



FMBA 2018 사전 교육

프로그래밍(엑셀 VBA)

KAIST

경영공학과 석박사통합과정 유승현

rambor12@business.kaist.ac.kr

목 차

1. 금융과 프로그래밍
2. 엑셀의 기초
3. 엑셀 해 찾기
4. 엑셀 VBA
5. 엑셀 VBA 응용 : Black-Scholes Option Pricing

Github Repository

□ 프로그래밍 교육 자료

https://github.com/Seung-hyeon/KAIST_FMBA_Programming_2018

This repository Search Pull requests Issues Marketplace Explore

Seung-hyeon / KAIST_FMBA_Programming_2018 Watch 0 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Insights Settings

KAIST FMBA Programming Prior-Education Materials Edit

Add topics

5 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

Seung-hyeon some files		Latest commit 871ad68 2 minutes ago
Excel VBA	some files	2 minutes ago
Matlab	some files	2 minutes ago
R	some files	2 minutes ago
README.md	some files	2 minutes ago

README.md

KAIST FMBA Programming 2018

This is KAIST FMBA Programming Prior-Education Materials.

This materials cover Excel VBA , Matlab , and R .

자료 다운로드

프로그래밍 사용 예

- 크롤링
- 알고리즘 트레이딩
- 이미지 인식

https://github.com/anuragmishracse/caption_generator

1. 금융과 프로그래밍

□ 금융의 특성

- ① 대부분의 자료가 **수치화** 되어 있음.
- ② 이미 **데이터베이스**가 잘 구축 되어 있음.
- ③ 데이터로부터 **새로운 정보**를 찾아내는 것이 중요.
→ 통계, 프로그래밍 기법을 사용

□ 왜 프로그래밍이 필요한가?

- ① 방대한 자료를 **빠르고 효율적으로** 처리.
- ② **실수없이 안정적**으로 일을 처리.
- ③ **어렵고 복잡한** 문제를 쉽게 풀기.

1. 금융과 프로그래밍

□ 프로그래밍(Programming)이란?

문제를 정의하고, 그 문제를 해결할 수 있는 프로그램을 만드는 모든 과정

참고

프로그램(Program) : 컴퓨터에 의해 실행되는 명령어 집합

코딩(Coding) : 실제로 프로그램 만드는 것

□ 프로그래밍 언어(Programming Language)의 종류는?

- ① 접근성 : **엑셀 VBA**
- ② 수치 계산(공학) : **MATLAB**
- ③ 통계 분석 : SAS, EViews, R 등
- ④ 다목적 : C, C++, Java, Python 등



1. 금융과 프로그래밍

□ 금융 데이터베이스

해외 : 블룸버그(Bloomberg), 로이터(Reuter), WRDS, Optionmetric 등

국내 : DataGuide, KIS-Value, QuantWise 등

기타 : 증권사 API, DDE 등

□ 참고 도서



엑셀 VBA 바이블

John Walkenbach 저 | 구미숙, 미현곤 역 | 프리렉 | 2010.11.05.

★★★★★ 10.0 | 네티즌리뷰[2건]

소개 - 독자대상 : **엑셀** 중고급 이상 숙련자 - 구성 : **엑셀** 매크로 기록, 사용자 지향 응용프로그램, 유틸리티 개발 - 특징 : 1) **VBA**를 처음 접하는 독자에게도 이해가 편한 내용과 서술 방식 2) **VBA**를 이해하기 위해 적합한 예제들을 다양하게 수록 3) **엑셀 VBA** 전 영역을 포괄하는 주제들로 구성 4) **엑셀** 2007에...



R언어로 짜는 금융프로그래밍

장기천, 강병진 저 | 서울경제경영 | 2015.12.30.

★★★★★ 0.0 | 네티즌리뷰[0건] | 도서구매 25,000원 → 23,750원(-5%)

소개 ▶ 이 책은 **R언어로 짜는 금융프로그래밍**에 대해 다룬 이론서입니다. R언어 및 금융프로그래밍의 기초적이고 전반적인 내용을 학습할 수 있도록 구성했습니다.

필요한 것만 인터넷 검색하는게 더 유용!
— stackoverflow.com



1. 금융과 프로그래밍

□ 프로그래밍 기본 구조 (Edsger W. Dijkstra's Structured Programming 참조)

1. 변수의 선언 :
(Declaration)

어떤 타입의 데이터를 사용할 것인가?

2. 조건문 :
(Selection)

조건에 따라 어떤 코드를 실행할 것인가?

3. 반복문 :
(Iteration)

이 코드를 언제까지 반복할 것인가?

위의 구조를 이용하여

- 데이터를 어떻게 구조화하고(Data Structure),
 - 어떤 방법을 사용하여(Algorithm),
- 문제를 해결할까?

1. 금융과 프로그래밍

1. 변수의 선언 : (Declaration)

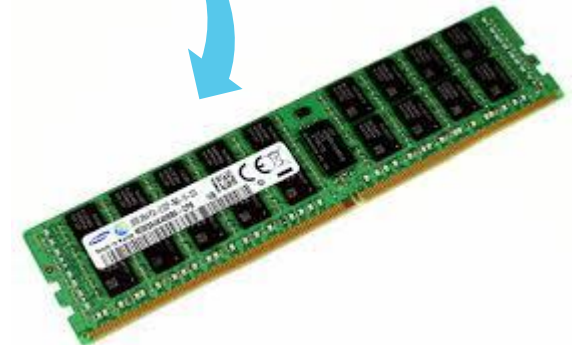
어떤 타입의 데이터를 사용할 것인가?

□ 기초 데이터 타입(Primitive Type)

데이터형	설명	크기(비트)	최소값	최대값
byte	부호있는 정수	8비트	-128	127
int	부호있는 정수	32비트	-2147483648	2147483647
long	부호있는 정수	64비트	-9223372036854775808	9883372036854775807
float	실수	32비트	약 $\pm 1.4 \times 10^{-45}$ (유효숫자 7개)	약 $\pm 3.4 \times 10^{38}$ (유효숫자 7개)
double	실수	64비트	약 $\pm 4.9 \times 10^{-324}$ (유효숫자 15개)	약 $\pm 1.7 \times 10^{308}$ (유효숫자 15개)
char	문자(유니코드)	16비트	'\u0000'(0)	'\uFFFF'(65535)
boolean	True 또는 False	8비트	False	True

변수 할당!

(e.g.) int a = 3



주소(16진수)	값
0xffffffff	
...	
0x80000000	
...	
0x508a0210	3





해당하는 주소 값에 3을 할당

1. 금융과 프로그래밍

1. 변수의 선언 : (Declaration)

어떤 타입의 데이터를 사용할 것인가?

□ 기초 데이터 타입(Primitive Type) 각 언어별 예

데이터형	값	C	 python	 Excel VBA	 MATLAB	
int	3	int a=3;	a=3	Dim a As Integer = 3	a=3;	a=3
long	3	long a=3;	a=3	Dim a As Long = 3	a=3;	a=3
float	3.3	float a=3.3;	a=3.3	Dim a As Single = 3.3	a=3.3;	a=3.3
double	3.3	double a=3.3;	a=3.3	Dim a As Double = 3.3	a=3.3;	a=3.3
char	'A'	char a='A';	a='A'	Dim a As String = 'A'	a='A';	a='A'
boolean	True	사용자가 지정	a=True	Dim a as Boolean = True	a=true;	a=TRUE

1. 금융과 프로그래밍

1. 변수의 선언 : (Declaration)

어떤 타입의 데이터를 사용할 것인가?

□ 참조 데이터 타입(Reference Type)

변수 할당!

(e.g.) `int a[10] = [0,1,2,3,4,5,6,7,8,9]`

데이터형	설명	크기(비트)	최소값	최대값
list or array	배열 혹은 리스트	제한 없음	제한 없음	제한 없음
class	클래스	제한 없음	제한 없음	제한 없음
function	함수	제한 없음	제한 없음	제한 없음



여기서부터 실제 데이터 시작





주소(16진수)	값
0xffffffff	
0x01118708	데이터 시작 주소 값을 할당 참조! 0x508a01b0
...	
0x508a01b0	0
0x508a01d0	1
0x508a01f0	2
0x508a0210	3
0x508a0230	4
0x508a0250	5
0x508a0270	6
0x508a0290	7
0x508a02b0	8
0x508a02d0	9

1. 금융과 프로그래밍

1. 변수의 선언 : (Declaration)

어떤 타입의 데이터를 사용할 것인가?

□ 참조 데이터 타입(Reference Type) 각 언어별 예

데이터형	값	C	 python™	 Excel VBA	 MATLAB	
list or array	[1,2,3]	<code>int a[3]=[1,2,3];</code>	<code>a=[1,2,3]</code>	<code>Dim a(3) As Integer</code> <code>a(1) = 1</code> <code>a(2) = 2</code> <code>a(3) = 3</code>	<code>a=[1,2,3];</code>	<code>a=c(1,2,3)</code>
class				생략		
function	add	<code>int add(int a, int b){</code> <code> return a+b;</code> <code>}</code>	<code>def add(a, b):</code> <code> return a+b</code>	<code>Function add(a, b)</code> <code> add = a+b</code> <code>End Function</code>	<code>function c = add(a, b)</code> <code> c = a+b;</code> <code>end</code>	<code>add <- function(a, b){</code> <code> return(a+b)</code> <code>}</code>

1. 금융과 프로그래밍

1. 변수의 선언 : (Declaration)

어떤 타입의 데이터를 사용할 것인가?

□ 연산자(Operator) : 데이터를 연산하는 함수 예) +, -, ×, ÷,

산술 연산자

비교 연산자

연산자	설명
+	더하기
-	빼기
*	곱하기
/	나누기
^ or **	거듭제곱
==	같은가?
!= or ~=	다른가?
>, <	왼쪽(혹은 오른쪽)이 큰가?
>=, <=	왼쪽(혹은 오른쪽)이 크거나 같은가?

비교 연산자

(기타) 연산자





연산자	설명
& or &&	and, 그리고
or	or, 또는
~ or ^	not, 역
= or <-	값 대입
++	1씩 증가
--	1씩 감소
sizeof or len	데이터 길이

1. 금융과 프로그래밍

2. 조건문 : (Selection)

조건에 따라 어떤 코드를 실행할 것인가?

- 조건의 참, 거짓에 따라 다른 계산을 실행
주로 If ... else 형태의 문법을 사용

C	 python™	 Excel VBA	 MATLAB	
<pre>if(a==3){ printf("a is 3"); }else if(a==3.3{ printf("a is 3.3"); }else{ printf("a is ?"); }</pre>	<pre>if a==3: print('a is 3') elif a==3.3: print('a is 3.3') else: print('a is ?')</pre>	<pre>If a = 3 Then MsgBox "a is 3" Elseif a = 3.3 Then MsgBox "a is 3.3" Else MsgBox "a is ?"</pre>	<pre>if a==3 disp('a is 3') elseif a==3.3 disp('a is 3.3') else disp('a is ?') end</pre>	<pre>if(a==3){ print('a is 3') } else if(a==3.3){ print('a is 3.3') } else { print('a is ?') }</pre>

1. 금융과 프로그래밍





3. 반복문 : (Iteration)

이 코드를 언제까지 반복할 것인가?

□ 같은 명령을 반복해서 실행

반복 횟수가 정해질 경우 : for 루프

반복 횟수가 정해지지 않았을 경우 : while 루프

C	 python™	 Excel VBA	 MATLAB	
<pre>int a = 3; for(i=0;i<=2;i++){ a = a + i; }</pre>	<pre>a = 3 for i in range(3): a = a + i</pre>	<pre>a = 3 For i=0 To 2 a = a + i Next</pre>	<pre>a = 3; for i = 0:2 a = a + i end</pre>	<pre>a = 3 for (i in 0:2){ a = a + i }</pre>

추가 : MATLAB 설치

- 다음 실습을 위하여 MATLAB을 미리 설치

엑셀 VBA



2. 엑셀의 기초

□ 셀의 구조

① 내용

- 값(Value) : 문자열(String), 숫자(Int), 논리값(Boolean) 등
- 수식 (Formula) : "="로 시작함

② 서식 : 글꼴, 정렬, 테두리, 색깔 등

□ 상대 참조와 절대 참조 (\$ 표시로 구분)

- ① 상대 참조 : 수식을 복사하면 셀 참조는 새로운 위치에 맞게 변경.
- ② 절대 참조 : 수식 복사해도 셀 참조는 바뀌지 않음.

	상대 참조 "=A3+A2"	절대 참조 "=A3+A\$2"
-1		
1	0	0
2	3	1
3	5	2
4	7	3

2. 엑셀의 기초

□ 예제1 포트폴리오 평균 수익률과 분산 구하기

① 평균수익률

$$E(R_p) = \sum_{i=1}^N w_i E(R_i)$$

② 분산

$$Var(R_p) = \sum_{i=1}^N \sum_{j=1}^N w_i w_j Cov(R_i, R_j)$$

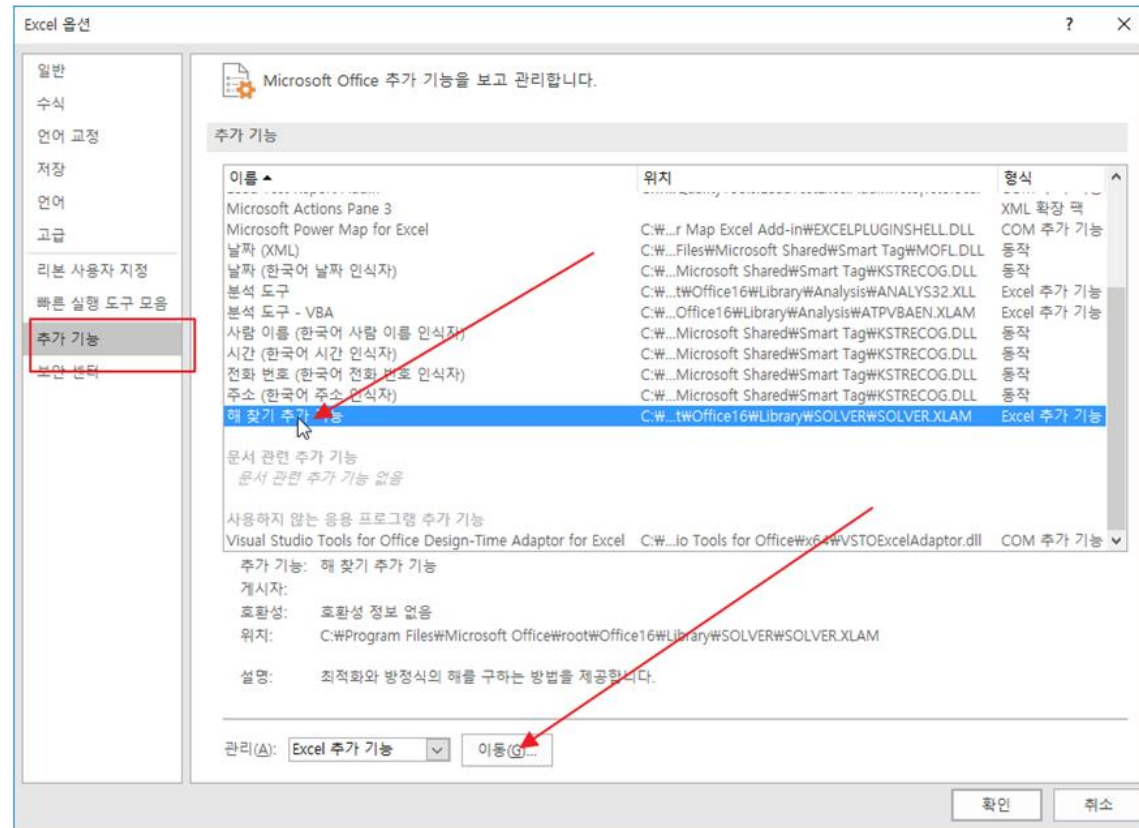
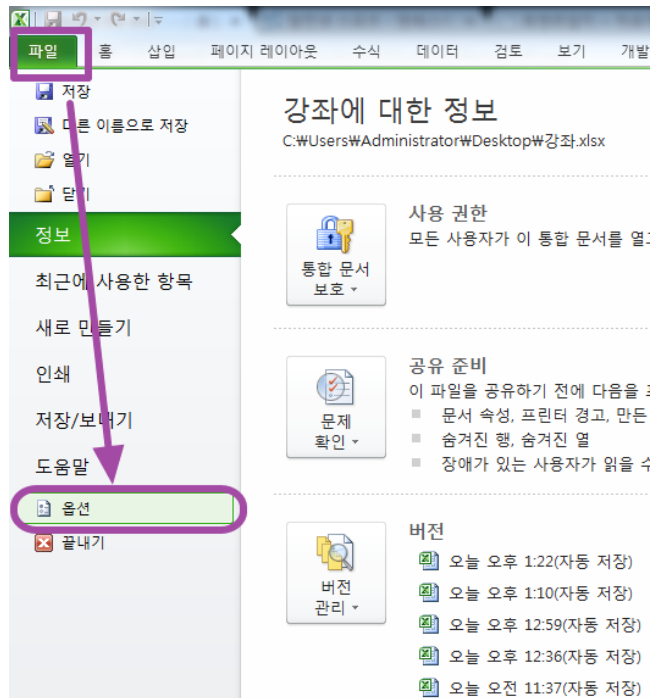
Hint

$$Cov(R_i, R_j) = mmult(transpose(\overrightarrow{R_{i,t}}), \overrightarrow{R_{j,t}}) / T$$

종목	비중
삼성전자	50%
POSCO	30%
현대중공업	10%
셀트리온	10%

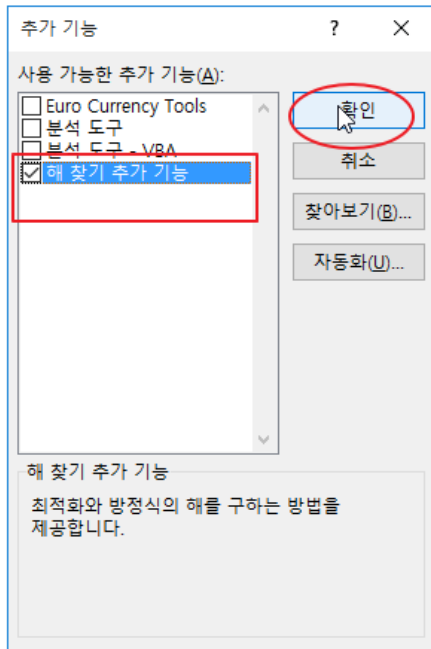
3. 엑셀 해 찾기

□ 해 찾기 추가



3. 엑셀 해 찾기

□ 해 찾기 추가



3. 엑셀 해 찾기

□ 해 찾기

The image shows an Excel spreadsheet with the Solver dialog box open. Red arrows indicate the mapping between spreadsheet cells and the Solver fields:

- 변수 (Variable):** Cell B4 contains 'X', which is mapped to the '변수 셀 변경 (B)' field in the Solver dialog.
- 목적함수 (Objective Function):** Cell C6 contains the formula $X^2 + 2X$, which is mapped to the '목표 설정 (T)' field in the Solver dialog.
- 제한조건 (Constraint):** Cell C8 contains the formula $3X^3 \leq 0$, which is mapped to the '제한 조건에 종속 (U)' field in the Solver dialog.

The Solver dialog box is titled '해 찾기 매개 변수' and contains the following settings:

- 목표 설정 (T):** \$C\$6
- 대상:** ☒ 최대값 (M) ☐ 최소 (N) ☐ 지정값 (V) 0
- 변수 셀 변경 (B):** \$C\$4
- 제한 조건에 종속 (U):** \$C\$8 <= 0
- ☒ 제한되지 않는 변수를 음이 아닌 수로 설정 (K)
- 해법 선택 (E):** GRG 비선형
- 해법:** 완전한 비선형으로 구성된 해 찾기 문제에 대해서는 GRG Nonlinear 엔진을 선택합니다. 비선형 문제에 대해서는 LP Simplex 엔진을 선택하고 완전하지 않은 비선형으로 구성된 해 찾기 문제에 대해서는 Evolutionary 엔진을 선택합니다.
- Buttons:** 추가 (A), 변화 (C), 삭제 (D), 모두 재설정 (R), 읽기/저장 (L), 도움말 (H), **해 찾기 (S)** (highlighted with a red circle), 닫기 (Q).

3. 엑셀 해 찾기

□ 예제2 해찾기.

- ① 아래 목적함수를 최소화하는 X, Y 찾기.

$$X^2 - XY + Y^2 - 3X$$

#Answer

X=2, Y=1일 때, 목적함수= -3

- ② 제약 조건이 있을 때 해 찾기

$$X^2 + Y^2 + Z^2$$

제약식 :

$$2X + Y - 5 \leq 0$$

$$X + Z - 2 \leq 0$$

$$-X + 1 \leq 0$$

$$-Y + 2 \leq 0$$

$$-Z \leq 0$$

#Answer

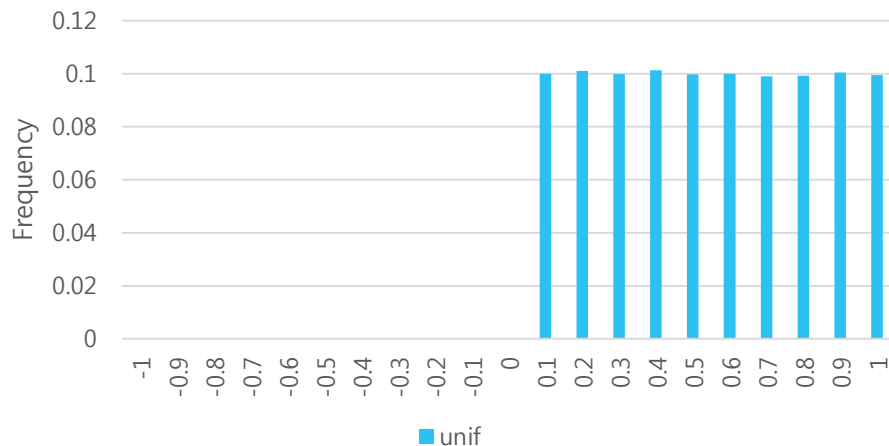
X=1, Y=2, Z=0 일 때, 목적함수=5

3. 엑셀 해 찾기

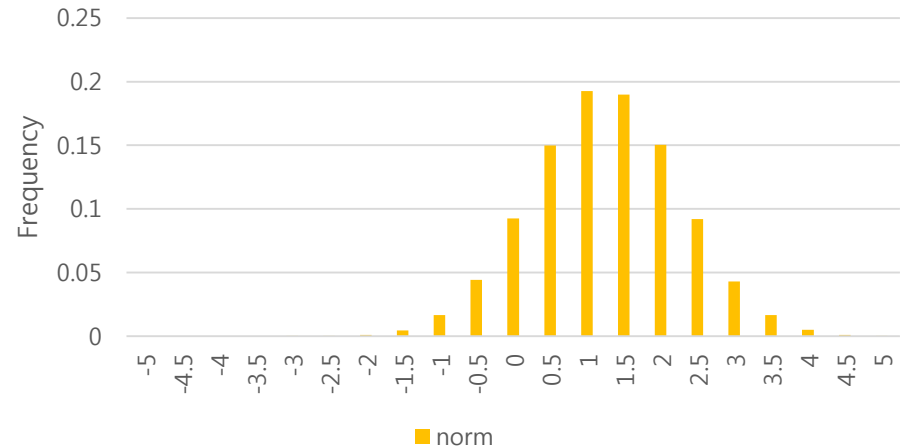
□ 예제3 난수 생성하기

- ① `rand()`, `norm.inv()`, `frequency()`의 활용
 - 균등분포(Uniform distribution)을 따르는 난수 생성하기
 - 정규분포(Normal distribution)을 따르는 난수 생성하기분포를 그려보자.

Uniform Distribution (0~1)



Normal Distribution (mean=1, std=1)

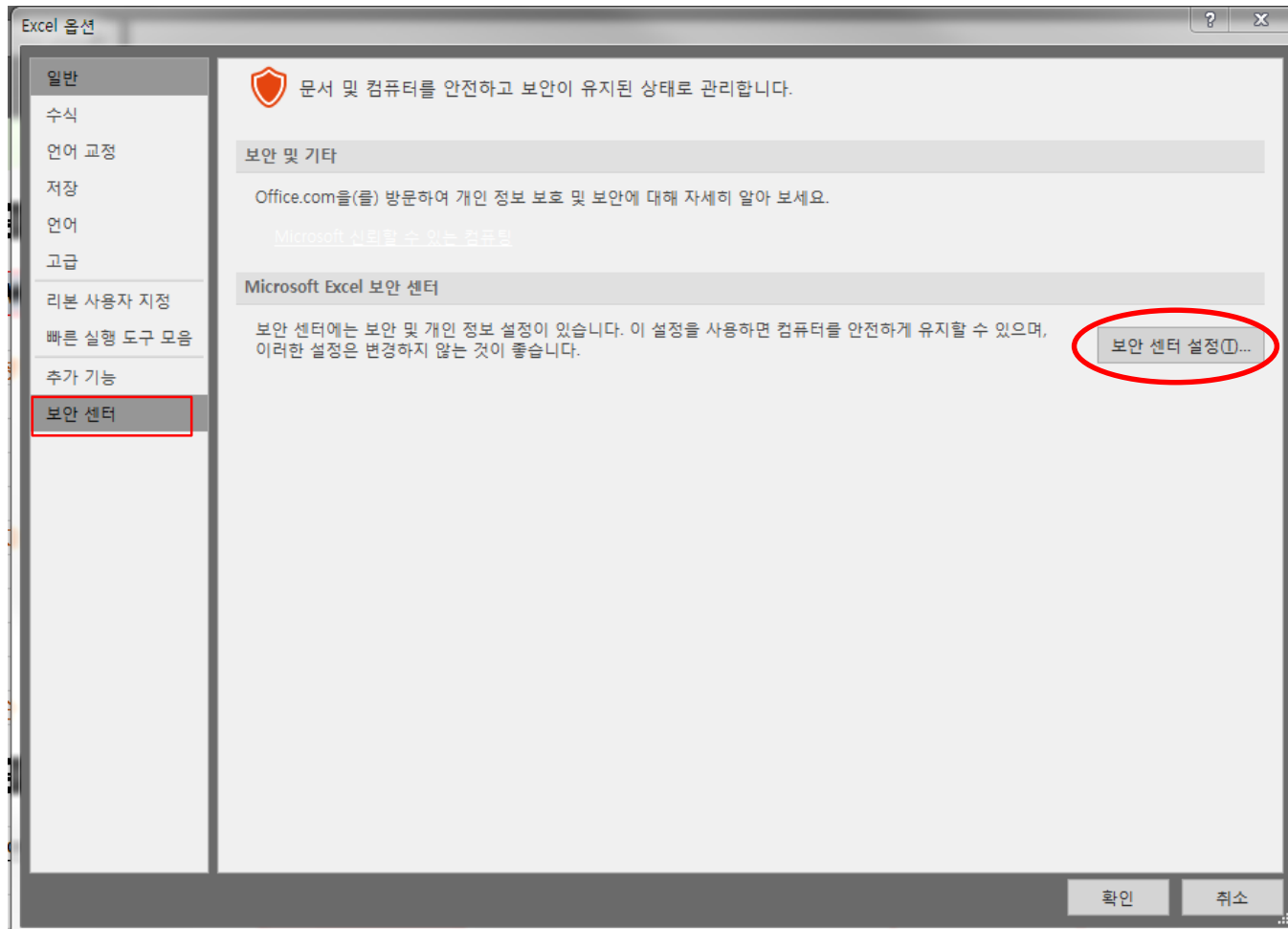


4. 엑셀 VBA

- **매크로** : 자주 사용하는 여러 개의 명령어를 묶어서 하나의 키 동작이나 메뉴 등을 사용해 실행되도록 만든 기능
(Alt + F11로 편집기 실행)
- **VBA** (Visual Basic for Application) : 오피스 각 제품의 패키지 별 특징을 기존 Visual Basic에 추가한 것 (일종의 프로그래밍 언어)

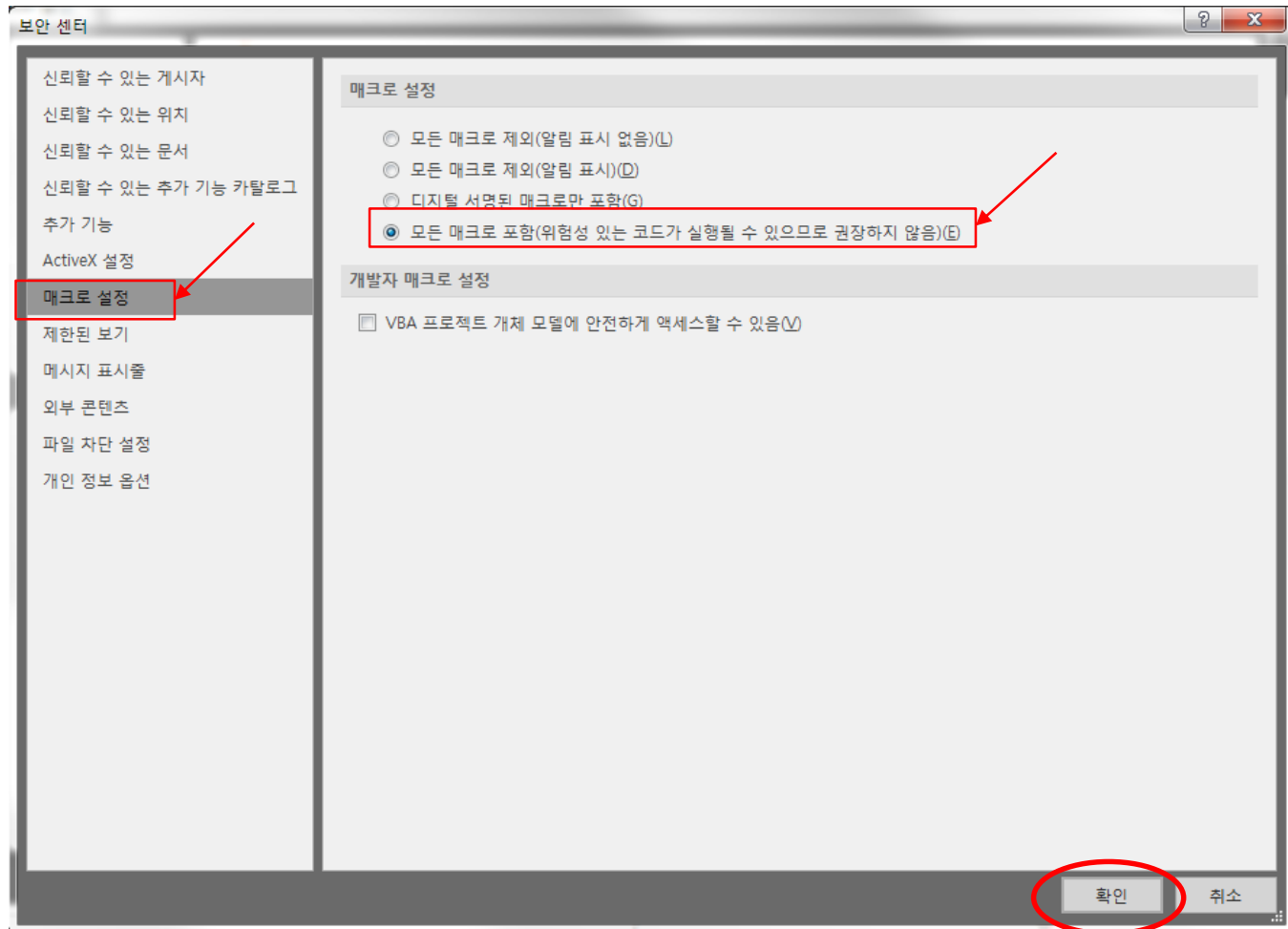
4. 엑셀 VBA

□ 매크로 설정



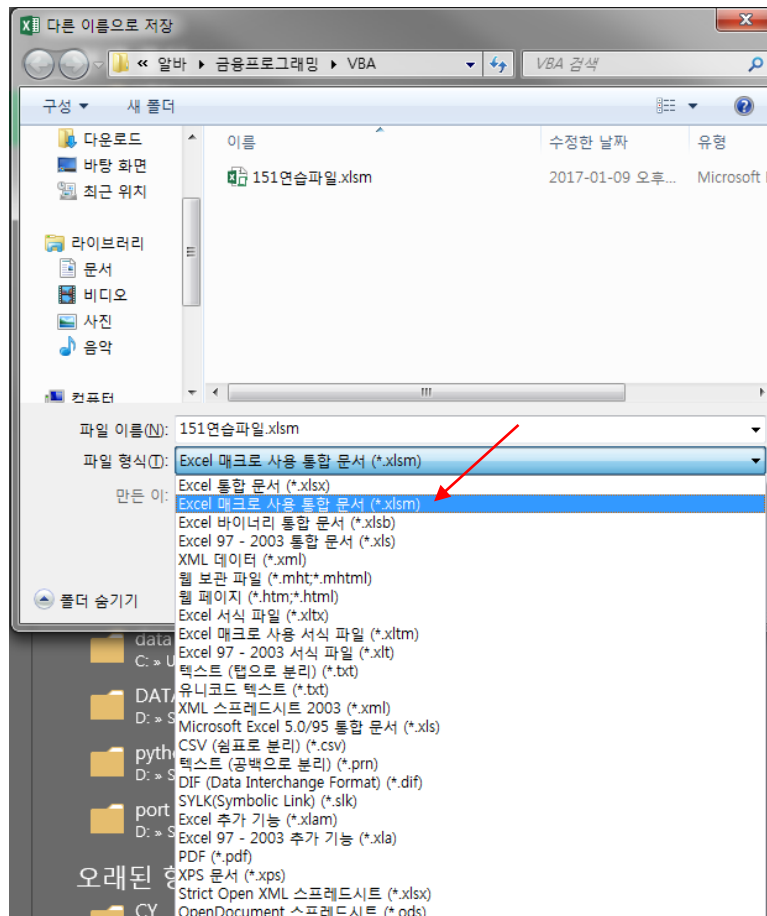
4. 엑셀 VBA

□ 매크로 설정



4. 엑셀 VBA

□ 매크로 파일 저장



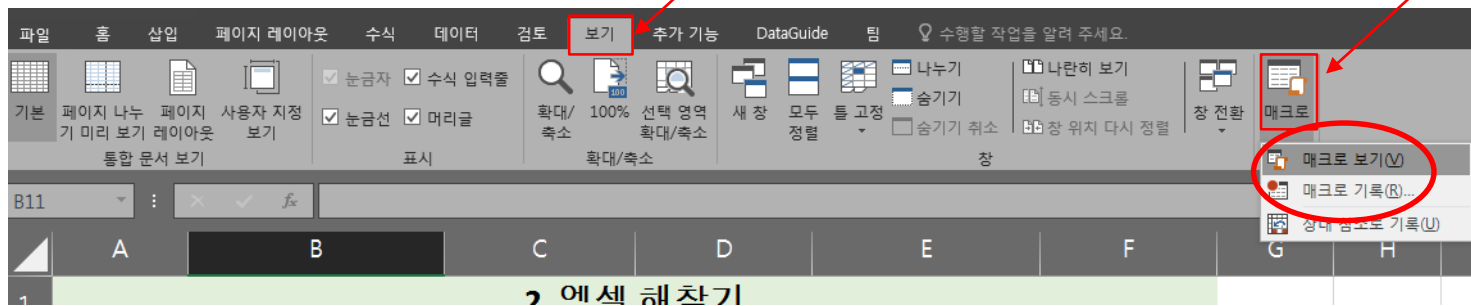
4. 엑셀 VBA

□ 매크로 코딩 및 실행

① 단축키

단축키	내용
Alt + F11	VBA 창
Alt + F8	매크로 선택창

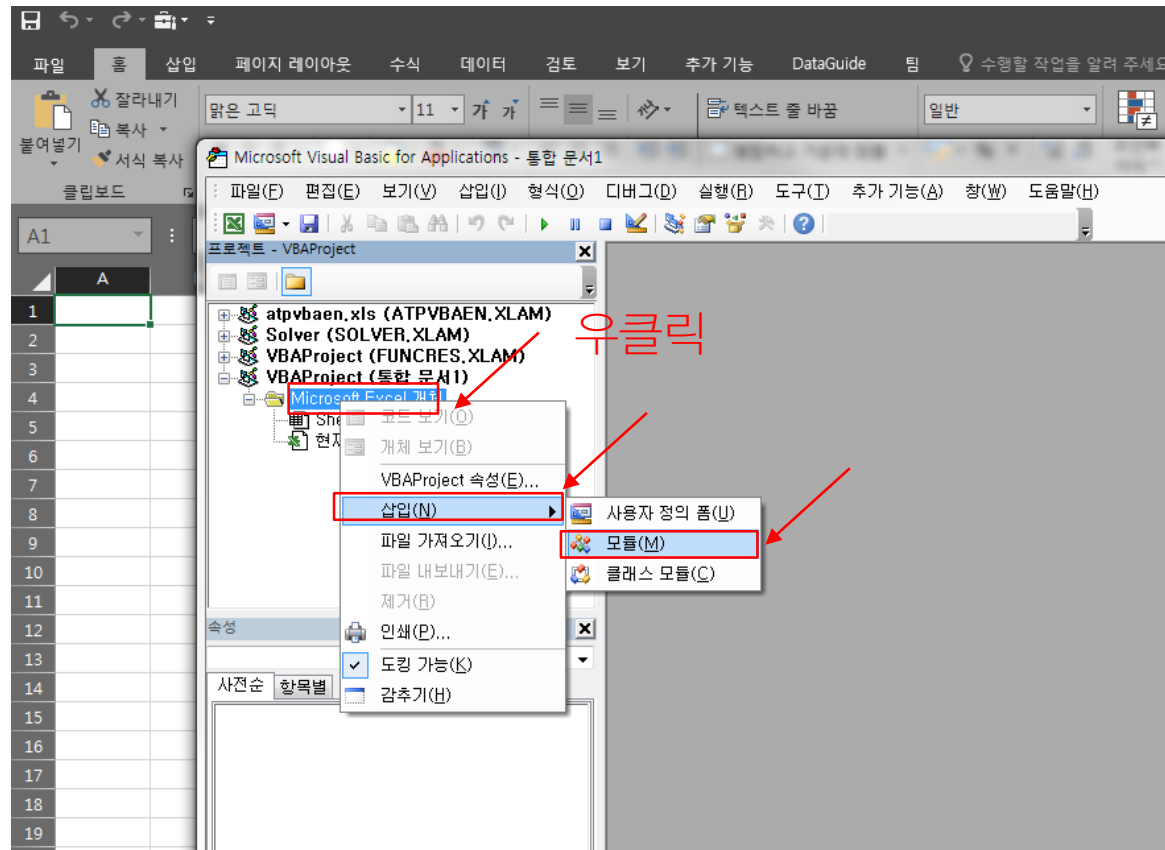
② 매크로 보기 및 기록



4. 엑셀 VBA

□ 매크로 코딩 및 실행

③ VBA 편집기 - 새로운 모듈 생성



4. 엑셀 VBA

□ 엑셀 VBA 기초 문법

1. 변수의 선언 : (Declaration)

어떤 타입의 데이터를 사용할 것인가?

a라는 '정수형' 변수 선언
`Dim a As Integer`

b라는 '문자열' 변수 선언
`Dim b As String`

c라는 '실수형' 변수 선언
`Dim c As Double`

d라는 '논리형' 변수 선언
`Dim d As Boolean`

변수의 이름

변수의 타입

4. 엑셀 VBA

□ 엑셀 VBA 기초 문법

2. 조건문 : (Selection)

조건에 따라 어떤 코드를 실행할 것인가?

If문은 조건을 보고 참인지 거짓인지를 판단하여, 참인 경우에 명령문을 실행

```
If [조건1] Then  
    [명령문]
```

```
Elseif [조건2] Then  
    [명령문]
```

```
Else  
    명령문
```

```
End If
```

실행 조건

실행할 명령문

Example

```
a = 1
```

```
If a < 0 Then  
    b = -1
```

```
Elseif a > 1 Then  
    b = 1
```

```
Else  
    b = 0
```

실행하면 b=0 이 실행됨

4. 엑셀 VBA

□ 엑셀 VBA 기초 문법

3. 반복문 : (Iteration)

이 코드를 언제까지 반복할 것인가?

반복 조건

For문은 반복적인 작업을 위해 사용하년 명령문, 일정한 횟수만큼 반복 수행

For 변수 = 시작값 To 종료값 Step [변화값]

[명령문]

If [탈출 조건] Then
Exit For
End If

Next

실행할 명령문

탈출 조건

Example

j = 0

For i = 0 To 100 Step 2

j = j + i

If i > 4 Then
Exit For
End If

Next

실행하면 j=12 이 실행 됨

4. 엑셀 VBA

□ 엑셀 VBA 기초 문법

함수 만들기

함수(function)은 어떤 기능을 수행하는 명령문, 같은 기능을 여러번 쓸 때 유용

Function 함수이름(함수인자들)

[명령문]

함수이름 = 함수값

End Function

실행할 명령문

함수의 결과값

Example

Function add(x, y)

z = x + y

add = z

End Function

x = 3

y = 7

z = add(x, y)

실행하면 z=12

5. 엑셀 VBA 응용 : Black-Scholes Option Pricing

□ Black-Scholes(-Merton) Model

- ✓ Fisher Black, Myron Scholes, (Robert C. Merton)이 1973년 제시
- ✓ 1997년에 노벨 경제학상
- ✓ 세상을 바꾼 10대 방정식 (출처 : <http://biz.heraldcorp.com/view.php?ud=20160308000285>)
- ✓ 파생상품(특히 European Option)의 가격 결정 편미분방정식
- ✓ 옵션(Option) : 미래 특정 시점(T)에 특정 상품(S)을 정해진 가격(K)에 사고(call) 팔(put) 권리

옵션 가격

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} - rV = 0$$

기초 자산 변동성

무위험 이자율

기초 자산 가격

5. 엑셀 VBA 응용 : Black-Scholes Option Pricing

□ Black-Scholes(-Merton) Model

옵션 가격

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS \frac{\partial V}{\partial S} - rV = 0$$

기초 자산 변동성

무위험 이자율

기초 자산 가격

방정식의 해(Solution)

표준정규분포 누적 함수

$$Call \ Price = SN(d_1) - Ke^{-r\tau} N(d_2)$$

$$Put \ Price = Ke^{-r\tau} \{1 - N(d_2)\} - S \{1 - N(d_1)\}$$

만기까지 남은 기간

$$where, \ d_1 = \frac{\ln S/K + (r + \frac{1}{2}\sigma^2)\tau}{\sigma\sqrt{\tau}}$$

$$d_2 = d_1 - \sigma\sqrt{T-t}$$

5. 엑셀 VBA 응용 : Black-Scholes Option Pricing

□ 예제4 Black-Scholes Option Pricing

변수 설명	코드 이름	값
기초 자산 가격	S	150
옵션 행사 가격	K	150
무위험 이자율	r	0.05
만기까지 남은 기간	tau	0.0833
기초자산의 변동성	sigma	0.25

5. 엑셀 VBA 응용 : Black-Scholes Option Pricing

□ **예제4** Black-Scholes Option Pricing (Sample Code)

```
Function BSPrice(S, K, r, tau, sigma, call_put)

    # Sample Code
    d1 = (Log(S / K) + (r + 0.5 * sigma ^ 2) * tau) / (sigma * Sqr(tau))
    d2 = d1 - sigma * Sqr(tau)
    Nd1 = Application.NormSDist(d1)
    Nd2 = Application.NormSDist(d2)

    If call_put = "call" Then
        BSPrice = S * Nd1 - K * Exp(-r * tau) * Nd2

    ElseIf call_put = "put" Then
        BSPrice = K * Exp(-r * tau) * (1 - Nd2) - S * (1 - Nd1)

    End If

End Function
```

End

Question & Comment