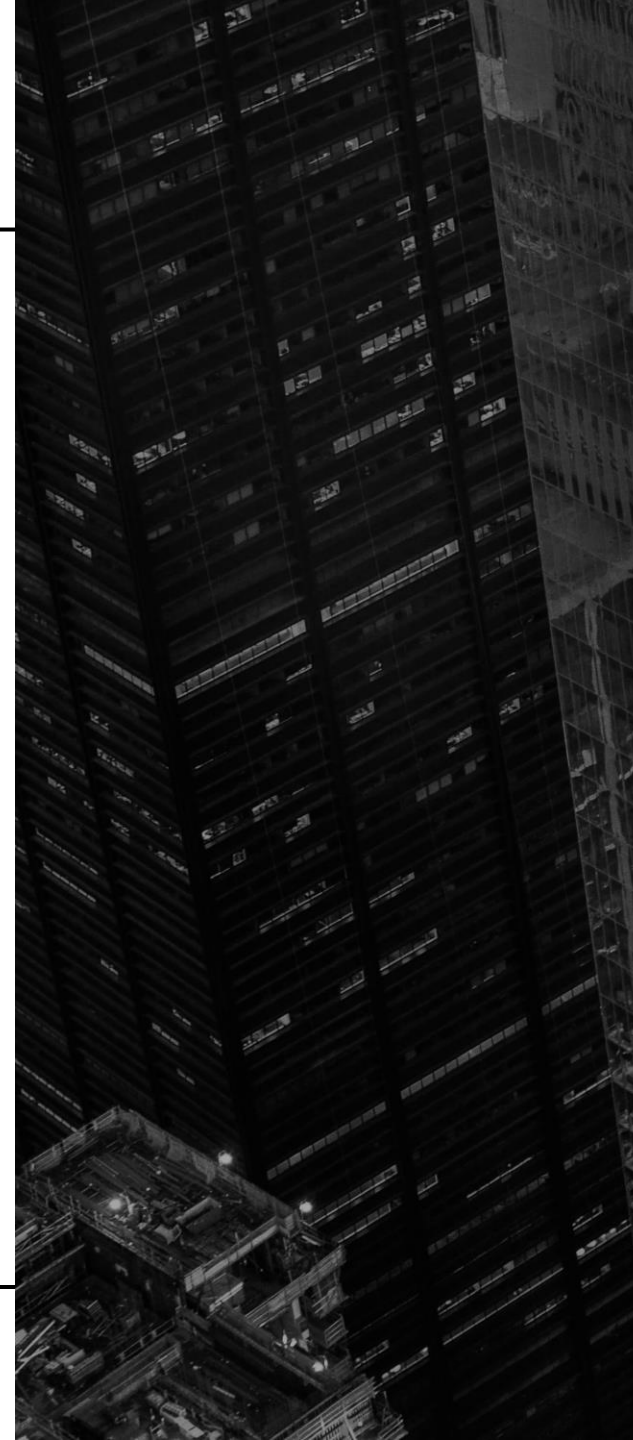


FI-blended Income Premium Balanced Index complementing Covered Call Strategies

Optimizing Call Option's Strike Price based on Volatility of Underlying Assets

Contents

1. Introduction
2. Data processing
3. Back-Testing & Implication
4. Limitations
5. Appendix



Content_I

Introduction of Covered Call

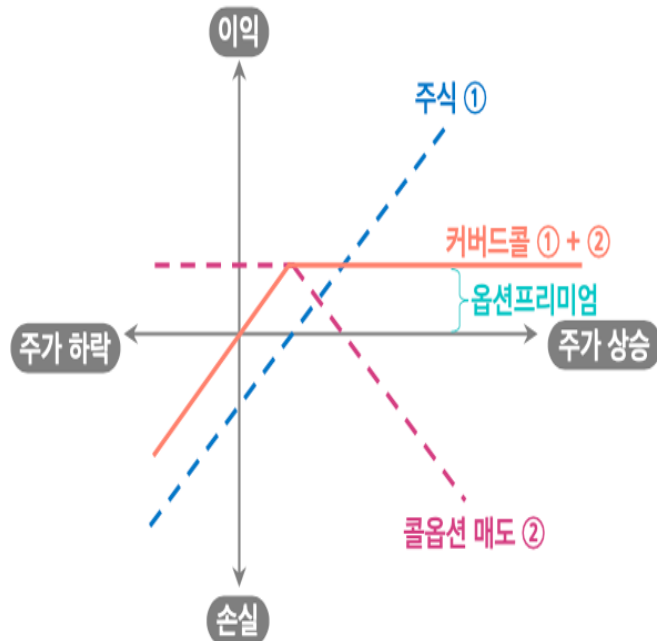
I. Introduction of Covered Call

[Covered Call]

기초자산을 보유하면서 동시에 해당 자산에 대한 콜옵션을 매도하여 옵션 프리미엄을 확보하는 방식

[Characteristics of Covered Call]

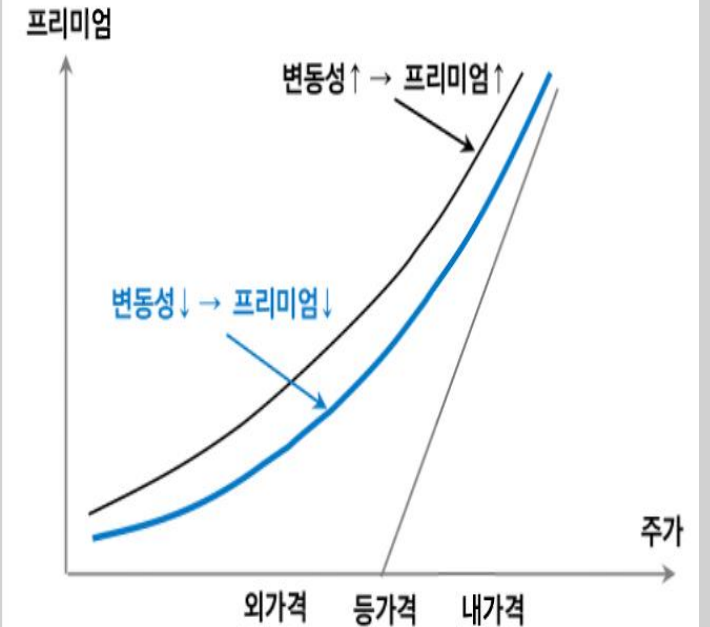
Covered Call Payoff



기초자산 횡보 시 유리



시장 변동성 상승 시 유리



I. Introduction of Covered Call

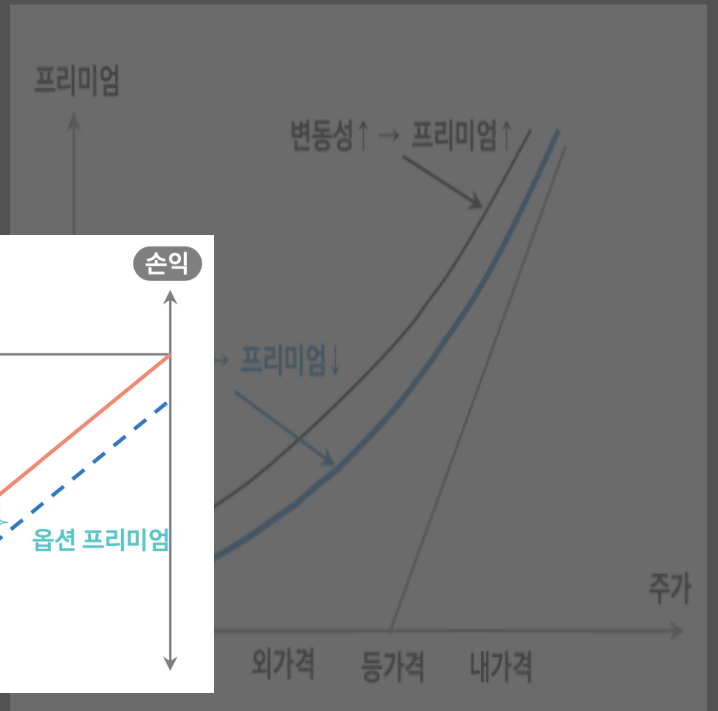
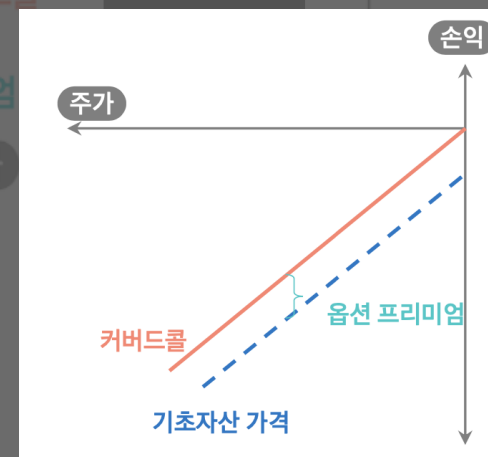
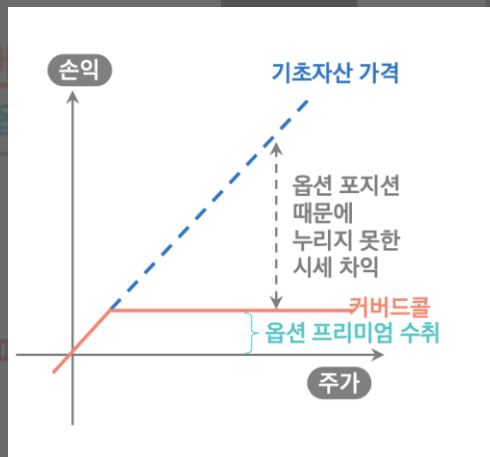
[Covered Call]

기초자산을 보유하면서 동시에 해당 자산에 대한 콜옵션을 매도하여 옵션 프리미엄을 확보하는 방식

[Characteristics of Covered Call]

본질적인 한계점 존재

1. 시장 큰 폭 상승 시 상승 잠재력 포기
2. 시장 큰 폭 하락 시 손실 방어력 제한



I. Introduction of Covered Call

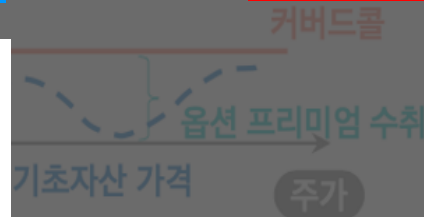
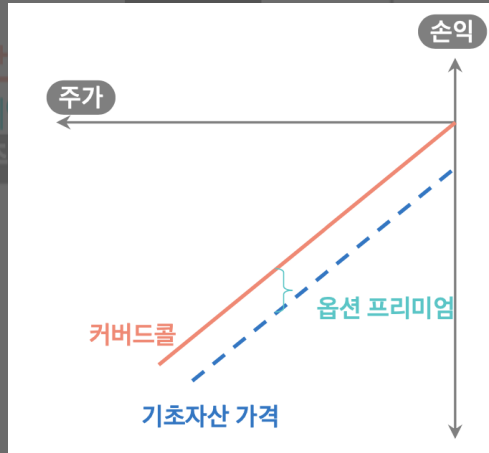
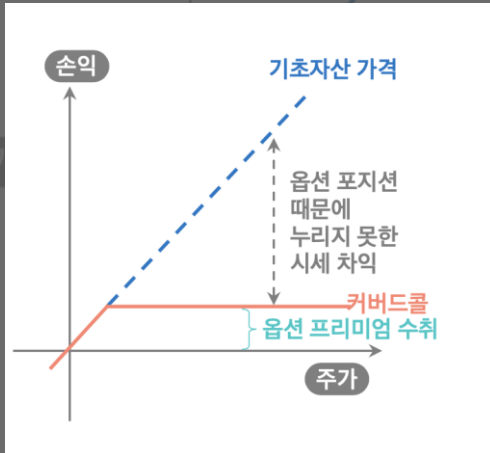
[Covered Call]

기초자산을 보유하면서 동시에 해당 자산에 대한 콜옵션을 매도하여 옵션 프리미엄을 확보하는 방식

[Characteristics of Covered Call]

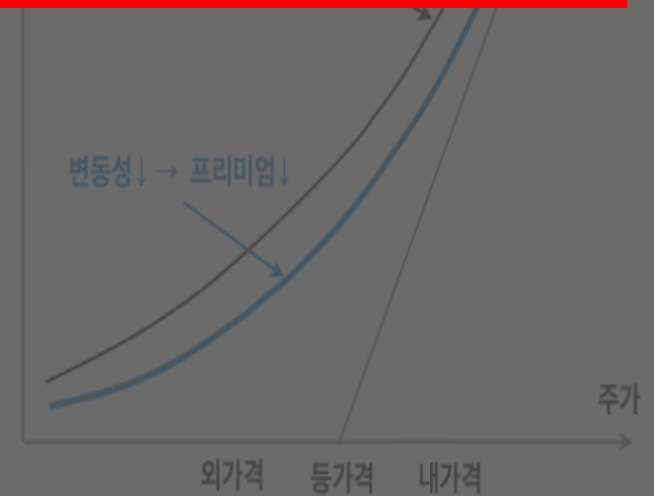
본질적인 한계점 존재

1. 시장 큰 폭 상승시 상승 잠재력 포기
2. 시장 큰 폭 하락시 손실 방어력 제한



Project Goal

1. 상승폭을 따라갈 수 있는 미스매칭 전략 활용
2. 하락 시 손실 방어를 위한 채권 혼합 전략 활용



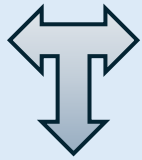
I. Introduction of Covered Call

[커버드콜 4 결정 요인]

Underlying Assets

주식형

채권형



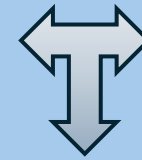
채권혼합형

고배당주 70% + 채권 30%

Call Option Maturity

Monthly Option

Daily Option



Weekly Option

Kospi200 Weekly Call Option

Proportion of Option

100% 매도



8 : 2 비중 일부 매도

미스매칭 전략

Strike Price

ATM



OTM

변동성 별 OTM 행사가격 최적화

Content_II

Data Processing

II. Data Processing

변동성에 따라 분류

포트폴리오 구성 비중 설정

콜옵션 행사가격 결정

채권 선정

Methodology

I. 변동성에 따른 1차 분류 진행

전체 고배당주 849개를 고변동성, 중변동성, 저변동성으로
카테고리별 유니버스 구성

II. 각 카테고리에서 배당수익률 상위 10개 종목을 선정

연 배당률을 기준으로 상위 10개 종목하여 포트폴리오 종목 구성

I. 안정성

하락장에서의 더 낮은 낙폭

II. 수익성

더 높은 월 배당금 지급 가능 & 금리 인하시 고배당주 매력도 증가

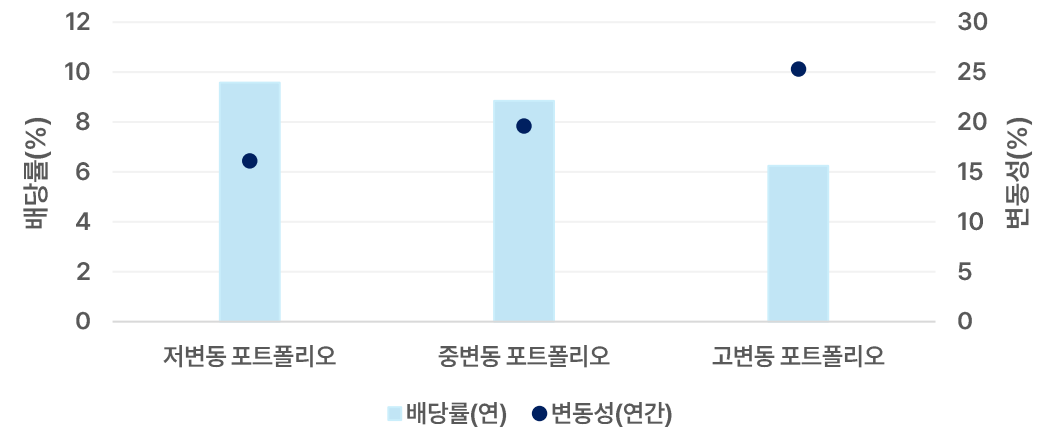
III. 세부 행사가격 설정 가능

변동성에 따른 세분화된 행사가격 설정 가능

Effect

저변동 고배당주 상위10개 종목	중변동 고배당주 상위10개 종목	고변동 고배당주 상위10개 종목
에스코홀딩스	에이블씨엔씨	동양생명
한국셀석유	넥스틸	동국홀딩스
현대엘리베이터	한샘	JB금융지주
동아타이어	한양증권	기아
삼성카드	하나투어	한국철강
기업은행	E1	하나금융지주
SK텔레콤	대신증권	삼화페인트공업
광주신세계	유안타증권	세아제강
HS애드	세아특수강	DB손해보험
아이마켓코리아	코리안리	SK가스

포트폴리오별 배당률 & 변동성



II. Data Processing

변동성에 따라 분류

포트폴리오 구성 비중 설정

콜옵션 행사가격 결정

채권 선정

Methodology

I. 옵션 선정

KOSPI200 Weekly Option (木)

II. 개별 주식 비중 설정

포트폴리오의 베타(β)가 KOSPI200의 베타(β)와 최대한

유사하게 비중 설정

각 종목의 비중은 5% ~ 20% 이내로 제한하여 분산효과 유지

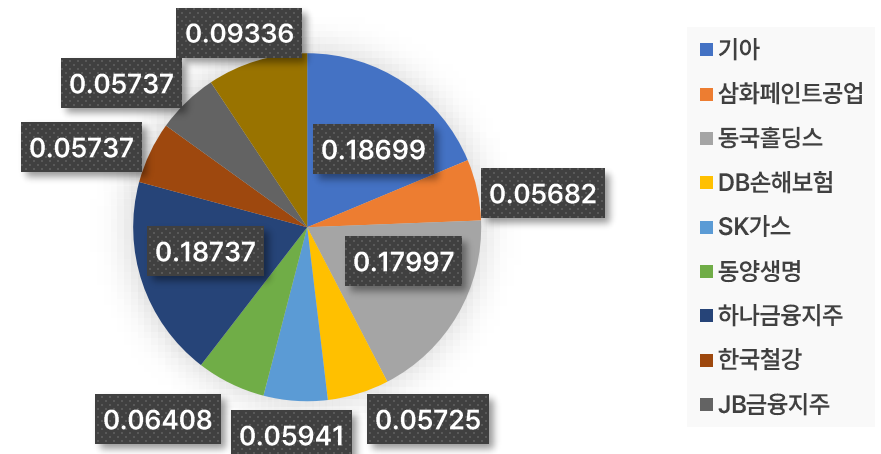
Effect

I. 헤지비율 조정 용이

II. 포트폴리오에 대응하는 옵션을 매도하는 효과 유지 가능

$$\beta_{\text{KOSPI}} = \sum_{i=1}^{10} \beta_i \times W_i$$

고변동 포트폴리오 비중 구성



II. Data Processing

변동성에 따라 분류

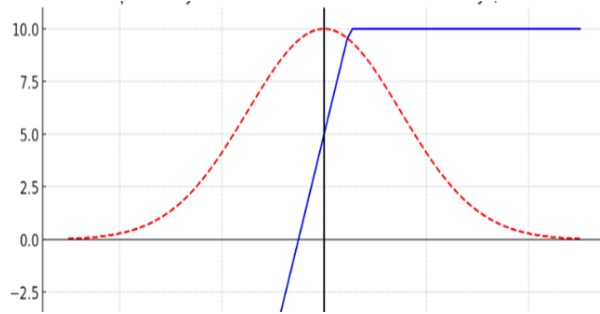
포트폴리오 구성 비중 설정

콜옵션 행사가격 결정

채권 선정

Methodology

기대수익률을 최대화하는 행사가격(K) 결정



$$\max_K \int_{-\infty}^{+\infty} N(x) \cdot f(x, K) dx$$

$f(x, K)$: covered call payoff

$N(x)$: pdf (normal distribution)

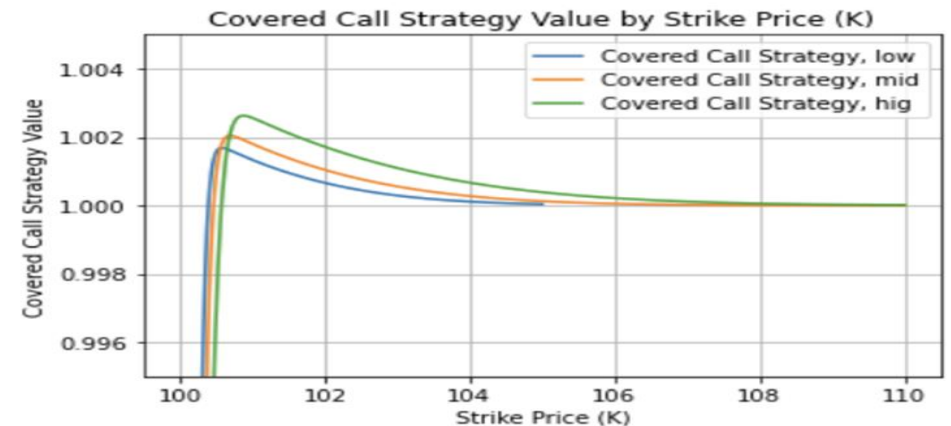
Result

I. 변동성이 클수록 높은 행사가의 옵션을 매도하는 것이

옳음을 확인

II. 각각의 최적의 행사가격(K)에서 옵션 매도 시 고변동성

포트폴리오의 기대수익률이 가장 높았음을 확인



II. Data Processing



Methodology

I. 채권 유니버스 선정
다양한 만기와 발행주체를 가지는 채권 16개 데이터 수집

II. 포트폴리오 편입 채권 선정
각각의 포트폴리오 수익률과 상관관계가 낮은 6개의 채권 선정 후 포트폴리오 편입

Effect

I. 분산 투자 효과
주식과 채권의 낮은 상관관계

II. 미래 기대수익률 예측 가능성 증가
채권의 만기 시점을 통해 가격이 명확히 결정

저변동 포트폴리오 채권 구성	중변동 포트폴리오 채권 구성	고변동 포트폴리오 채권 구성
국고채(3Y)	국고채(5Y)	국고채(3Y)
국고채(5Y)	국고채(3Y)	통안채(2Y)
통안채(2Y)	통안채(2Y)	국고채(5Y)
제1종 국민주택채권	국고채(10Y)	한전채(3Y)
국고채(10Y)	제1종 국민주택채권	제1종 국민주택채권
한전채(3Y)	한전채(3Y)	통안채(1Y)

Content_III

Back Testing & Implication

III. Back Testing & Implication

[변동성을 고려한 최적의 행사가격 결정 기준 제시]

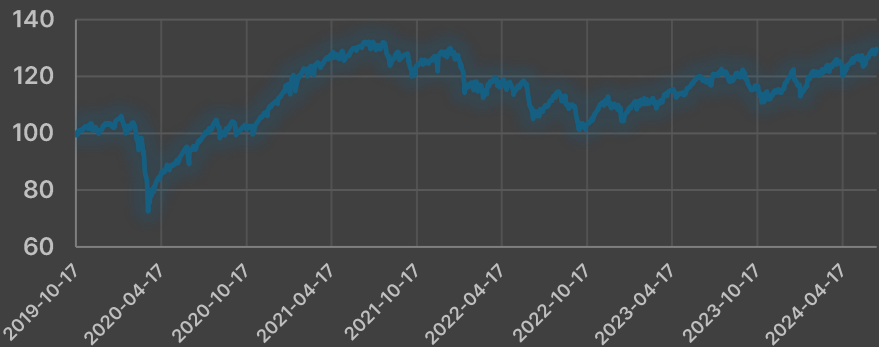
기존 커버드콜 지수의 행사가격 결정 방식

[KOSPI200 Weekly 커버드콜 ATM 지수]

✓ 지수개요

결제일 당일의 KOSPI200 종가지수보다
행사가격이 더 높은 KOSPI200 콜옵션 중 낮은
행사가격 순으로 두 종목 선정해 콜옵션 매도

KOSPI200WeeklyATM_CoveredCall_Index



최적화를 통한 커버드콜 지수의 행사가격 결정 방식

[기대수익률 최적화 결과]

I) Low Volatility Portfolio

▷ 최적의 행사가격(K) : 100.5% X ATM

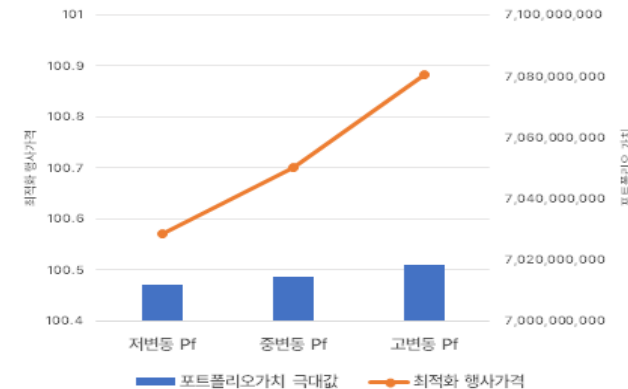
II) Middle Volatility Portfolio

▷ 최적의 행사가격(K) : 100.7% X ATM

III) High Volatility Portfolio

▷ 최적의 행사가격(K) : 100.9% X ATM

포트폴리오별 최적화 Strike Price 및 포트폴리오 가치



$$\max_K \int_{-\infty}^{+\infty} N(x) \cdot f(x, K) dx$$

III. Back Testing & Implication

[채권혼합형 인컴 밸런스드 지수 개발 방법론]

초기 투자금	10,000,000,000 (100억)
기간	2019.10.17 ~ 2024.06.28
기초자산 (7:3)	<ul style="list-style-type: none">➢ 고배당주 주식 포트폴리오 7,000,000,000 (70억)➢ 채권 3,000,000,000 (30억)➢ 변동성별로 3개의 포트폴리오 구성
콜옵션 비중 (8 : 2)	<ul style="list-style-type: none">➢ KOSPI200 Weekly Call Option 매도➢ 매주 목요일 옵션 청산 후 리밸런싱 진행➢ 포트폴리오마다 도출한 최적의 행사가격으로 콜옵션 매도

```
backtest_high, total_date_high, backtest_df_high = Backtest(stock, kospi_200, weekly_call, ticker_high, 100000000, 0.25, 0.3, 1.005)
backtest_mid, total_date_mid, backtest_df_mid = Backtest(stock, kospi_200, weekly_call, ticker_mid, 100000000, 0.25, 0.3, 1.007)
backtest_low, total_date_low, backtest_df_low = Backtest(stock, kospi_200, weekly_call, ticker_low, 100000000, 0.25, 0.3, 1.009)
```

[포트폴리오별 가치 계산 및 인덱스 계산 방법론]

채권 가치
+
배당주 포트폴리오 가치
+
현금 (콜옵션매도프리미엄 + 배당금)

1. $t = 1$, $YTM = r_1$:

$$= FV_1 \times \exp(-r_1 \times T)$$

2. $t = 2$, $YTM = r_2$:

$$= FV_2 \times \exp\left(-r_2 \left(T - \frac{1}{252}\right)\right)$$

:

$$\frac{2}{1} = \frac{\exp\left(-r_2 \left(T - \frac{1}{252}\right) \times \frac{1}{100}\right)}{\exp(-r_1 \times T \times \frac{1}{100})}$$
$$= \exp\left(-r_2 \left(T - \frac{1}{252}\right) \times \frac{1}{100} + r_1 \times T \times \frac{1}{100}\right)$$
$$= \exp\left(-r_2 \times T \times \frac{1}{100} + r_2 \times \frac{1}{252} \times \frac{1}{100} + r_1 \times T \times \frac{1}{100}\right)$$
$$= \exp\left(T \times \frac{(r_1 - r_2)}{100} + \frac{r_2}{25200}\right)$$

#채권 수익률로 할인해서 현재 가격 계산(연속복리)

```
bond_return_1 = np.exp(T*(ytm_1_yesterday - ytm_1)/100 + ytm_1/25200)
bond_return_2 = np.exp(T*(ytm_2_yesterday - ytm_2)/100 + ytm_2/25200)
bond_return_3 = np.exp(T*(ytm_3_yesterday - ytm_3)/100 + ytm_3/25200)
bond_return_4 = np.exp(T*(ytm_4_yesterday - ytm_4)/100 + ytm_4/25200)
bond_return_5 = np.exp(T*(ytm_5_yesterday - ytm_5)/100 + ytm_5/25200)
bond_return_6 = np.exp(T*(ytm_6_yesterday - ytm_6)/100 + ytm_6/25200)
```

III. Back Testing & Implication

[채권혼합형 인컴 밸런스드]

초기 투자금

기간

기초자산
(7:3)

- 고배
- 채권
- 변동

콜옵션 비중
(8:2)

- KOS
- 매주
- 포트
- 매도

[포트폴리오별 가치 계산 및]

채권 가치



배당주포트폴리오가치



현금
(콜옵션매도프리미엄 + 배당금)

1. $t = 1$, $YTM = r_1$:

$$= FV_1 \times \exp(-r_1 \times T)$$

2. $t = 2$, $YTM = r_2$:

$$= FV_2 \times \exp\left(-r_2 \left(T - \frac{1}{252}\right)\right)$$

:

$$\frac{2}{1} = \frac{\exp\left(-r_2 \left(T - \frac{1}{252}\right) \times \frac{1}{100}\right)}{\exp\left(-r_1 \times T \times \frac{1}{100}\right)}$$

$$= \exp\left(-r_2 \left(T - \frac{1}{252}\right) \times \frac{1}{100} + r_1 \times T \times \frac{1}{100}\right)$$

$$= \exp\left(-r_2 \times T \times \frac{1}{100} + r_2 \times \frac{1}{252} \times \frac{1}{100} + r_1 \times T \times \frac{1}{100}\right)$$

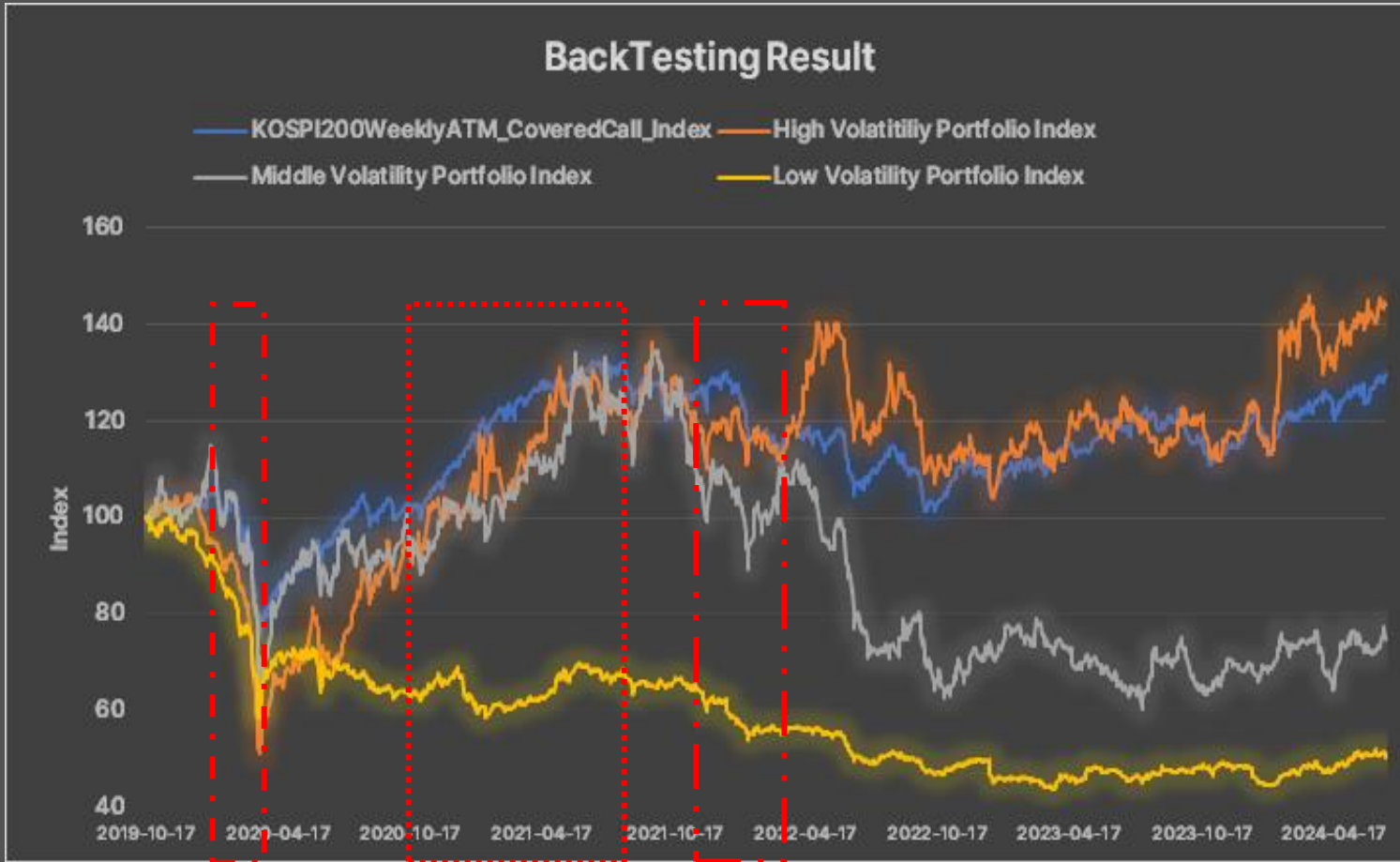
$$= \exp\left(T \times \frac{(r_1 - r_2)}{100} + \frac{r_2}{25200}\right)$$

```
cker_high, 100000000, 0.25, 0.3, 1.005)
r_mid, 100000000, 0.25, 0.3, 1.007)
r_low, 100000000, 0.25, 0.3, 1.009)
```

```
1)/100 + ytm_1/25200)
2)/100 + ytm_2/25200)
3)/100 + ytm_3/25200)
4)/100 + ytm_4/25200)
5)/100 + ytm_5/25200)
6)/100 + ytm_6/25200)
```


III. Back Testing & Implication

[백테스팅을 통한 시장국면별 특징 분석 및 최적화 커버드콜 전략 효용성 분석]



Project Goal

1. 콜옵션 일부 비중 매도(8:2 미스매칭) 전략은 유용한가?
2. 하락장에서의 손실 방어를 위한 채권혼합 전략은 유용한가?

Content_IV

Limitations

IV. Limitations

[배당금 계산의 한계]

- 고배당주의 연배당률 일 평균으로 환산하는 과정에서 1/252 단순평균을 적용해 계산함으로써 실제 배당주기 반영 X
- Weekly 콜옵션 매도를 통한 프리미엄을 일 평균으로 환산하는 과정에서 1/5로 단순평균화 적용

[β 계산의 정밀도 개선 필요성]

- 개별 종목의 β 산출 시 장기간(5년) 동안의 데이터 사용
- 시장과 반대 방향으로 움직이는 종목들 (음수 β) 적절히 반영 X
- 베타 산출 기간을 더 짧게 설정했다면 보다 정확한 비중 조정과 성과 개선이 가능했을 것



[채권 관련 가정의 현실성 부족 및 분산 효과 미흡]

- 채권 10년 만기 무이표채로 가정
- 채권의 만기, 쿠폰이자, 롤오버 등 중요요소 고려 X
- 주식 포트폴리오가 이미 분산투자 효과 내재 & 데이터 분석 기간동안 채권과 주식의 동조화 경향 강화
- 채권과의 음의 상관관계가 유의미 X

[옵션의 내재변동성 반영 부족]

- 실제 전략 구현시 옵션의 내재변동성을 활용한 기초자산변동성 예측을 통해 최적 행사가 도출해야 함
- 내재변동성이 아닌 과거 실제 데이터를 활용하여 변동성 계산
- 실제 적용가능한 전략으로의 정교화 필요성

Content_V

Appendix

V. Appendix

[개별 주식의 β 산출 코드]

```
file_path = "C://Users//user//OneDrive//바탕 화면//통합 문서1.xlsx"
df = pd.read_excel(file_path)

kospi = yf.download('^KS11', start=df['날짜'].min(), end=df['날짜'].max())
t_bond_10y = yf.download('^TNX', start=df['날짜'].min(), end=df['날짜'].max())

kospi['Market_Return'] = kospi['Adj Close'].pct_change()

portfolio_columns = ['주식A', '등락률']
betas = {}

for portfolio in portfolio_columns:
    merged_data = pd.merge(df[['날짜', portfolio]], kospi[['Market_Return']], left_on='날짜', right_index=True)

    merged_data = merged_data.dropna(subset=[portfolio, 'Market_Return'])

    covariance = np.cov(merged_data[portfolio], merged_data['Market_Return'])[0, 1]

    market_variance = np.var(merged_data['Market_Return'])

    # 베타 계산
    beta = covariance / market_variance
    betas[portfolio] = beta

for portfolio, beta in betas.items():
    print(f"{stock}의 베타 값: {beta}")
```

V. Appendix

[KOSPI200지수 β 와 일치하는 포트폴리오별 개별 주식 비중 조정 코드]

분산 효과를 위해 주식종목 별 최소 5% 비중, 최대 20% 비중 제약 하 최적 비중 산출

```
def objective(coeffs, key):
    betas = list(data[key].values())
    return (sum(coeffs[i] * betas[i] for i in range(len(betas))) - target)**2

def constraint(coeffs):
    return sum(coeffs) - 1

initial_guess = [0.05] * 10

bounds = [(0.05, 0.2)] * 10

cons = {'type': 'eq', 'fun': constraint}

solution_저변동 = minimize(objective, initial_guess, args=("저변동",), bounds=bounds, constraints=cons)
solution_중변동 = minimize(objective, initial_guess, args=("중변동",), bounds=bounds, constraints=cons)
solution_고변동 = minimize(objective, initial_guess, args=("고변동",), bounds=bounds, constraints=cons)

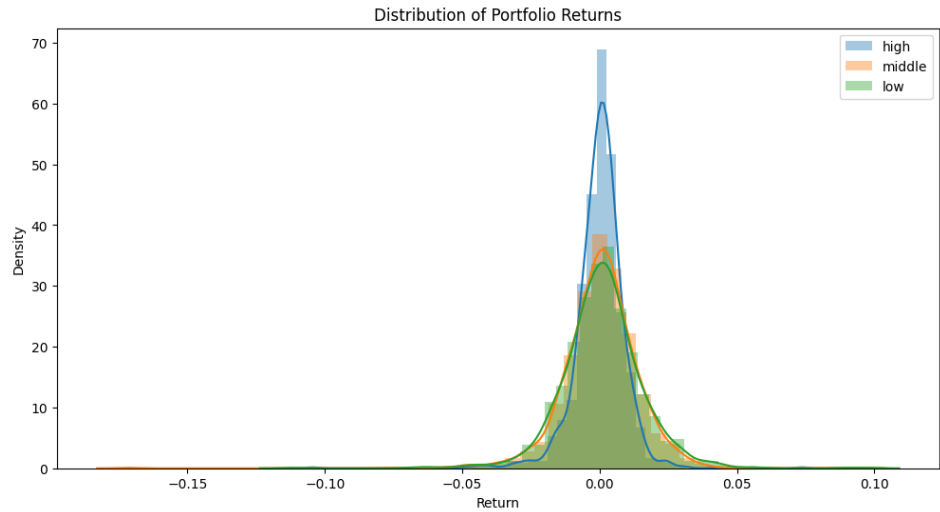
print("저변동에 대한 찾은 계수들:", solution_저변동.x)
print("중변동에 대한 찾은 계수들:", solution_중변동.x)
print("고변동에 대한 찾은 계수들:", solution_고변동.x)

betas_저변동 = list(data["저변동"].values())
betas_중변동 = list(data["중변동"].values())
betas_고변동 = list(data["고변동"].values())

eq1_result = sum(solution_저변동.x[i] * betas_저변동[i] for i in range(len(betas_저변동)))
eq2_result = sum(solution_중변동.x[i] * betas_중변동[i] for i in range(len(betas_중변동)))
eq3_result = sum(solution_고변동.x[i] * betas_고변동[i] for i in range(len(betas_고변동)))
```

V. Appendix

[기초자산의 수익률 분포가 정규분포임을 증명하는 PLOT]



[행사가격 K에 대한 최적화 함수식 코드]

```
# 블랙숄스 프리미엄 계산 함수
def black_scholes_call(S, K, T, r, sigma):
    d1 = (np.log(S / K) + (r + 0.5 * sigma**2) * T) / (sigma * np.sqrt(T))
    d2 = d1 - sigma * np.sqrt(T)
    call_price = S * norm.cdf(d1) - K * np.exp(-r * T) * norm.cdf(d2)
    return call_price

# 적분 계산 함수
def calculate_integral(sigma, S, K, T, r):
    def pdf(x):
        return norm.pdf(x, loc=0, scale=sigma)

    p = black_scholes_call(S, K, T, r, sigma)

    integral_value, _ = quad(lambda x: ((x + S + p*0.25) * pdf(x)) if (x + S) < K else (((x+S)*0.75 + p*0.25) * pdf(x)), -np.inf, +np.inf)
    return integral_value

def optimize_strike(S, T, r, sigma, K):
    K_values = np.linspace(S, S * K, 500)
    integral_results = [calculate_integral(sigma, S, K, T, r) for K in K_values]
    covered_call = np.array(integral_results) / S
    return covered_call, K_values

# 최대값을 찾는 함수
def find_max(covered_call, K_values):
    max_value = np.max(covered_call)
    max_index = np.argmax(covered_call)
    return max_value, K_values[max_index]
```

V. Appendix

[포트폴리오 가치 계산 코드 (리밸런싱 주기 반영)]

```
255 #리밸런싱 날짜가 아닌 경우
256 if i not in rebalancing_date:
257
258     current_portfolio_price = current_stock_price * stock_weight + bond_price
259
260     #현재 가치를 전체 포트폴리오 가치 리스트에 추가
261     portfolio_price.append(current_portfolio_price)
262
263 #리밸런싱 날짜인 경우
264 else:
265
266     #현재 ksp1200 가격(증가)
267     current_kospi_price = Ksp1200_price(i)
268
269     #새로운 콜 옵션 행사가 결정
270     target_strike_price = Call_option_strategy(current_kospi_price, vol)
271
272     new_option_short, new_strike_price = Option_price(i, target_strike_price)
273
274     strike_price_list.append(new_strike_price)
275
276     #시작날은 이전에 매도한 옵션이 없기 때문에 큰 행사가를 설정하여 옵션이 행사될 경우를 제외시킴.
277     if rebalancing_count == 0:
278
279         last_strike_price = 100000000000000
280
281     else:
282         #지난주 매도한 콜옵션 가격(증가)
283         last_strike_price = strike_price_list[rebalancing_count-1]
284
285     #현재 포트폴리오 가격 = 현재 기초자산 총가격 + 새로 매도한 옵션 프리미엄 + 채권 가격
286     current_portfolio_price = current_stock_price * stock_weight + new_option_short * call_option_weight + bond_price
```

```
#Kospi 200 콜 옵션이 행사되었을 경우
if current_kospi_price >= last_strike_price:

    #콜 옵션 행사로 인한 손실
    call_option_loss = (current_kospi_price - last_strike_price) * call_option_weight

    #감소해야 하는 기초자산의 비중 계산(잃은 손실만큼 반올림해서 제외)
    new_stock_weight = stock_weight - round(call_option_loss/current_stock_price)

    #손실이 자산 전체 가격보다 클 때: 파산
    if new_stock_weight <= 0:
        break

    #감소한 기초자산의 비중만큼 콜 옵션 매도 비중 조절
    call_option_weight = round(call_option_weight * (new_stock_weight/stock_weight))

    #옵션을 더 이상 매도할 수 없을 때: 전략 중지
    if call_option_weight < 1:
        break

    #감소한 기초자산 적용
    stock_weight = new_stock_weight

    #현재 포트폴리오 총 가격
    current_portfolio_price = current_portfolio_price - call_option_loss
```


[

E.o.D

]