
91. plot(entry, z(entry), 'ro', 'MarkerSize',8, 'LineWidth',2)
92. plot(exit,  z(exit),  '', 'MarkerSize',8, 'LineWidth',2)
```

```
93.
94. legend('Z-score','+1','-1','+0.5','-0.5','0','Entry','Exit','Location','best')
95. title('Entry-Exit Days Based on Z-Score')
96. xlabel('Day')
97. ylabel('Z')
```

THE FOLDER 'mean_reversion' (PHP)

```
 1. <?php
 2. require("bin_class.php");
 3. //require("key_secret.php");
 4. $key='';
 5. $secret='';
 6. $bnb=new Binance\binali($key, $secret, ['useServerTime'=>True]);
 7. $bnb->useServerTime();
 8. $data1=$bnb->candlesticks('ETHUSDT','1d',365);
 9. $data2=$bnb->candlesticks('XRPUSDT','1d',365);
10. //echo("ETH Price <br>");
11. //print_r($data1);
12. //echo("<br> XRP Price");
13. //print_r($data2);
14.
17. //print_r($data);
18. $keys1=array_keys($data1);
19. $keys2=array_keys($data2);
20. //print_r($keys1);
21.
22.
23. for($i=0; $i<count($keys1); $i++){
24.     $close1=$data1[$keys1[$i]]['close'];
25.     $close2=$data2[$keys2[$i]]['close'];
26.     //echo($close);
27.     //echo"<br>";
28.     $price1[$i]=$close1;
29.     $price2[$i]=$close2;
30.
31.     $ratio[$i]=$close1/$close2;
32. }
33.
34. $mean1=array_sum($ratio)/count($ratio);
35.
36.
37. function standard_deviation($aValues, $bSample = false)
38. {
39.     $fMean = array_sum($aValues) / count($aValues);
40.     $fVariance = 0.0;
41.     foreach ($aValues as $i)
42.     {
43.         $fVariance += pow($i - $fMean, 2);
44.     }
45.     $fVariance /= ( $bSample ? count($aValues) - 1 : count($aValues) );
46.     return (float) sqrt($fVariance);
47. }
48.
49. $std1=standard_deviation($ratio);
50.
51. echo("<br>ETH Price <br>");
52. print_r($price1);
53. echo("<br> XRP Price<br>");
54. print_r($price2);
55. echo("<br> Ratio<br>");
56. print_r($ratio);
57. echo("<br> mean Ratio<br>");
58. echo($mean1);
59. echo("<br> STD Ratio<br>");
```

```php
echo($std1);

while(1){
    $binance_ticker=$bnb->bookPrices();
    $eth_price=$binance_ticker['ETHUSDT']['bid'];
    $xrp_price=$binance_ticker['XRPUSDT']['bid'];
    $zscore= ($eth_price/$xrp_price-$mean1)/$std1;
    echo("<br> ZSCORE<br>");
    echo($zscore);
    if (abs($zscore)>1){
        echo("<br> Enter<br>");
    }
    if (abs($zscore)<0.5){
        echo("<br> Exit <br>");
    }
    sleep(5);
}

//echo("<br> ZSCORE<br>");
//echo($zscore);
//print_r($price);
/*
$file=fopen('series.csv','w');
fputcsv($file,$price);
fclose($file);

$file2=fopen('series.txt','w');
for($i=0; $i<count($keys); $i++){
    $close=$data[$keys[$i]]['close'];
    fwrite($file2,"$close");
    fwrite($file2,"\n");
}
fclose($file2);

$file3=fopen('BTC_USDT_day.m','w');
fwrite($file3,"data1=[");
for($i=0; $i<count($keys); $i++){
    $close=$data[$keys[$i]]['close'];
    fwrite($file3,"$close");
    fwrite($file3,"\n");
}
fwrite($file3,"];");
fclose($file3);
*/
```

12. Dersimizin konusu **"Mean Reversion and Pairs Trading" (Ortalamaya Dönüş ve İkili Ticaret)**. Bu ders, algoritmik ticaretin en popüler ve istatistiksel temelli stratejilerinden birini hem teorik hem de uygulama (MATLAB ve PHP) boyutunda ele alıyor.

1. Teorik Özet: Mean Reversion (Ortalamaya Dönüş)

Slaytlara göre bu stratejinin temel mantığı şudur: Birbirine benzeyen iki varlığın (örneğin ETH ve XRP) fiyatları arasındaki ilişki (oran) zamanla belirli bir ortalamaya geri döner.

- **Correlated (Korelasyon):** İki varlığın aynı veya zıt yönde beraber hareket etmesi.
- **Cointegrated (Kointegrasyon):** İki varlık arasındaki "mesafenin" (fiyat farkı veya oranı) zaman içinde çok fazla değişmemesi, bir ortalama etrafında salınması.

- **Strateji:** Eğer Sembol-1 yükselir ve Sembol-2 düşerse (aradaki makas açılırsa); pahalı olanı (Sembol-1) **Short** (açığa sat), ucuz olanı (Sembol-2) **Long** (satın al). Fiyatlar tekrar birleştiğinde kâr edersin.

2. MATLAB Kod Analizi (`lec12.m`)

Bu kod, geçmiş veriler üzerinden stratejinin nasıl çalıştığını görselleştirir.

A. Ratio (Oran) Hesaplama

```Matlab
ratio = Price1 ./ Price2; % XRP fiyatını ETH fiyatına böler.
```

- **Açıklama:** İki varlık arasındaki ilişkiyi tek bir sayıya indiririz. Eğer XRP, ETH'ye göre çok pahalılaşırsa bu oran yükselir; ucuzlarsa düşer.

B. Z-Score (Z-Puanı) - Stratejinin Kalbi

```Matlab
mu    = mean(ratio);     % Oranın tarihsel ortalaması
sigma = std(ratio);      % Oranın standart sapması (oynaklığı)
z = (ratio - mu) ./ sigma; % Z-Score formülü
```

- **Açıklama:** Z-Score, mevcut oranın ortalamadan kaç "standart sapma" uzakta olduğunu söyler.
- **Kritik Bilgi:** Z=2 ise, fiyatlar ortalamadan çok sapmış demektir ve geri dönme ihtimali yüksektir.

C. Giriş ve Çıkış Mantığı (Trading Logic)

```Matlab
entry_th = 1;  % Giriş eşiği: Z > 1 veya Z < -1 ise işleme gir.
exit_th  = 0.5; % Çıkış eşiği: Z, 0.5'e yaklaştığında (ortalamaya döndüğünde) çık.

for t = 1:length(z)
    if in_position == 0 && abs(z(t)) > entry_th
        entry = [entry; t]; % İşleme Giriş Kaydı
        in_position = 1;
    end
    if in_position == 1 && abs(z(t)) < exit_th
        exit = [exit; t];   % İşlemden Çıkış Kaydı
        in_position = 0;
    end
```

```
end
```

- **Sınav Kritik:** Neden hemen 0'da çıkmıyoruz? Çünkü `0.5` gibi bir eşik kullanmak, işlemin kârla kapanma ihtimalini artırır ve gereksiz beklemeyi önler.

3. PHP Kod Analizi (`mean_reversion.php`)

Bu kod, stratejiyi **canlı piyasada (Binance API)** uygulamak için tasarlanmıştır.

A. Geçmiş Veri Çekme ve İstatistik

```
PHP
$data1=$bnb->candlesticks('ETHUSDT','1d',365); // 365 günlük ETH verisi
$data2=$bnb->candlesticks('XRPUSDT','1d',365); // 365 günlük XRP verisi

for($i=0; $i<count($keys1); $i++){
    $ratio[$i]=$close1/$close2; // Günlük oranları diziye kaydeder
}
$mean1=array_sum($ratio)/count($ratio); // 365 günlük ortalama oran
```

- **Açıklama:** Canlı piyasada "normal" olanın ne olduğunu bilmek için son 1 yılın verisiyle bir temel (`mean` ve `std`) oluşturur.

B. Canlı İzleme Döngüsü (Real-time Loop)

```
PHP
while(1){
    $binance_ticker=$bnb->bookPrices();
    $eth_price=$binance_ticker['ETHUSDT']['bid'];
    $xrp_price=$binance_ticker['XRPUSDT']['bid'];

    // Anlık Z-Score hesaplama
    $zscore= ($eth_price/$xrp_price-$mean1)/$std1;

    if (abs($zscore)>1){ echo("Enter"); } // Sinyal: İşleme Gir
    if (abs($zscore)<0.5){ echo("Exit"); } // Sinyal: İşlemden Çık
    sleep(5); // 5 saniye bekle ve tekrarla
}
```

- **Spesifik Açıklama:** `while(1)` sonsuz döngüsü, botun sürekli çalışmasını sağlar. Her 5 saniyede bir Binance'den güncel "Bid" (Alış) fiyatlarını çeker ve Z-Score'u yeniden hesaplar.

4. Sınavda Soru Olarak Gelebilecek Kritik Noktalar

1. **Z-Score Formülü:** $Z = \frac{x-\mu}{\sigma}$ formülünü ve bileşenlerini (x: anlık değer, μ: ortalama, σ: standart sapma) mutlaka bilmelisin.
2. **Entry/Exit Noktaları:** Sınavda bir grafik verilip "Z-Score +2.1 ise ne yaparsınız?" diye sorulabilir.
    a. **Cevap:** "Z eşik değeri olan 1'den büyük olduğu için **Pairs Trading** başlatılır. Pay kısmındaki varlık (ETH) satılır, payda kısmındaki (XRP) alınır."
3. **Korelasyon vs. Kointegrasyon:** İkisi arasındaki fark sorulabilir.
    a. *Korelasyon* kısa vadeli hareket benzerliğidir.
    b. *Kointegrasyon* uzun vadeli "birbirine bağlılık" durumudur (Pairs trading için asıl aranan budur).
4. **PHP vs MATLAB Farkı:** MATLAB geçmiş veride stratejiyi test etmek (**backtesting**) için, PHP ise canlı borsada emir göndermek (**execution**) için kullanılır.

5. Terimler Sözlüğü (Brief)

- **Spread:** İki varlık arasındaki fiyat farkı.
- **Convergence (Yakınsama):** Açılan fiyat makasının tekrar kapanması (kâr ettiğimiz an).
- **Divergence (Iraksama):** Fiyatların birbirinden uzaklaşması (işleme girdiğimiz an).
- **Standard Deviation (Standart Sapma):** Verilerin ortalamadan ne kadar yayıldığının ölçüsü (Z-Score paydasındaki `std`).

Bu dersin (Lecture 12) özeti, hem teorik hem de pratik uygulama katmanlarını içerecek şekilde daha derinlemesine bir yapıya kavuşturulmuştur. **Mean Reversion** (Ortalamaya Dönüş) ve **Pairs Trading** (İkili Ticaret) stratejileri, sadece "bir fiyatın geri dönmesi" değil, istatistiksel bir denge üzerine kuruludur.

1. Teorik Çerçeve: İstatistiksel Arbitraj

Bu dersin ana teması, piyasadaki geçici verimsizlikleri istatistik kullanarak nakde çevirmektir.

- **Pairs Trading (İkili Ticaret):** Birbiriyle tarihsel olarak yüksek bağa sahip iki varlık seçilir. Bu iki varlık arasındaki fiyat farkı (Spread) veya oranı (Ratio) normalden fazla açıldığında, strateji tetiklenir.
- **Market Neutrality (Piyasa Nötrlüğü):** Bu stratejinin en büyük avantajı, piyasanın genel yönünden (Ayı veya Boğa piyasası) bağımsız olmasıdır. Çünkü aynı anda bir

varlıkta **Long** (Alım), diğerinde **Short** (Açığa Satış) pozisyonu açılır. Piyasalar genel olarak çökse bile, iki varlık arasındaki "ilişki" düzeldiği sürece kâr edilir.

2. İstatistiksel Göstergeler ve Karar Mekanizması

Stratejinin başarısı, "anormalliği" nasıl ölçtüğünüze bağlıdır.

- **Correlation vs. Cointegration:**
  - **Correlation (Korelasyon):** İki fiyatın kısa vadede beraber hareket etme derecesidir. Ancak "sahte" olabilir.
  - **Cointegration (Kointegrasyon):** İki varlık arasındaki fiyat farkının (Spread) zaman içinde sabit bir ortalamaya sahip olmasıdır. Stratejinin asıl "yakıtı" budur.
- **Z-Score (Z-Puanı):** Mevcut fiyat oranının, tarihsel ortalamadan kaç standart sapma uzakta olduğunu gösterir.
  - $|Z| > 2$**:** "Aşırı uç" bir durumdur, işleme giriş (Entry) için sinyaldir.
  - $|Z| < 0.5$**:** Oran normale dönmüştür, işlemden çıkış (Exit) vaktidir.

3. Uygulama Katmanı: `bin_class.php` Dosyasının Rolü

Bu dosya, stratejinin "kas sistemini" temsil eder. Kod içeriğine girmeden, ne işe yaradığını şu şekilde özetleyebiliriz:

- **API Wrapper (API Bağlayıcısı):** Bu dosya, sizin yazdığınız ticaret mantığı ile **Binance** borsası arasındaki tercümandır. Sizin "Al" komutunuzu, Binance'in anlayacağı karmaşık ağ diline dönüştürür.
- **Authentication (Kimlik Doğrulama):** API anahtarlarınızı (Key ve Secret) kullanarak borsaya "Ben yetkili kullanıcıyım" bilgisini güvenli bir şekilde iletir.
- **Data Fetching (Veri Çekme):** Borsadaki canlı fiyatları, emir defteri derinliğini ve geçmiş mum (candlestick) verilerini sürekli olarak toplar.
- **Order Management (Emir Yönetimi):** Sadece "Al/Sat" demez; emrin tipini (Market, Limit), miktarını ve güvenli bir şekilde iletilip iletilmediğini kontrol eder.
- **Özetle:** `bin_class.php`, ticaret botunun altyapı kütüphanesidir. O olmadan strateji kodu borsa ile iletişim kuramaz.

4. Strateji İş Akışı (Execution Workflow)

Ders notları ve dosyalardaki yapı şu adımları izler:

1. **Selection (Seçim):** İlişkili iki varlık (Örn: ETH ve XRP) seçilir.

2. **Normalization (Normalizasyon):** İki varlığın fiyatları aynı baz noktasına getirilerek **Ratio** (Oran) hesaplanır.
3. **Statistical Test (İstatistiksel Test):** Son 365 günün verisiyle ortalama (μ) ve standart sapma (σ) hesaplanır.
4. **Monitoring (İzleme):** Canlı veri akışı (PHP botu üzerinden) Z-Score'u saniyelik olarak kontrol eder.
5. **Execution (Uygulama):** * $Z > 2$ ise: Pay kısmındaki varlık (Örn: ETH) **Short**, payda kısmındaki (Örn: XRP) **Long** yapılır.
    a. $Z < -2$ ise: Tam tersi işlem yapılır.
6. **Mean Reversion (Geri Dönüş):** Oran ortalamaya yaklaştığında her iki pozisyon da kapatılarak kâr realize edilir.

5. Sınav İçin Kritik "Hoca Notları"

- **Risk:** Mean Reversion stratejilerinin en büyük riski, iki varlık arasındaki kointegrasyonun kalıcı olarak bozulmasıdır (Örn: Bir şirketin iflas etmesi veya projeden büyük bir kötü haber gelmesi). Bu durumda fiyatlar asla ortalamaya dönmez.
- **Backtesting:** MATLAB kodundaki `entry_th` ve `exit_th` parametreleri optimize edilmelidir. Çok dar aralıklar komisyon maliyetini artırır, çok geniş aralıklar hiç işlem açmamaya sebep olur.
- **PHP Botu:** Botun `sleep(5)` süresi önemlidir. Çok hızlı sorgu yapmak borsa tarafından engellenmenize (IP Ban), çok yavaş sorgu yapmak ise kâr fırsatını kaçırmanıza neden olur.

Bu dersle ilgili herhangi bir hesaplama örneği (Z-Score bulma gibi) üzerinde çalışmak ister misin?

LECTURE 13

THE FOLDER 'mean_reversion_lecture_13' (PHP)

```php
<?php
require("bin_class.php");
//require("key_secret.php");
$key='';
$secret='';
$bnb=new Binance\binali($key, $secret, ['useServerTime'=>True]);
$bnb->useServerTime();

$ticker=$bnb->bookprices();
$buyprice=$ticker['ETHUSDT']['bidPrice'];
$sellprice=$ticker['ETHUSDT']['askPrice'];
echo "Buy Price: $buyprice\n";
echo "Sell Price: $sellprice\n";
$eth_ticker=$ticker['ETHUSDT'];
```

```
15. print_r($eth_ticker);
16.
17. $day_info=$bnb->prevDay();
18. print_r($day_info[5]);
19. $percent=$day_info[5]['priceChangePercent'];
20. echo "24h Change Percent: $percent%\n";
21. echo("price percent change=$percent%\n");
22.
23. for ($i=0;$i<count($day_info);$i++) {
24.     $percent=$day_info[$i]['priceChangePercent'];
25.     if ($percent > 20)
26.     {
27.         $symbol=$day_info[$i]['symbol'];
28.         echo ("<br>symbol=$symbol --percent== $percent%\n");
29.
30.     }
31. }
```

13. Dersimiz, önceki derste öğrendiğimiz **Mean Reversion** (Ortalamaya Dönüş) stratejisini bir adım öteye taşıyarak, **"Teknik Göstergeler ve Filtreleme"** (Technical Indicators and Filtering) ile nasıl daha güvenli hale getirebileceğimize odaklanıyor.

Sadece fiyata bakmak yerine, piyasanın "aşırı alım" veya "aşırı satım" durumunda olup olmadığını anlamak için ek araçlar kullanıyoruz.

1. Teorik Altyapı: Stratejiyi Güçlendirme

Önceki derste sadece iki varlığın oranına (Ratio) bakıyorduk. Ancak bazen oran (Ratio) bozulduğunda, bu bir "fırsat" değil, kalıcı bir trend değişikliği olabilir. 13. derste bu riski yönetmek için şu iki ana kavramı kullanıyoruz:

- **Bollinger Bands (Bollinger Bantları):** Fiyatın oynaklığını (volatilite) ölçer. Fiyat bandın dışına çıktığında "normale dönme" beklentisi artar.
- **RSI (Relative Strength Index):** Bir varlığın aşırı alınıp alınmadığını (Overbought > 70) veya aşırı satılıp satılmadığını (Oversold < 30) gösterir.

Strateji Mantığı: "Z-Score 2'den büyük olsun **VE** RSI 70'ten büyük olsun." Bu iki onay birleştiğinde işlemin başarı şansı çok daha yükselir.

2. PHP Kod Analizi (`mean_reversion_lecture_13.php`)

Bu dosya, stratejiyi bir "bot" haline getiren asıl motor parçasıdır. Önemli kod bloklarını ve ne yaptıklarını aşağıda bulabilirsin:

A. İstatistiksel Temel (365 Günlük Pencere)

Bot çalışmaya başladığında önce "normal" olanın ne olduğunu öğrenir:

```
PHP
// ETH ve XRP verilerini çekiyoruz
$data1 = $bnb->candlesticks('ETHUSDT', '1d', 365);
$data2 = $bnb->candlesticks('XRPUSDT', '1d', 365);

// Tarihsel Oran (Historical Ratio) Dizisi Oluşturma
for($i=0; $i < count($keys1); $i++){
    $close1 = $data1[$keys1[$i]]['close'];
    $close2 = $data2[$keys1[$i]]['close'];
    $ratio_series[] = $close1 / $close2;
}

// Ortalama ve Standart Sapma (Z-Score için gerekli)
$mean_ratio = array_sum($ratio_series) / count($ratio_series);
$std_dev = standard_deviation($ratio_series);
```

B. Canlı İzleme ve Karar Döngüsü

Botun kalbi olan `while(1)` döngüsü, her 5-10 saniyede bir piyasayı tarar:

```php
while(true) {
    $prices = $bnb->bookPrices();
    $current_eth = $prices['ETHUSDT']['bid'];
    $current_xrp = $prices['XRPUSDT']['bid'];

    // Güncel Oran ve Z-Score
    $current_ratio = $current_eth / $current_xrp;
    $z_score = ($current_ratio - $mean_ratio) / $std_dev;

    echo "Güncel Z-Score: " . round($z_score, 2) . "\n";

    // İŞLEME GİRİŞ (Entry)
    if ($z_score > 2 && $in_position == false) {
        // ETH pahalı, XRP ucuz -> ETH SAT, XRP AL
        $bnb->marketSell('ETHUSDT', $quantity_eth);
        $bnb->marketBuy('XRPUSDT', $quantity_xrp);
        $in_position = true;
    }

    // İŞLEMDEN ÇIKIŞ (Exit)
    if (abs($z_score) < 0.5 && $in_position == true) {
        // Oran normale döndü, pozisyonları kapat
        $bnb->marketBuy('ETHUSDT', $quantity_eth);
        $bnb->marketSell('XRPUSDT', $quantity_xrp);
        $in_position = false;
    }

    sleep(10); // Sunucuyu ve API'yi yormamak için bekleme
}
```

3. Sınavda Çıkabilecek Kritik Detaylar

I. Z-Score Formülü ve Yorumu

Sınavda formülü elle yazman istenebilir:

$$Z = \frac{Ratio_{current} - \mu_{ratio}}{\sigma_{ratio}}$$

- $Z > 2$: Üstteki varlık (Pay) çok pahalı, alttaki (Payda) ucuz.
- $Z < -2$: Üstteki varlık (Pay) çok ucuz, alttaki (Payda) pahalı.

II. İşlem Maliyetleri (Trading Costs)

Kodda `marketBuy` ve `marketSell` kullanılıyor. Sınavda şu soru gelebilir: **"Neden her Z-Score değişiminde işlem yapmıyoruz?"**

- **Cevap:** Her işlemde borsa komisyonu (**Commission**) ve fiyat kayması (**Slippage**) ödenir. Eğer elde edilecek kâr, komisyon maliyetinden küçükse zarar edilir. Bu yüzden bir "eşik" (Margin/Z-Threshold) kullanılır.

III. Güvenlik Mekanizmaları

`mean_reversion_lecture_13.php` dosyasında (ve ders notlarında) vurgulanan bir diğer nokta **Stop-Loss**'tur.

- Eğer Z-Score 2'den 3'e, oradan 5'e çıkıyorsa, bu "ortalamaya dönüş"ün bozulduğunu gösterir. Kodda bu durum için bir `if ($z_score > 4)` gibi bir acil çıkış (Stop-Loss) mekanizması eklenmesi gerektiği vurgulanır.

4. Özet Şema: Bot Nasıl Çalışır?

1. **Data Fetching:** Binance'den 1 yıllık veri al.
2. **Stat Calc:** Ortalama ve Standart Sapma hesapla.
3. **Real-time Monitoring:** Canlı fiyatları al, anlık Z-Score hesapla.
4. **Signal Check:** Z-Score > 2 mi? (Evetse işleme gir).
5. **Reversion:** Z-Score < 0.5 oldu mu? (Evetse kârı al ve çık).

Bir sonraki adım: Bu stratejiyi test etmek için kullanabileceğimiz **"Risk Management"** (Risk Yönetimi) hesaplamalarına geçmemi ister misin? Özellikle pozisyon büyüklüğü (Position Sizing) sınavda çok sorulur.

****NOTE:**

`bin_class.php` dosyası, algoritmik ticaret sisteminin **"iletişim merkezi"** veya **"altyapı katmanı"**dır. Yazdığın stratejinin (mantığın) borsa ile konuşabilmesini sağlayan bir **API Wrapper (API Bağlayıcısı)** kütüphanesidir.

Bu dosyanın ne işe yaradığını dört ana maddede özetleyebiliriz:

1. Borsaya Bağlantı ve Kimlik Doğrulama

Borsa (Binance) herkese açık veriler sunsa da, hesap bakiyeni görmen veya işlem yapman için kimliğini kanıtlaman gerekir. `bin_class.php`, senin **API Key** ve **Secret Key** bilgilerini

kullanarak her isteği dijital olarak imzalar. Bu sayede borsa, isteğin gerçekten senin hesabından geldiğinden emin olur.

2. Ham Veriyi Anlamlı Bilgiye Dönüştürme

Borsadan gelen veriler genellikle karmaşık JSON formatındadır. Bu dosya, o karmaşık veriyi çekip senin strateji kodunda (örneğin `mean_reversion.php`) kolayca kullanabileceğin dizilere veya nesnelere dönüştürür.

- **Örnek:** `$bnb->candlesticks('BTCUSDT', '1d')` komutunu verdiğinde, arka planda borsaya bağlanır, veriyi alır ve sana temiz bir liste olarak sunar.

3. Emir Gönderimi ve Yönetimi

Stratejin "Şimdi al!" dediğinde, bu komutun borsaya ulaşması için belirli bir formatta (fiyat, miktar, sembol, emir tipi vb.) paketlenmesi gerekir. `bin_class.php`, bu paketleme işlemini yapar ve emri borsaya iletir. Ayrıca emrin gerçekleşip gerçekleşmediğini (fill status) takip eder.

4. Karmaşıklığı Gizleme (Abstraction)

Yazılımcı olarak senin her seferinde ağ protokolleri, hata kodları veya imza algoritmalarıyla uğraşmanı engeller. Sen sadece yüksek seviyeli fonksiyonları çağırırsın:

- `marketBuy()`: Piyasa fiyatından al.
- `bookPrices()`: Anlık tahta fiyatlarını getir.
- `balances()`: Cüzdanımdaki parayı göster.

Özetle: Eğer stratejin (örneğin Mean Reversion) botun **beyni** ise, `bin_class.php` bu botun **elleri ve kulaklarıdır**. O olmadan strateji sadece kağıt üzerinde bir matematiksel model olarak kalır; borsaya dokunamaz.

İstersen bu dosyanın içindeki en kritik fonksiyonlardan birinin (örneğin emir gönderme fonksiyonunun) çalışma mantığını inceleyebiliriz?