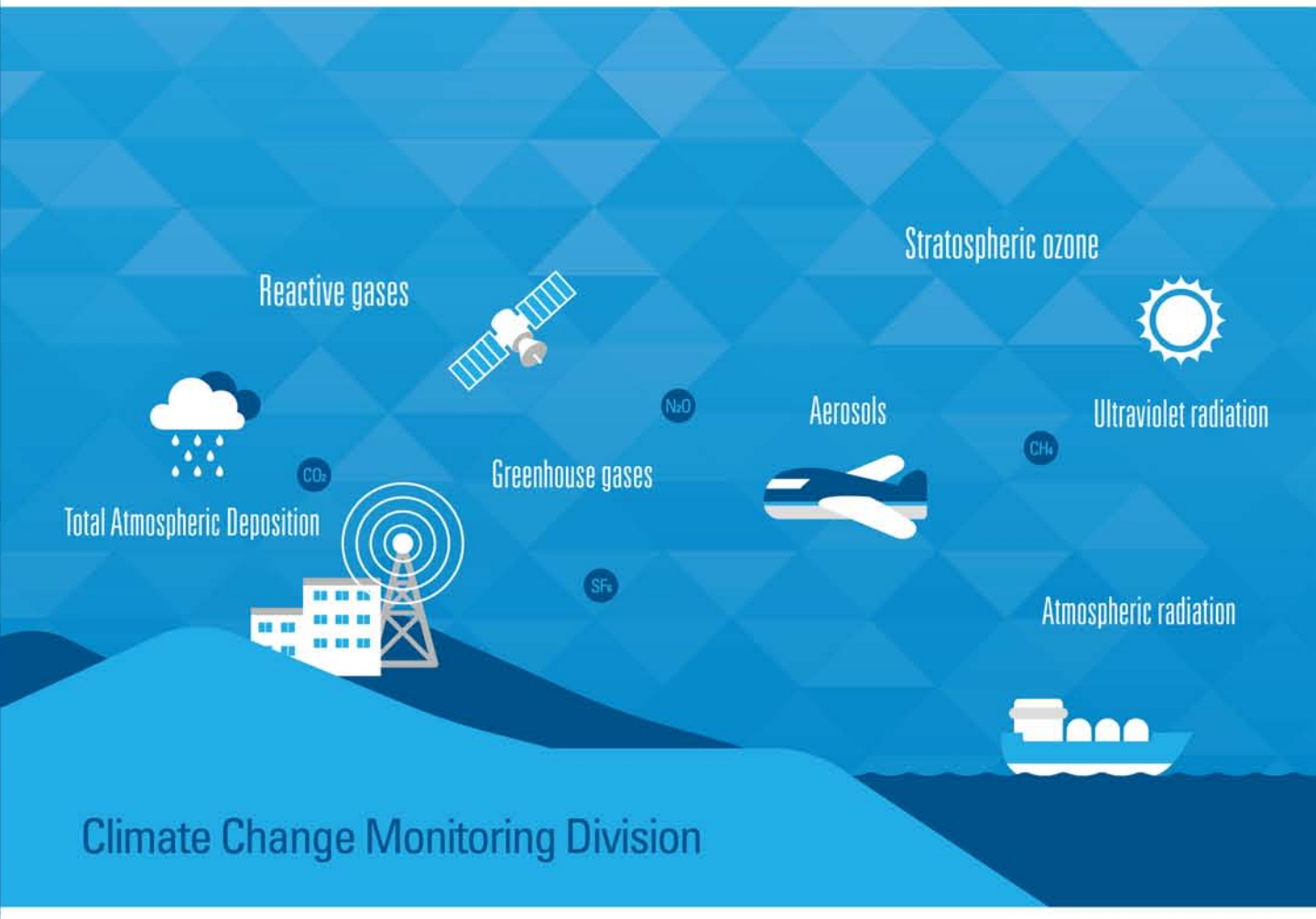
발 간 등 록 번 호
11-1360000-001256-01

2016
02

기후변화감시 기술노트 2016-02

인체 비타민D 생성을 위한 태양자외선복사 노출시간 산정 및 서비스

Estimation of exposure time required for vitamin-D production in human skin



기상청 Korea Meteorological Administration

기후변화감시 기술노트 2016-02

인체 비타민D 생성을 위한 태양자외선복사 노출시간 산정 및 서비스

Estimation of exposure time required for vitamin-D production in human skin

기상청 Korea Meteorological Administration

발행일 : 2016년 1월

발행인 : 과장 김세원
기상연구관 이철규
기상서기 이매향

발행처 : 기후변화감시과
서울특별시 동작구 여의대방로16길 61
전화 02-2181-0647

목 차

1. 서론 ····· 1

2. 용어 정의 ····· 2

3. 국외 선행 연구 사례 ····· 4

4. 인체 비타민D 생성 적정 자외선 노출시간 산출 방법 ····· 9

4.1 홍반 발생 노출시간 산출 ····· 9

4.2 비타민D 생성 노출시간 산출 ····· 10

4.3 비타민D 생성 적정 자외선 노출시간 적용 ····· 12

5. 산출 프로그램 개발 ····· 14

5.1 비타민D 생성 적정 노출시간의 동시 서비스 체계 개발 ····· 14

5.2 비타민D 생성 노출시간 자료 표출 ····· 15

부록

비타민D 생성 적정 노출시간 산출 코드 ····· 18

참고문헌 ····· 36

기후변화감시과 기술노트 발간 목록 ····· 37

1. 서론

태양 자외선복사는 파장에 따라서 315～400 nm 영역을 UV-A, 280～315 nm 영역을 UV-B, 100～280 nm 영역을 UV-C로 분류된다. 태양빛이 대기를 통과할 때 UV-C는 성층권 오존층에서 100% 흡수되어 지표면에 도달하지 않으며, UV-B는 오존층에서 약 95% 흡수되고 나머지 5%만이 지표면에 도달한다. 그러나 UV-A는 성층권오존 및 대기의 영향을 상대적으로 적게 받아 약 95%가 지표면에 도달한다.

지표면에 가장 많이 도달하는 UV-A는 피부 노화 및 주름에 영향을 주고, UV-B는 피부 홍반, 피부암, 백내장 등을 일으킨다. 이에 기상청은 피부 건강 관리와 야외 활동에 유용한 정보로써 총자외선지수를 2015년 3월 4일부터 기상청 홈페이지를 통해 제공해왔다.

그러나 UV-B는 인체에 흡수되면 비타민D를 생성하는 등 긍정적인 면도 가지고 있다. 비타민D는 인체가 햇볕에 노출되면 자연적으로 생성되거나 우유, 생선 등 특정 음식을 먹어야 흡수되며, 뼈 성장과 건강에 필요한 칼슘과 인의 흡수를 돕는 역할을 한다. 인체에 비타민D가 부족할 경우 뼈 성장, 빈혈 뿐 아니라 심장질환, 고혈압 등의 질환을 유발할 수도 있다.

해외에서는 이미 자외선의 긍정적인 영향에 대한 서비스를 시행 중이거나 연구 중인 사례가 보고되고 있다. 유럽에서는 위성과 복사전달모델을 활용한 '비타민D 생성 자외선 복사량'을 TEMIS(Tropospheric Emission Monitoring Internet Service)를 통해 표출 중이다. 뉴질랜드에서는 자외선에 대한 피부화상 가중함수와 비타민D 생성 가중함수를 적용해 백인 피부 유형에 맞는 피부 화상 발생 및 비타민D 생성에 관한 자외선 노출시간을 제공하고 있다. 또한 북미에서는 캐나다 45개 관측지점, 미국 52개 관측지점의 자외선량과 전천일사량, 기상인자(오존전량, 이슬점 온도 등) 관측자료를 활용하여 정오의 비타민D 생성과 관련한 자외선 복사량을 산출하는 연구가 진행되고 있다.

세계보건기구(WHO)에서 제시하고 있는 자외선 노출과 질병 발생 사이의 관계를 살펴보면, 자외선의 과다 노출은 일광화상, 피부암, 백내장 발생 위험을 높이고, 자외선 노출이 부족하면 비타민D 결핍, 근골격계 약화와 같은 질환을 유발하게 된다고 설명하면서 질병 위험성을 최소화 시킬 수 있는 적정 자외선 노출량, 적정 자외선 노출 시간에 대한 정보가 반드시 필요하다고 제시하였다. 따라서 자외선의 부정적인 영향을 반영해 질병 예방 차원에서 제공하는 총자외선지수 뿐 아니라, 긍정적인 영향을 포함한 자외선 적정 노출시간 서비스 제공의 필요성이 제기되었다.

본 기술노트에서는 UV-B(280～315 nm)의 의학적·생물학적 영향을 적용해 동양인의 피부 유형, 피부 노출 습관에 적합한 비타민D 권장량 생성을 위한 적정 자외선 노출시간 산출 방법을 기술하였다.

2. 용어 정의

- **자외선복사** (Ultraviolet radiation, UV radiation)

  가시광선보다 짧은 파장으로 눈에 보이지 않는 빛이다. 광화학·살균·형광·전리 등의 작용을 하며 인체에 과도하게 노출될 경우 피부암을 유발하기도 한다. 태양복사에너지 중 약 7%를 차지하며 대부분은 상층에서 오존에 의해 흡수된다. 자외선복사는 파장에 따라 자외선A (UV-A), 자외선B(UV-B), 자외선C(UV-C)로 구분된다.

- **UV-A**

  자외선 영역 중 315~400 nm 파장 영역에 해당된다. 대기와 성층권 오존의 영향을 적게 받아 95%가 지표에 도달한다. UV-A의 피부에 대한 영향은 UV-B의 약 30% 수준이며, 피부의 표피·상피보다 더 깊게 침투하여 피부노화와 주름 등에 영향을 미친다. 연중 5~6월에 최대값을 가진다. 기후변화감시망에서 운영 중인 UV-A 관측 장비의 관측 영역은 320~400 nm 이다.

- **UV-B**

  자외선 영역 중 280~315 nm 파장 영역에 해당된다. 성층권 오존층을 통과하면서 대부분 흡수되어 5% 만이 지표에 도달한다. 인체가 과하게 노출될 경우 피부의 표피·상피까지 침투하여 색소침착, 홍반, 백내장, 피부암 등을 유발하나, 적당히 노출될 경우 뼈 성장과 건강에 필수적인 비타민D를 생성시킨다. 7~8월에 최대값을 가진다. 기후변화감시망에서 운영 중인 UV-A 관측 장비의 관측 영역은 280~320 nm 이다.

- **UV-C**

  자외선 영역 중 100~280 nm 파장 영역에 해당된다. 염색체 변이를 일으키고 눈의 각막을 해치는 등 해로운 영향을 미치나, 성층권 오존층에서 100% 흡수되어 지표면에 도달하지 않는다.

- **홍반자외선복사** (Erythemal UV radiation)

  자외선B 영역 중 인체에 홍반을 발생시키는 자외선으로 여름에 높고 겨울에 낮게 나타난다.

- **자외선지수** (UV Index)

  태양고도가 최대인 남중시각에 지표에 도달하는 자외선 B(UV-B) 영역의 복사량을 지수식으로 환산한 것으로, 태양에 대한 과다 노출로 예상되는 위험에 대한 정보를 제공한다.

- **홍반자외선지수** (Erythemal UV Index)

  자외선B 영역 중 인체에 홍반(피부화상)을 발생시키는 홍반자외선복사를 지수화한 것으로, 파장별로 인체 피부 홍반에 예민한 홍반 작용스펙트럼을 파장별 자외선 복사에 가중시켜 적분해 산출한다.

- **총자외선지수** (Total Ultraviolet Index)

  자외선B 영역과 자외선A 영역의 복사량에 파장별 가중함수를 적용하고 지수식으로 환산하여 합한 값이다.

3. 국외 선행 연구 사례

노출시간 산출에 앞서 자외선복사 적정 노출시간에 대한 서비스 또는 연구가 진행된 국외 사례를 조사하였다.

유럽에서는 TEMIS(Tropospheric Emission Monitoring Internet Service)를 통해 비타민D 자외선복사량과 홍반자외선복사량 정보를 제공하고 있다. 여기서 제공하는 복사량 정보는 유럽 지역에 국한되어 있으며, 다만 복사전달모델을 활용한 맑은 날씨에서의 자외선 정보는 전지구 대상으로 서비스되고 있다.

뉴질랜드 NIWA(National Institute of Water & Atmospheric Research)에서는 홍반자외선의 부정적인 영향과 비타민D 생성 자외선의 긍정적인 영향을 모두 고려한 자외선 노출시간 정보에 관한 연구가 진행되었다.

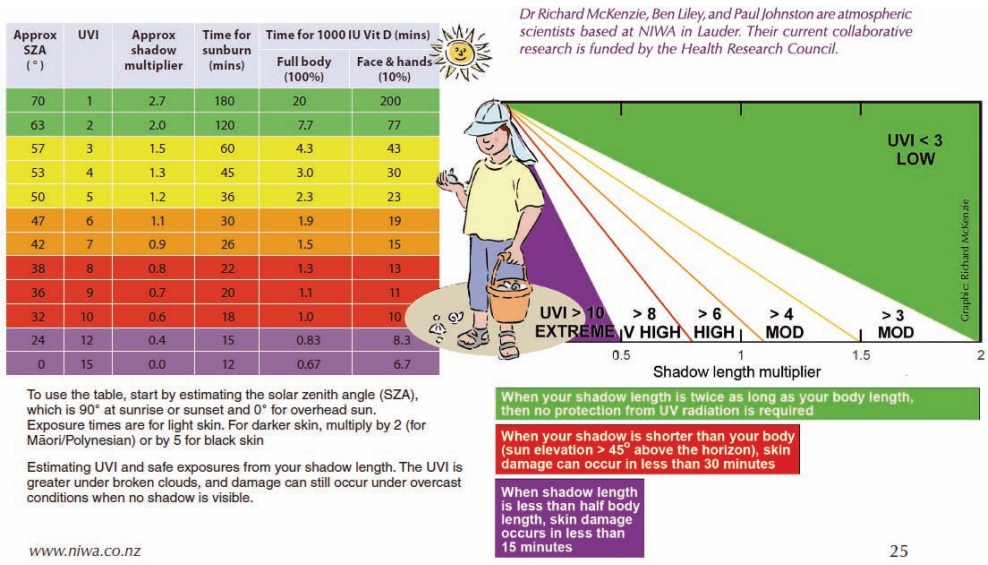


그림 1. NIWA에서 수행한 연구에서 제시된 자외선지수 별 적정 노출시간 정보 서비스 예 (McKenzie et al., 2009).

Kazantzidis et al. (2009)에서는 유럽 3개 지역의 파장별 자외선복사량 관측값을 이용하여 비타민D 생성 자외선복사량을 산출하였다. 이 논문에서는 비타민D 생성 자외선복사량 사이의 비율을 분석하였고, 관측소의 지방시(Local Time)에 대해 평균된 비타민D 생성 자외선복사량이 제시되었다. 또한, 비타민D 경구 투여량 1000 IU에 해당하는 인체 노출 면적의 1/4을 기준으로 한 비타민D 생성 자외선복사량인 1 SDD(Standard Vitamin D Dose)를 피부 유형에 따라 구분하고, 그 값에 해당하는 복사량을 얻기 위해 필요한 노출 시간의 평균을 제시하였다. 표1은 피부 유형에 따른 1 SDD를 나타낸다.

표 1. Kazantzidis et al.(2009)에서 제시한 피부 유형에 따른 1 SSD 값

Skin type	Color	1 SDD/kJ·m^{-2}
I	Caucasian; blonde or red hair, freckles, fair skin, blue eyes	0.0372
II	Caucasian; blonde or red hair, freckles, fair skin, blue eyes or green eyes	0.0465
III	Darker Caucasian, light Asian	0.0558
IV	Mediterranean, Asian, Hispanic	0.0836
V	Middle Eastern, Latin, light-skinned black, Indian	0.1114
VI	Dark-skinned black	0.1851

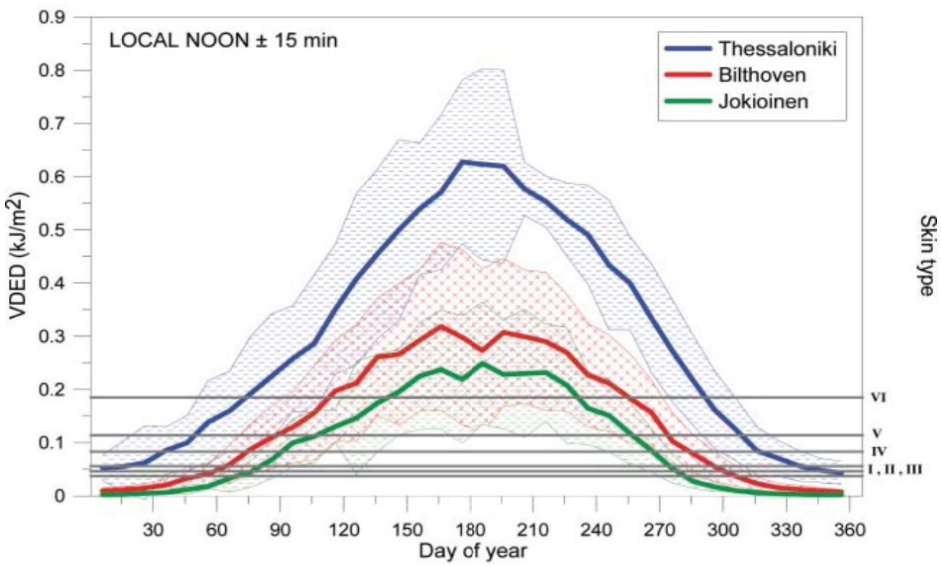


그림 2. 피부 유형별 비타민D 생성 자외선복사량과 관측소별 정오 기준 ±15분 동안에 비타민D를 생성하기 위한 자외선복사량(Kazantzidis et al., 2009).

그림 2는 피부 유형별 비타민D 생성 노출시간(회색 실선)과 관측소별 노출시간을 나타내었다. 어두운 피부를 가진 사람의 경우 겨울철에 고위도인 핀란드 Jokioinen 지역에서 정오 기준 ±15분, 즉 30분 정도 노출된다 하더라도 충분한 비타민D를 생성하지 못한다. 이를 통해 비타민D 생성 자외선 노출시간을 계산할 때 피부 유형 뿐 아니라 위도 또한 고려해야 함을 알 수 있다.

그림 3은 McKenzie et al. (2009)에서 산출한 값으로 (a)는 뉴질랜드 Lauder에서 관측한 파장별 자외선복사량과 CIE에서 제공하는 홍반가중함수, 비타민D 가중함수를 나타내며, (b)는 여름철과 겨울철에 관측된 파장별 자외선복사량에 각각의 가중함수를 적용하여 홍반자외선과 비타민D 생성 자외선을 산출한 것이다. 이를 바탕으로 홍반자외선복사량(UV_{Ery})와 비타민D 생성 자외선복사량(UV_{VitD})의 비율($R = UV_{VitD}/UV_{Ery}$)을 계산해 약 2.0의 비율을 가짐을 확인하였다.

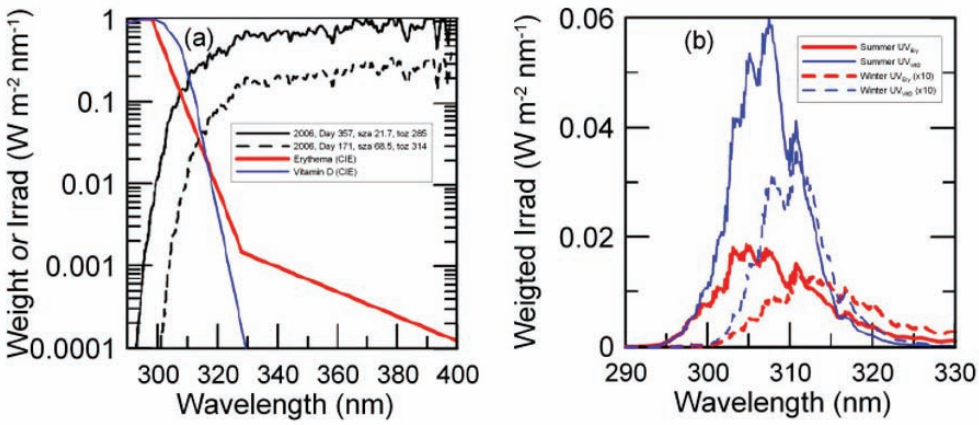


그림 3. (a)는 뉴질랜드의 Lauder 지역에서 측정한 파장별 자외선복사량 관측 자료와 홍반자외선 가중함수, 비타민D 생성 자외선 가중함수를 나타내며, (b)는 파장별 자외선복사량에 각각의 가중함수를 적용하여 홍반자외선과 비타민D 생성 자외선을 산출한 것을 나타냄 (McKenzie et al., 2009).

McKenzie et al. (2009)을 통해 홍반자외선 노출시간(T_E)와 비타민D 생성 자외선 노출시간(T_D)을 산출할 수 있다.

$$T_E\,(\min) = \frac{4000}{60}\frac{MEDF \cdot SPF}{UVI} \tag{1}$$

$$T_D = T_{D0}\frac{UVI_0}{UVI}\frac{R_0}{R}\frac{A_0}{A}\frac{MEDF}{MEDF_0}\frac{SPF}{SPF_0} \tag{2}$$

식(1)과 식(2)로 계산된 값을 자외선지수에 따른 값으로 적용하면 일광화상에 대한 피해를 예방하면서 자연적으로 비타민D를 충분히 생성할 수 있는 임계시간을 산출할 수 있다.

표2는 식(1)과 식(2)를 피부 유형Ⅱ(백인 피부)에 대해 적용하고, UVI 값별 홍반피해 노출시간 및 노출 면적에 따른 비타민D 생성 시간을 나타낸 것이며, 그림 4는 피부유형Ⅱ에 대한 홍반자외선 노출 시간과 노출 면적에 따른 비타민D 자외선 노출시간을 나타낸다.

표 2. 식(1)과 식(2)를 피부 유형Ⅱ(백인 피부)에 대하여 적용하고, UVI 값별 홍반피해 노출 시간 및 노출 면적에 따른 비타민D 생성 시간

UVI	Approx. [1]SZA	Shadow multiplier (tan[SZA])	Multiplier (ratio [2]VitD/[3]Ery)	Time for erythema (min)	Time for 1000 IU VitD (min)	
					Full body (100%)	Face and hands (10%)
1	70	2.7	1.0	180	20	200
2	63	2.0	1.3	120	7.7	77
3	57	1.5	1.55	60	4.3	43
4	53	1.3	1.65	45	3.0	30
5	50	1.2	1.75	36	2.3	23
6	47	1.1	1.8	30	1.9	19
7	42	0.9	1.85	26	1.5	15
8	38	0.8	1.9	22	1.3	13
9	36	0.7	1.95	20	1.1	11
10	32	0.6	2.0	18	1.0	10
11	28	0.5	2.0	16	0.9	9
12	24	0.4	2.0	15	0.83	8.3
13	15	0.3	2.0	14	0.8	8
14	10	0.2	2.0	13	0.71	7.1
15	0	0.0	2.0	12	0.67	6.7

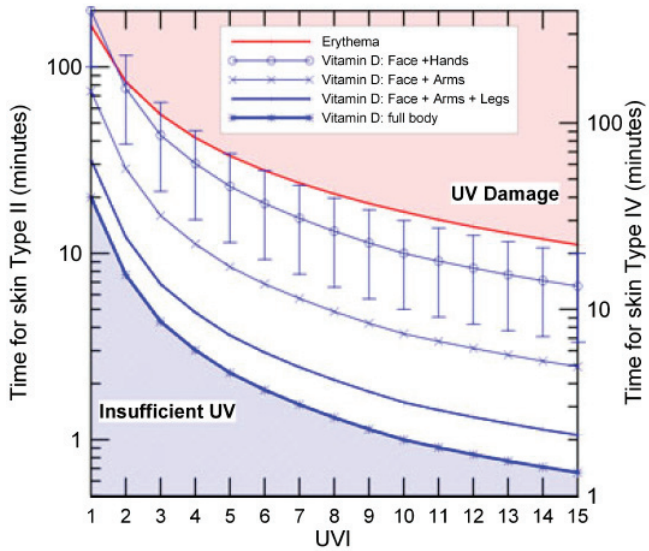


그림 4. 피부 유형Ⅱ에 대한 노출 면적과 UVI에 따른 홍반자외선 노출 시간과 비타민D 자외선 노출 시간(McKenzie et al., 2009).

1) SZA : 태양천정각(Solar Zenith Angle)
2) VitD : 비타민D(Vitamin D)
3) Ery : 홍반(Erythema)

이상의 국외 선행 연구 사례로 볼 때, 비타민D 생성 자외선 노출 시간 산출과 관련된 많은 연구들이 백인 피부 유형에 맞춰 개발되어 아직까지 동양인의 피부 유형에 적합한 비타민D 생성 노출 시간에 대한 연구는 거의 이루어지지 않았음을 알 수 있다.

4. 인체 비타민D 생성 적정 자외선 노출시간 산출 방법

홍반과 비타민D 생성에 관여하는 자외선 파장대는 자외선B 영역인 280~320 nm로, 홍반 발생을 예방하면서 비타민D를 충분히 생성할 수 있는 노출 시간 정보가 필요하다. 따라서 국제조명위원회(International Commission on Illumination)에서 제시한 비타민D 생성 가중함수를 적용하여 비타민D 생성 자외선을 계산하였다.

4.1 홍반 발생 노출시간 산출

홍반이 나타나기 시작하는 시간(홍반 발생 노출시간; T_E)과 충분한 비타민D 생성에 필요한 노출 시간(비타민D 생성 노출시간; T_D)은 피부 유형에 따라 달라진다. 따라서 피부가 홍반 피해를 입는 최소한의 홍반자외선 복사량인 최소홍반량조사(MED, Minimal Erythemal Dose) 개념을 도입했다. MED는 피부 유형에 따라 홍반 피해를 입게 되는 임계값으로 홍반자외선 복자에 일정시간(T_E) 노출될 때 MED 값을 초과하면 홍반이 발생하게 된다. MED는 표준홍반량조사(SED, Standard Erythemal Dose[=100 Jm^{-2}]) 값에 피부 유형에 따른 상수를 곱한 값이다.

$$MED = MEDF \times SED \tag{3}$$

MEDF는 6가지 피부 유형에 대한 최소홍반량인자(Minimal Erythemal Dose Factor)이다. 표 3에서 동양인과 가장 가까운 지중해형(Skin Type Ⅳ, Mediterranean) 기준일 경우 상수 4.5~6으로 최소 MED = 4.5 × SED에 도달할 때 피부에 홍반이 발생한다.

표 3. 피부유형에 따른 MEDF 값(McKenzie et al., 2009).

Skin Type	Description	SED to burn	
		Fitzpatric (1998)	Aust/NZS (2002)
Ⅰ	Celtic (always burns)	2–3	⟨2.5
Ⅱ	Pale (burns easily)	2.5–3	2.5
Ⅲ	Caucasian (may burn)	3–5	3.5
Ⅳ	Mediterranean (burns rarely)	4.5–6	4.5
Ⅴ	S. American (rarely burns)	6–20	–
Ⅵ	Negroid (rarely burns)	6–20	–

피부 유형에 따라 홍반 발생에 필요한 최소 복사량은 관측된 홍반 복사량(UV_{Ery}, 단위 $W{\cdot}m^{-2}$) × 노출시간(T_E)이다.

$$UV_{Ery}(Wm^{-2}) \times T_E = MEDF \times SED(100Jm^{-2}) \tag{4}$$

$$UV_{Ery}(Js^{-1}m^{-2}) \times T_E = MEDF \times 100(Jm^{-2}) \tag{5}$$

$$T_E = \frac{MEDF \times 100(Jm^{-2})}{UV_{Ery}(Js^{-1}m^{-2})} \tag{6}$$

$$T_E(\mathrm{sec}) = \frac{MEDF \times 100}{UV_{Ery}} \tag{7}$$

홍반자외선량(UV_{Ery})에 40을 곱하여 자외선지수(UVI)로 바꾸어 계산하면 다음과 같이 홍반 발생 노출시간(T_E)이 산출된다.

$$T_E(\mathrm{sec}) = \frac{MEDF \times 100}{UV_{Ery}} = \frac{MEDF \times 100}{\frac{1}{40} \times UVI} \tag{8}$$

$$T_E(\mathrm{sec}) = \frac{40 \times 100 \times MEDF}{UVI} \tag{9}$$

$$T_E(\mathrm{min}) = \frac{4000 \times MEDF}{60 \times UVI} \tag{10}$$

홍반자외선과 비타민D 생성 자외선의 비율(Ratio)과 UVI(B)에 대한 관계식은 다음과 같다.

$$Ratio = 1.5293 + 0.6735 \times UVI(B)^{0.0887} \tag{11}$$

4.2 비타민D 생성 노출시간 산출

자외선B 영역에 노출되었을 때, 비타민D를 생성할 수 있는 복사량인 비타민D 생성 자외선 복사량(UV_{VitD})은 UV_{Ery}와 마찬가지로 파장별 가중함수를 자외선B 복사량에 적용해 산출한다. 또한, 비타민D 생성 노출시간도 홍반 발생 노출시간과 동일하게 일정시간(T_D) 자외선에 노출되었을 때 인체에 필요한 최소한의 비타민D를 생성할 수 있음을 적용한다.

$$UV_{VitD}(Wm^{-2}) \times T_D = Sufficient\ Vitamin\ D\ Radiation \quad (12)$$

먼저, 피부 노출 면적에 대한 고려 및 UV_{VitD} 지수 개발을 위해 UV_{ERY}와 UV_{VitD}의 비율을 구한다.

$$R = \frac{UV_{VitD}}{UV_{Ery}} = \frac{UV_{VitD}}{UVI} \quad (13)$$

식(12)의 UV_{VitD}에 식(13)에서 산출된 UV_{VitD} 값(R×UVI)을 대입하면 다음과 같이 비타민D 생성을 위한 자외선B 복사량을 계산할 수 있다.

$$UVI \times R \times T_D = Sufficient\ Vitamin\ D\ Radiation \quad (14)$$

Sufficient Vitamin D Radiation은 피부 유형(MEDF)과 노출 면적(A)에 따라 달라지기 때문에 T_D는 R, UVI, A에 반비례하고, MEDF에 비례한다.

$$T_D = \frac{10 \times 2 \times 1 \times MEDF}{UVI \times R \times A \times 2.5} \quad (15)$$

식 (10)과 식(15)에서 구한 T_D와 T_E의 관계를 비율(R)에 대한 식으로 구하면 다음과 같다.

$$\frac{T_D}{T_E} = \frac{20 \times 60}{2.5 \times 4000 \times A} \times \frac{1}{R} = \frac{0.12}{A} \times \frac{1}{R} \quad (15)$$

요약하면, 자외선B의 관측값으로부터 홍반자외선지수를 산출하고, 산출된 홍반자외선량과 비타민D 생성 자외선의 비를 홍반자외선지수에 적용해 비타민D 생성 노출시간을 구할 수 있다.

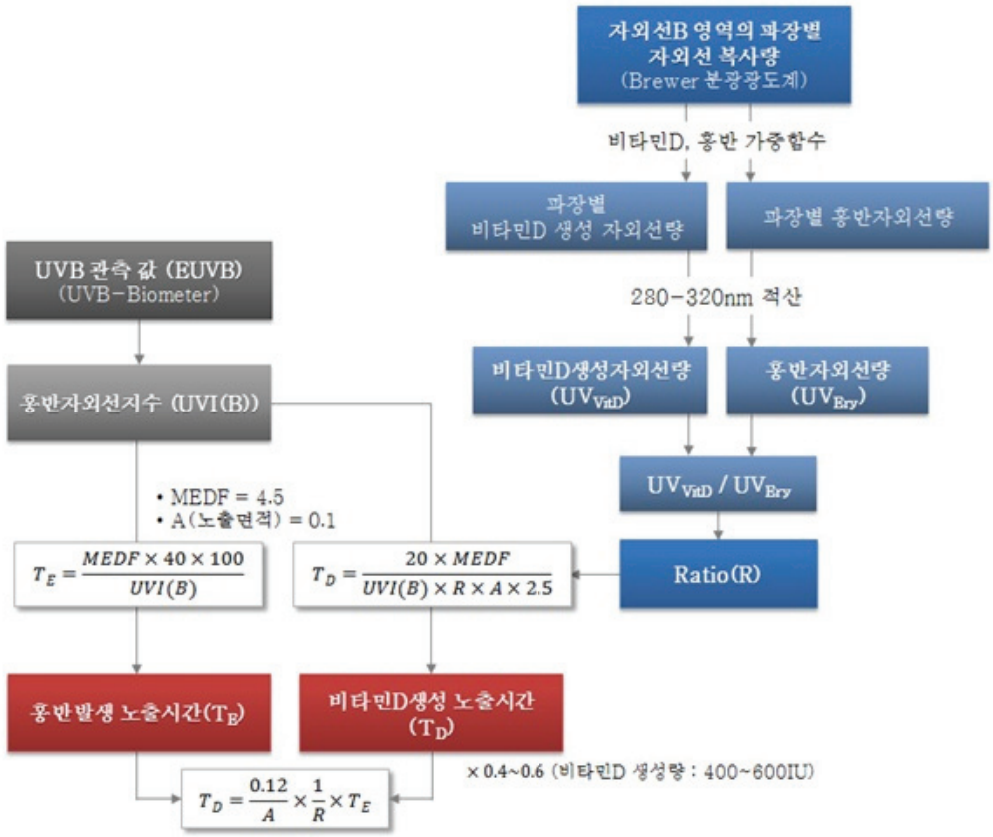


그림 5. 홍반자외선지수와 비타민D 생성 자외선지수 산출값을 이용한 홍반 발생 노출시간과 비타민D 생성 노출시간 계산 알고리즘

4.3 비타민D 생성 적정 자외선 노출시간 적용

인체에서 비타민D 일일 권장량 생성을 위한 적정 자외선 노출 시간을 산출하기 위해 MEDF는 동양인 피부유형에 평균적으로 산정된 4.5를 대입하였다. 또한, 햇볕에 노출이 부족한 겨울철(11월~익년 3월)에 서비스가 제공되는 것을 감안하여 노출면적 A는 겨울철에 얼굴과 손이 자외선에 노출되는 면적인 0.1을 적용하였다(여름철에 수영복을 입었을 때 노출 면적을 1로 가정).

관측된 자외선지수가 6.0이라 할 때, 식(11)를 사용하여 비율(R) 값을 구한다.

$$Ratio = 1.5293 + 0.6735 \times 6.0^{0.0887} \approx 2.3 \quad (16)$$

홍반발생 노출시간(T_E)은 식(10)에 UVI 값을 대입해 계산할 수 있다.

$$T_E(\min) = \frac{4000}{60} \times \frac{4.5}{6.0} = 50(\min) \quad (17)$$

비타민D 생성을 위한 노출시간(T_D)은 식(14) 또는 식(15)로부터 산출 가능하다. 식(15)에는 식(17)의 T_E 값을 대입한다.

$$T_D = \frac{10 \times 2 \times 1 \times 4.5}{2.5 \times 6.0 \times 2.3 \times 20.1} = 26.09 \approx 26(\min) \quad (18)$$

$$T_D = \frac{0.12}{0.1} \times \frac{1}{2.3} \times T_E = 26.09 \approx 26(\min) \quad (19)$$

위 식에서 산출된 비타민D 생성 적정 노출시간은 일일 비타민D 필요량이 1000 IU일 때를 기준으로 하였으며, 한국인에 요구되는 비타민D 필요량 400~600 IU(2015 한국인 영양소 섭취기준, 보건복지부)을 적용할 경우 계산된 식에 0.4~0.6을 곱하면 된다. 이번 서비스에서는 최소량인 0.4를 곱하여 계산하였다. 따라서 겨울철 자외선지수가 6.0일 때 비타민D 생성을 위한 최소 노출시간(T_D)은 식(20)또는 식(21)에 0.4를 곱한 10분으로 햇볕에 적어도 10분 이상 노출되어야 인체에 충분한 비타민D를 생성할 수 있게 된다.

5. 산출 프로그램 개발

5.1 비타민D 생성 적정 노출시간의 동시 서비스 체계 개발

앞에서 산출한 알고리즘을 바탕으로 기존에 서비스 되고 있는 총자외선지수와 비타민D 생성 적정 노출시간의 동시 서비스 체계를 개발하였다. 서비스 체계의 흐름도는 아래와 같다.

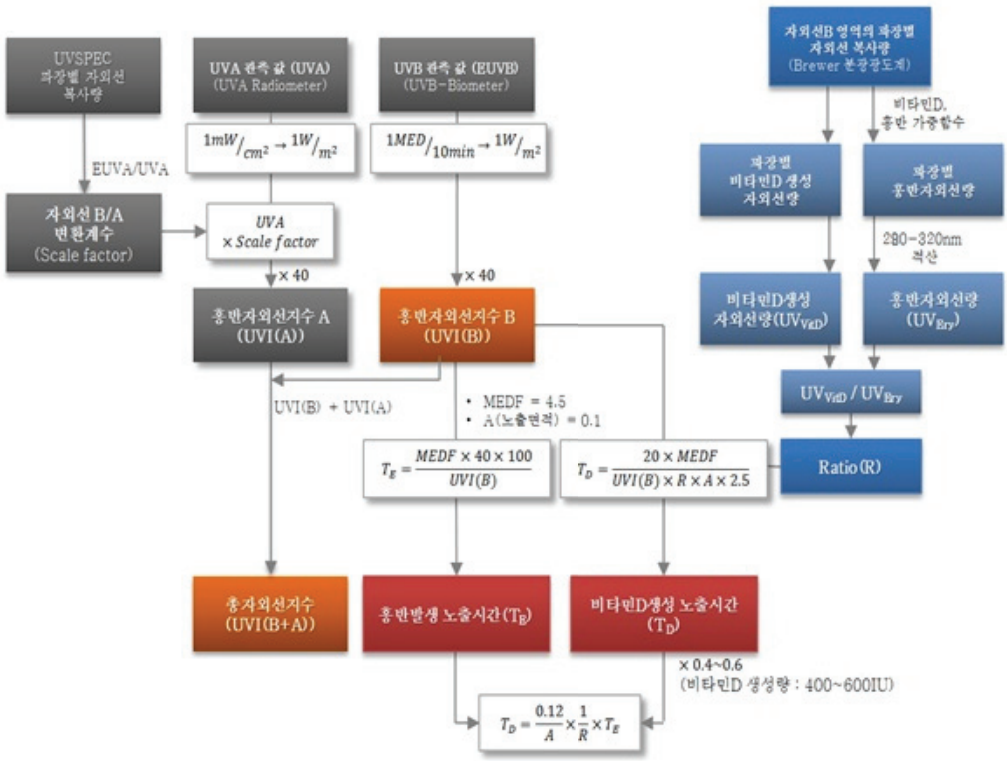


그림 6. 총자외선지수, 비타민D 생성 자외선지수의 서비스 체계

총자외선지수(UVI(B+A))와 홍반자외선지수(UVI(B))는 UV-Biometer의 관측값으로부터 산출할 수 있다. 자외선B의 홍반자외선량과 자외선A의 홍반자외선량 사이의 변환계수(Scale factor)는 복사전달모델(UVSPEC)의 파장별 자외선A 복사량을 이용해 계산되며, 홍반자외선지수(UVI(B))로부터 홍반발생 노출시간(T_E)을 산출할 수 있다.

비타민D 생성 적정 노출시간을 계산하기 위하여 Brewer 분광광도계의 파장별 자외선복사량 관측값에 홍반과 비타민D에 대하여 각각 가중한 두 자외선량의 비율(R)이 사용되며, 이 때 R은 비타민D 생성 자외선복사량(UV_{VitD})에 대한 홍반자외선량(UV_{Ery})의 비율로 비타민D 자외선지수의 정보가 포함되어 있다. 이를 통해 홍반발생 노출시간(T_E)와 비타민D 생성 적정 노출

시간(T_D)가 산출되며, 두 식의 공통변수인 MEDF와 UVI를 이용해 두 노출시간의 관계식을 수립할 수 있다.

5.2 비타민D 생성 노출시간 자료 표출

개발된 비타민D 생성 적정 노출시간을 기상청 홈페이지(www.kma.go.kr) 및 기후정보포털(www.climate.go.kr)에 표출하기 위해서 10분 간격으로 자외선 노출 시간이 산출되도록 서버의 백엔드 IDL 프로그램을 제작해 구동하도록 설정하였다.

IDL 프로그램은 텍스트 파일 형태로 만들어지고 있는 7개 자외선 관측지점(서울, 안면도, 강릉, 울릉도, 포항, 목포, 제주 고산)의 홍반자외선지수를 읽어 들여 비타민D 생성 노출시간을 계산하고 있다. 계산된 노출시간은 서버 메모리에 임시로 저장되어, 웹페이지에 표출된다.

비타민D 생성 노출시간의 표출은 기존 총자외선지수 그래프에 현재 시각 기준 적정 노출시간을 녹색 박스에 함께 표출하도록 하였다. 또한, 11월부터 익년 3월까지 5개월간 연중 제공중인 총자외선지수와 함께 서비스되며, 09~14시에 10분 간격으로 표출하여 동절기 낮 시간대 야외활동에 활용할 수 있도록 하였다.

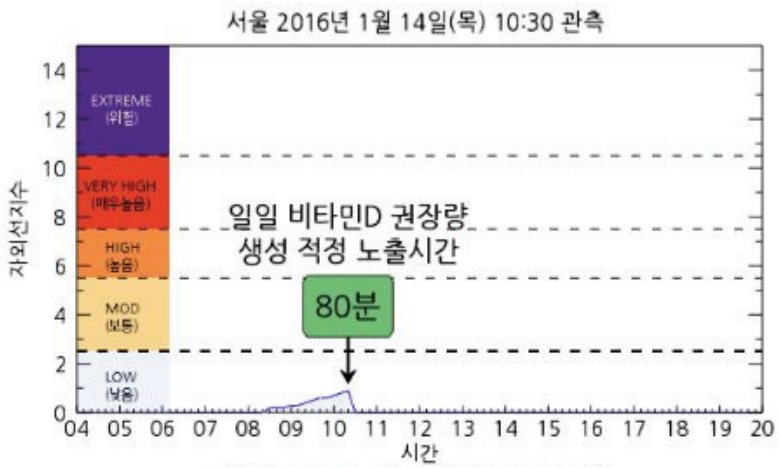


그림 7. 비타민D 생성 적정 노출시간 표출

위 그림은 관측소 일곱 개 지점에 대하여 다음과 같은 위치로 저장된다.

· 경로: /n0005/LOCAL_INTERGRATION/webapps/ccic/ROOT/data/graph/

· 파일명 규칙: (관측소 코드)_(일자 코드).png

비타민D 생성 적정 노출시간 정보에 대한 국민들의 이해를 돕기 위해 참고 문구를 아래와 같이 추가 하였다.

· 일일 비타민D 권장량 생성 적정 노출시간은 하루 중 현재 시각 이전에 비타민D 섭취 혹은 자외선에 노출이 없고 얼굴과 손만 노출되는 것을 가정하여 계산하였습니다. 하루 중 09~14시에 제공됩니다.

· 일일 비타민D 권장량은 보건복지부(2015년)에서 제시한 400 IU를 기준으로 하였으며, 인체 피부 유형은 동양인과 가까운 지중해형(Mediterranean)을 적용 하였습니다 (Kazantzidis et al., 2009).

총자외선지수와 비타민D 생성 적정 노출시간 정보는 그림 4와 같이 함께 제공되며, 그 제공 위치는 다음과 같다.

· 기상청 홈페이지(www.kma.go.kr) 〉 날씨 〉 생활과 산업 〉 생활기상정보 〉 자외선 관측자료

· 기후변화정보센터(www.climate.go.kr) 〉 기후변화감시 〉 실시간 관측자료 〉 총자외선지수

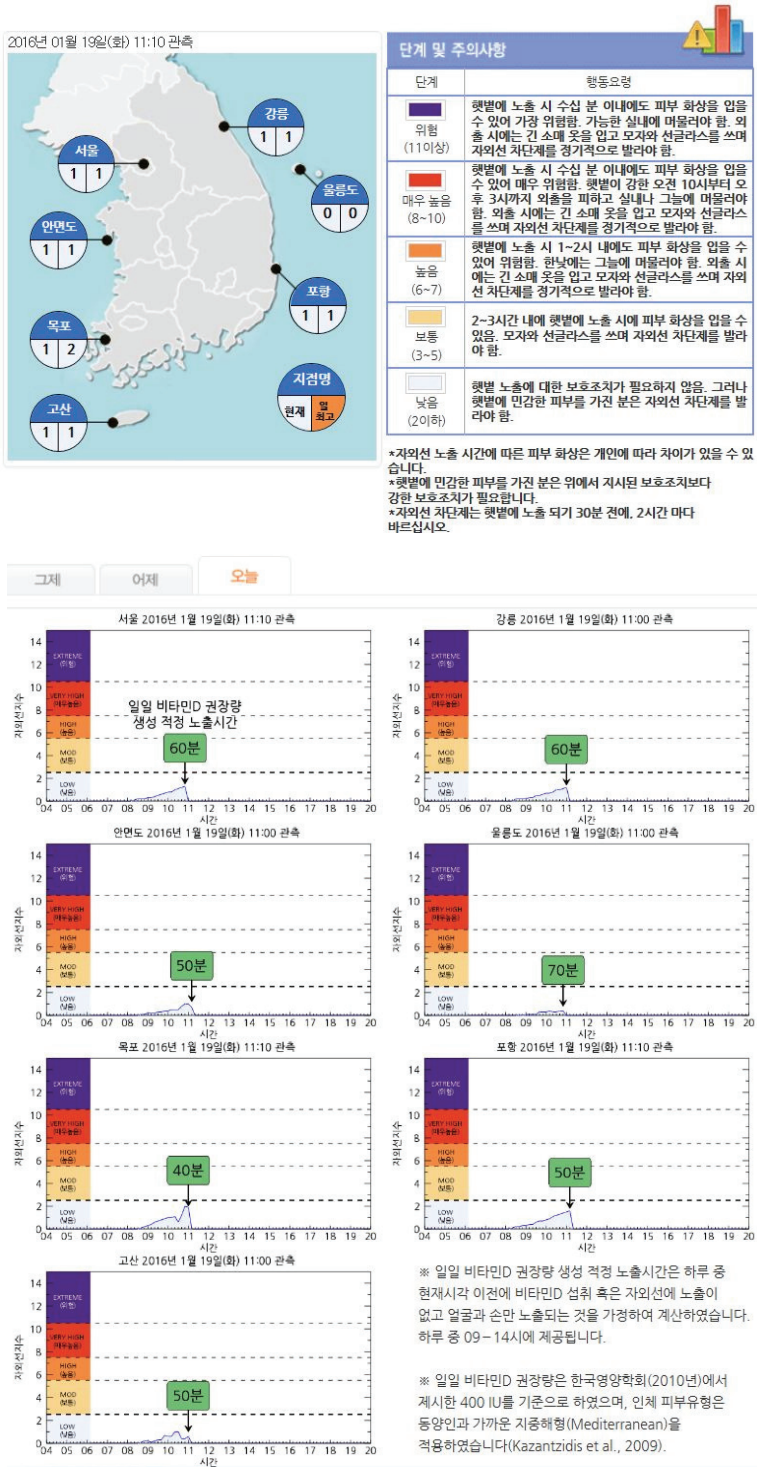


그림 8. 총자외선지수와 비타민D 생성 자외선지수 표출 화면

〈 비타민D 생성 적정 노출시간 산출 코드 〉

```
PRO UVIndex_vitd

  ; Pick a UV index data file
  root_dir_b = '/n0005/LOCAL_INTEGRATION/webapps/ccic/ROOT/data/uvb'; 'C:₩Users
₩WaveGuide₩Documents₩workspace₩2014₩KMA₩CCIC₩UVI₩uvb'; '/Users/WaveGuide/
Documents/workspace/2014/UVI/uvb';
  root_dir_a = '/n0005/LOCAL_INTEGRATION/webapps/ccic/ROOT/data/uva'; 'C:₩Users
₩WaveGuide₩Documents₩workspace₩2014₩KMA₩CCIC₩UVI₩uva'; '/Users/WaveGuide/
Documents/workspace/2014/UVI/uva';
  site_name=['013', '105', '115', '132', '138', '165', '108']; 고산, 강릉, 울릉도, 안면도, 포항,
목포, 서울

  ; declaration of varables
  flux_b=fltarr(7,97)
  flux_a=fltarr(7,97)

  xbar=dblarr(97)

  ypoly1 = replicate(2.5, 13)
  ypoly2 = replicate(5.5, 13)
  ypoly3 = replicate(7.5, 13)
  ypoly4 = replicate(10.5, 13)
  ypoly5 = replicate(15, 13)

  day_index = ['1','2','3']

  date_tday = systime(/julian)
  date_yday = systime(/julian)-1
  date_tdby = systime(/julian)-2
  date_tdow = systime(0)
```

```
caldat, date_tday, month_tday, day_tday, year_tday, hour_tday, minute_tday
caldat, date_yday, month_yday, day_yday, year_yday
caldat, date_tdby, month_tdby, day_tdby, year_tdby
dow_week_e = strsplit(date_tdow, /extract, ' ')

if dow_week_e(0) eq 'Mon' then begin
  dow_week = '월'
  dow_week_y = '일'
  dow_week_dby = '토'
endif
if dow_week_e(0) eq 'Tue' then begin
  dow_week = '화'
  dow_week_y = '월'
  dow_week_dby = '일'
endif
if dow_week_e(0) eq 'Wed' then begin
  dow_week = '수'
  dow_week_y = '화'
  dow_week_dby = '월'
endif
if dow_week_e(0) eq 'Thu' then begin
  dow_week = '목'
  dow_week_y = '수'
  dow_week_dby = '화'
endif
if dow_week_e(0) eq 'Fri' then begin
  dow_week = '금'
  dow_week_y = '목'
  dow_week_dby = '수'
endif
if dow_week_e(0) eq 'Sat' then begin
```

```
   dow_week = '토'
   dow_week_y = '금'
   dow_week_dby = '목'
endif
if dow_week_e(0) eq 'Sun' then begin
   dow_week = '일'
   dow_week_y = '토'
   dow_week_dby = '금'
endif

; main module starts
for i=0, 6 do begin
   for l=0, 2 do begin

      yg=fltarr(97)
      yy=fltarr(97)
      yo=fltarr(97)
      yr=fltarr(97)
      yp=fltarr(97)

      print, site_name(i)+'_'+day_index(l)

      ;  read uvindex table
      index_file_b = file_search(root_dir_b, site_name(i)+'_'+day_index(l)+'.tbl')
      index_file_a = file_search(root_dir_a, site_name(i)+'_'+day_index(l)+'.tbl')
      index_file_c = file_search(root_dir_b, 'uvbdate.txt')
      readcol, index_file_c, y_obs, m_obs, d_obs, h_obs, mn_obs;, 'l, l, l, l, l'

      ; check table is available
      if index_file_b ne '' and index_file_a ne '' then begin
         readcol, index_file_b, y_b, m1_b, d_b, h_b, m2_b, f_b, /silent
         readcol, index_file_a, y_a, m1_a, d_a, h_a, m2_a, f_a, /silent
```

```
        for j=0, 96 do begin
          flux_b(i, j) = f_b(j)
          flux_a(i, j) = f_a(j)
        endfor
        file_exist_index = 'complete'
        value_existence_index_b = where(flux_b(i,0:96) ne -999.0, count_b)
        value_existence_index_a = where(flux_a(i,0:96) ne -999.0, count_a)
        if count_a eq 0 then file_exist_index = 'positiveB'
        if i eq 0 then begin
          ; The data to plot.
          time = timegen(97, units='minutes', step_size=10, start=julday(m1_b(0),
d_b(0), y_b(0), h_b(0), m2_b(0)) $
            , final=julday(m1_b(96), d_b(96), y_b(96), h_b(96), m2_b(96)))
        endif
        if count_b eq 0 then begin
          file_exist_index = 'positiveA'
          if i eq 0 then begin
            ; The data to plot.
            time = timegen(97, units='minutes', step_size=10, start=julday(m1_a(0),
d_a(0), y_a(0), h_a(0), m2_a(0)) $
              , final=julday(m1_a(96), d_a(96), y_a(96), h_a(96), m2_a(96)))
          endif
        endif
      endif

      if index_file_b ne '' and index_file_a eq '' then begin
        readcol, index_file_b, y_b, m1_b, d_b, h_b, m2_b, f_b
        for j=0, 96 do begin
          flux_b(i, j) = f_b(j)
        endfor
        file_exist_index = 'positiveB'
        if i eq 0 then begin
```

```
        ; The data to plot.
        time = timegen(97, units='minutes', step_size=10, start=julday(m1_b(0),
d_b(0), y_b(0), h_b(0), m2_b(0)) $
          , final=julday(m1_b(96), d_b(96), y_b(96), h_b(96), m2_b(96)))
      endif
    endif

    if index_file_b eq '' and index_file_a ne '' then begin
      readcol, index_file_a, y_a, m1_a, d_a, h_a, m2_a, f_a
      for j=0, 96 do begin
        flux_a(i, j) = f_a(j)
      endfor
      file_exist_index = 'positiveA'
      if i eq 0 then begin
        ; The data to plot.
        time = timegen(97, units='minutes', step_size=10, start=julday(m1_a(0),
d_a(0), y_a(0), h_a(0), m2_a(0)) $
          , final=julday(m1_a(96), d_a(96), y_a(96), h_a(96), m2_a(96)))
      endif
    endif

    if index_file_b eq '' and index_file_a eq '' then begin
      break
    endif

    ; make an array for each level cross line
    xpoly = time(0:12)

    ; make values for x-axis domain
    x = time

    ; make values for y-axis domin by each data status
```

```
    if file_exist_index eq 'positiveB' or file_exist_index eq 'complete' then begin
      y = flux_b(i,*)
      if file_exist_index eq 'complete' then begin
        for j=0, n_elements(y)-1 do begin
          if y(j) lt 0.0 then y(j)=0.0
          if f_a(j) lt 0.0 then f_a(j)=0.0
        endfor
      endif
      if file_exist_index eq 'positiveB' then begin
        for j=0, n_elements(y)-1 do begin
          if y(j) lt 0.0 then y(j)=0.0
        endfor
      endif
    endif
    if file_exist_index eq 'positiveA' then begin
      y = flux_a(i,*)
      for j=0, n_elements(y)-1 do begin
        if y(j) lt 0.0 then y(j)=0.0
      endfor
    endif

    ; Draw the plot axes.
    win = window(dimensions=[600, 350], /buffer)
    win.Erase

    dummy = label_date(date_format=['%H'])
    uvi=plot(x, y, xtickunits='hour', xtickinterval=1, xminor=5, xstyle=1, yminor=1,
xrange=[time(0), time(96)], yrange=[0.0, 15.0], axis_style=2 $
      , /current, xtickformat='label_date', XTitle='시간', YTitle='자외선지수', font_name=
'NanumGothic', thick=2 $
      , xticklen=0.02, yticklen=0.02, /nodata, font_size=13, xtickfont_size=12, ytickfont_
size=12, margin=[0.09,0.12,0.02,0.08], title='안면도')
```

```
uvi['axis2'].ticklen=0

; calculate half value of x-axis bin
xterm = (x(49)-x(48))/2.0d

; calculate values of each uvindex for coloring of each bar
FOR k=0, n_elements(y)-1 DO BEGIN

  xbar(k) = x(k) - xterm
  if f_a(k) eq -999.0 then f_a(k) = 0.0

  if y(k)+f_a(k) ge 0.0 and y(k) lt 2.5 then begin
    yg(k) = y(k)+f_a(k)
  endif
  if y(k)+f_a(k) ge 2.5 and y(k) lt 5.5 then begin
    yy(k) = y(k)+f_a(k)
    ;print, k
  endif
  if y(k)+f_a(k) ge 5.5 and y(k) lt 7.5 then begin
    yo(k) = y(k)+f_a(k)
  endif
  if y(k)+f_a(k) ge 7.5 and y(k) lt 10.5 then begin
    yr(k) = y(k)+f_a(k)
  endif
  if y(k)+f_a(k) ge 10.5 then begin
    yp(k) = y(k)+f_a(k)
  endif

ENDFOR

; draw color bar
feat_green=barplot(xbar, yg, fill_color='EFF3FF'x, /overplot, outline=0, /histogram)
```

```
        feat_yellow=barplot(xbar, yy, fill_color='FED98E'x, /overplot, outline=0, /histogram)
        feat_orange=barplot(xbar, yo, fill_color='FD8D3C'x, /overplot, outline=0, /histogram)
        feat_red=barplot(xbar, yr, fill_color='F03B20'x, /overplot, outline=0, /histogram)
        feat_purple=barplot(xbar, yp, fill_color='54278F'x, /overplot, outline=0, /histogram)

        if file_exist_index eq 'complete' then begin
;         uvab=plot(x, y+f_a, /overplot, thick=1, color='blue', name='총자외선지수(A + B)',
ANTIALIAS=1, yrange=[0.0, 15.0])
         uvab=plot(x, y+f_a, /overplot, thick=1, color='blue', name='자외선지수(A + B)',
ANTIALIAS=1, yrange=[0.0, 15.0])
        endif

        ; draw level info on left-side
        index_5 = barplot(xpoly, ypoly5, /overplot, fill_color='54278F'x, histogram=1, outline = 0)
        index_4 = barplot(xpoly, ypoly4, /overplot, fill_color='F03B20'x, histogram=1, outline = 0)
        index_3 = barplot(xpoly, ypoly3, /overplot, fill_color='FD8D3C'x, histogram=1, outline = 0)
        index_2 = barplot(xpoly, ypoly2, /overplot, fill_color='FED98E'x, histogram=1, outline = 0)
        index_1 = barplot(xpoly, ypoly1, /overplot, fill_color='EFF3FF'x, histogram=1, outline = 0)

        ; draw level line
        warn_line = polyline([x(0), x(96)], [2.5,2.5], linestyle=2, color='black', /data, thick=2)
        mid_line1 = polyline([x(0), x(96)], [5.5,5.5], linestyle=2, color='black', /data, thick=0.5)
        mid_line2 = polyline([x(0), x(96)], [7.5,7.5], linestyle=2, color='black', /data, thick=0.5)
        mid_line3 = polyline([x(0), x(96)], [10.5,10.5], linestyle=2, color='black', /data, thick=0.5)

        ; draw level line info
        mesg1_1 = text(x[4], 1.2, 'LOW', font_color='black', font_size=8, font_name='Nanum
Gothic', /data)
        mesg1_2 = text(x[4], 0.5, '(낮음)', font_color='black', font_size=8, font_name='Nanum
Gothic', /data)
        mesg2_1 = text(x[4], 4.0, 'MOD', font_color='black', font_size=8, font_name='Nanum
Gothic', /data)
```

```
        mesg2_2 = text(x[4], 3.3, '(보통)', font_color='black', font_size=8, font_name='Nanum
Gothic', /data)
        mesg3_1 = text(x[4], 6.5, 'HIGH', font_color='black', font_size=8, font_name='Nanum
Gothic', /data)
        mesg3_2 = text(x[4], 5.8, '(높음)', font_color='black', font_size=8, font_name='Nanum
Gothic', /data)
        mesg4_1 = text(x[1], 9.0, 'VERY HIGH', font_color='black', font_size=8, font_name=
'NanumGothic', /data)
        mesg4_2 = text(x[2], 8.3, '(매우높음)', font_color='black', font_size=8, font_name=
'Nanum Gothic', /data)
;         mesg5_1 = text(x[2], 12.5, 'EXTREME', font_size=8, font_name='NanumGothic', font_
color='E6E6E6'x, /data)
;         mesg5_2 = text(x[4], 11.8, '(위험)', font_size=8, font_name='NanumGothic', font_color
='E6E6E6'x, /data)
        mesg5_1 = text(x[2], 12.5, 'EXTREME', font_size=8, font_name='NanumGothic',
font_color=[230,230,230], /data)
        mesg5_2 = text(x[4], 11.8, '(위험)', font_size=8, font_name='NanumGothic',
font_color=[230,230,230], /data)

        ; read files for calculation of proper exposure time
        if m1_b(count_b-1) lt 10 then month_temp = '0'+strtrim(string(round(m1_b(count_b-1))),1)
        if m1_b(count_b-1) ge 10 then month_temp = strtrim(string(round(m1_b(count_b-1))),1)

        index_file_climate = file_search(root_dir_b, 'Climatology_'+month_temp+'_5station.txt')
        index_file_date = file_search(root_dir_b, 'time_arr_'+month_temp+'.txt')
;         print, 'Climatology_'+month_temp+'_5station.txt', 'time_arr_'+month_temp+'.txt'
        readcol, index_file_climate, uvb_105, uvb_132, uvb_138, uvb_165, uvb_013;, 'I, I, I, I, I'
        readcol, index_file_date, list_day, list_hour, list_min;, 'I, I, I, I, I'

        day_sync_arr=fltarr(4,97)
        date_dum1=where(list_day eq d_b(count_b-1) and list_hour ge 4. and list_hour lt 20.)
        ;print, date_dum1
```

```
;print, n_elements(date_dum1)
for m=0, 96 do begin
  if m le (count_b-1) then begin
    if m lt 96 then day_sync_arr(0,m) = y(m)
    if m eq 96 then day_sync_arr(0,96) = y(96)
  endif else if site_name(i) eq '013' then begin
    if m lt 96 then day_sync_arr(0,m) = uvb_013(date_dum1(m))
    if m eq 96 then day_sync_arr(0,96) = uvb_013(date_dum1(95)+1)
  endif else if site_name(i) eq '105' then begin
    if m lt 96 then day_sync_arr(0,m) = uvb_105(date_dum1(m))
    if m eq 96 then day_sync_arr(0,96) = uvb_105(date_dum1(95)+1)
  endif else if site_name(i) eq '115' then begin
    if m lt 96 then day_sync_arr(0,m) = uvb_105(date_dum1(m))
    if m eq 96 then day_sync_arr(0,96) = uvb_105(date_dum1(95)+1)
  endif else if site_name(i) eq '132' then begin
    if m lt 96 then day_sync_arr(0,m) = uvb_132(date_dum1(m))
    if m eq 96 then day_sync_arr(0,96) = uvb_132(date_dum1(95)+1)
  endif else if site_name(i) eq '138' then begin
    if m lt 96 then day_sync_arr(0,m) = uvb_138(date_dum1(m))
    if m eq 96 then day_sync_arr(0,96) = uvb_138(date_dum1(95)+1)
  endif else if site_name(i) eq '165' then begin
    if m lt 96 then day_sync_arr(0,m) = uvb_165(date_dum1(m))
    if m eq 96 then day_sync_arr(0,96) = uvb_165(date_dum1(95)+1)
  endif else if site_name(i) eq '108' then begin
    if m lt 96 then day_sync_arr(0,m) = uvb_105(date_dum1(m))
    if m eq 96 then day_sync_arr(0,96) = uvb_105(date_dum1(95)+1)
  endif
  if m lt 96 then day_sync_arr(1,m) = list_day(date_dum1(m))
  if m eq 96 then day_sync_arr(1,96) = list_day(date_dum1(95)+1)
  if m lt 96 then day_sync_arr(2,m) = list_hour(date_dum1(m))
  if m eq 96 then day_sync_arr(2,96) = list_hour(date_dum1(95)+1)
  if m lt 96 then day_sync_arr(3,m) = list_min(date_dum1(m))
```

```
          if m eq 96 then day_sync_arr(3,96) = list_min(date_dum1(95)+1)
        endfor
;       print, 'Important: ', count_b, day_sync_arr(0,count_b-1), day_sync_arr(1,count_b-1),
day_sync_arr(2,count_b-1), day_sync_arr(3,count_b-1),y_b(count_b-1), m1_b(count_b-1),
d_b(count_b-1), h_b(count_b-1), m2_b(count_b-1)
;       print, day_sync_arr(0,0), day_sync_arr(1,0), day_sync_arr(2,0), day_sync_arr(3,0)
;       print, day_sync_arr(0,96), day_sync_arr(1,96), day_sync_arr(2,96), day_sync_arr(3,96)
        ;stop
        ;print, day_sync_arr(0,0:96)

        optimal_time = fltarr(6)
;       print, 'date sync: ', (((h_obs*60)+mn_obs)-((day_sync_arr(2,count_b-1)*60)+day_
sync_arr(3,count_b-1)))
        if  (((h_obs*60)+mn_obs)-((day_sync_arr(2,count_b-1)*60)+day_sync_arr(3,count_b-1)))
le 20 then begin
        if l eq 0 then begin
          if y(count_b-1) lt 0.1 then y(count_b-1) = 0.1

          for mn=count_b, 95 do begin
            td = (mn-(count_b-1))*10
            uvi_temp_d = total(day_sync_arr(0,count_b-1:mn))/(mn-(count_b-2))
            td_uvi = td*uvi_temp_d
            ratio_d = 1.5239+0.6735*(uvi_temp_d^0.0887)
            ;print, day_sync_arr(0,count_b-1:mn)
            ;print, td_uvi, 144/ratio_d, td, uvi_temp_d, day_sync_arr(0,mn), day_sync_arr
(0,mn-1)
            if td_uvi ge (144/ratio_d) then begin
              optimal_time(0)=day_sync_arr(0,mn) & optimal_time(1)=day_sync_arr(1,mn) &
optimal_time(2)=day_sync_arr(2,mn)
              vitd_exp_time = td
              break
            endif else begin
```

```
            vitd_exp_time = 9
          endelse
        endfor
        ;stop
        for nm=count_b, 95 do begin
          te = (nm-(count_b-1))*10
          uvi_temp_e = total(day_sync_arr(0,count_b-1:nm))/(nm-(count_b-2))
          te_uvi = te*uvi_temp_e
          ;print, te_uvi,  te, uvi_temp_e, day_sync_arr(0,nm), day_sync_arr(0,nm-1)
          if te_uvi ge 300 then begin
            optimal_time(3)=day_sync_arr(0,nm) & optimal_time(4)=day_sync_arr(1,nm) &
optimal_time(5)=day_sync_arr(2,nm)
            crit_exp_time = te
            break
          endif else begin
            crit_exp_time = 9
          endelse
        endfor
        ;stop
        crit_exp_time_prev = (4000*4.5)/(60*y(count_b-1))*0.4
        vitd_exp_time_prev = (20*4.5)/(0.25*2*y(count_b-1))*0.4
;          print, 'results: ', vitd_exp_time, crit_exp_time, vitd_exp_time_prev, crit_exp_time_
prev;, optimal_time

        ;print, 'recommended exposure time=', strtrim(vitd_exp_time,1),'    ', strtrim(round
(vitd_exp_time/10)*10,1),'    ', strtrim(vitd_exp_time_v1,1), crit_exp_time,'    ', 'UV-b index=',
strtrim(y(count_b-1),1)

        now_time = bin_date(systime())

        ;if vitd_exp_time ge 11 and now_time(3) lt 13 and now_time(3) ge 10 then begin
        if vitd_exp_time le 500 and vitd_exp_time ge 10 and now_time(3) lt 14 and
```

```
now_time(3) ge 9 then begin
        if i eq 6 then begin
          if count_b lt 100 then begin
            vitd_leg_bg = polygon([x(count_b+11), x(count_b-13), x(count_b-13), x(count_b+11)], [y(count_b-1)+5.2,y(count_b-1)+5.2,y(count_b-1)+8.1,y(count_b-1)+8.1], /data, fill_color='white', color='white')
            vitd_leg_t = text(x(count_b-1), y(count_b-1)+6.9, '일일 비타민D 권장량', font_name='NanumGothic', /data, font_size=16, alignment=0.5)
            vitd_leg_b = text(x(count_b-1), y(count_b-1)+5.5, '생성 적정 노출시간', font_name='NanumGothic', /data, font_size=16, alignment=0.5);일일 비타민D 권장량 생성 적정 노출시간
          endif; else begin
          ; vitd_leg_t = text(x(count_b-17), y(count_b-1)+6.9, '비타민 생성', /data, font_size=16)
          ;vitd_leg_b = text(x(count_b-18), y(count_b-1)+5.5, '적정 노출시간', /data, font_size=16)
          ; endelse
        endif
        ;vitd_box = polygon(vitd_dt_x, vitd_dt_y, /data, fill_color='74c476'x, thick=2)
        if vitd_exp_time ge 1000 then begin
          ;vitd_ellipse = ellipse(x(count_b-1), y(count_b-1)+3.7, major=0.115, minor=1.3, /data, fill_color='74c476'x, thick=2)
          ;vitd_box = polygon([x(count_b+13), x(count_b+11), x(count_b-14), x(count_b-15), x(count_b-15), x(count_b-14), x(count_b+11), x(count_b+13)], [y(count_b-1)+5.2, y(count_b-1)+5.0,y(count_b-1)+5.0,y(count_b-1)+5.2,y(count_b-1)+8.1,y(count_b-1)+8.3, y(count_b-1)+8.3,y(count_b-1)+8.1], /data, fill_color='74c476'x, color='black')
        endif else begin
          ;vitd_ellipse = ellipse(x(count_b-1), y(count_b-1)+3.7, major=0.095, minor=1.3, /data, fill_color='74c476'x, thick=2)
          a=x(count_b+5)
          b=x(count_b-7)
          c=(x(count_b+13)-x(count_b+12))/20
```

```
            d=y(count_b-1)
            e=2.5
            f=4.8
            g=0.02
        ;       g2=0.04
            vitd_box = polygon([a+5*c, a+2*c, a+c, a, a-c, a-2*c, a-5*c, b+5*c, b+2*c, b+c, b, b-c, b-2*c, b-5*c, b-5*c, b-2*c, b-c, b, b+c, b+2*c, b+5*c, a-5*c, a-2*c, a-c, a, a+c, a+2*c, a+5*c], [d+e+5*g, d+e+4*g, d+e+2*g, d+e, d+e-2*g, d+e-4*g, d+e-5*g, d+e-5*g, d+e-4*g, d+e-2*g, d+e, d+e+2*g, d+e+4*g, d+e+5*g, d+f-5*g, d+f-4*g, d+f-2*g, d+f, d+f+2*g, d+f+4*g, d+f+5*g, d+f+5*g, d+f+4*g, d+f+2*g, d+f, d+f-2*g, d+f-4*g, d+f-5*g], /data, fill_color='74c476'x, color='black', thick=1.8)
            ;vitd_box = plot([x(count_b+15), x(count_b+14),x(count_b+13), x(count_b+12), x(count_b+11), x(count_b-13), x(count_b-14), x(count_b-15), x(count_b-16), x(count_b-17), x(count_b-17), x(count_b-16), x(count_b-15), x(count_b-14), x(count_b-13), x(count_b+11), x(count_b+12), x(count_b+13), x(count_b+14), x(count_b+15)], [y(count_b-1)+6.0,y(count_b-1)+5.7,y(count_b-1)+5.4,y(count_b-1)+5.1,y(count_b-1)+4.8,y(count_b-1)+4.8,y(count_b-1)+5.1,y(count_b-1)+5.4,y(count_b-1)+5.7,y(count_b-1)+6.0,y(count_b-1)+7.5,y(count_b-1)+7.8,y(count_b-1)+8.1,y(count_b-1)+8.4,y(count_b-1)+8.7,y(count_b-1)+8.7,y(count_b-1)+8.4,y(count_b-1)+8.1,y(count_b-1)+7.8,y(count_b-1)+7.5], /data, fill_color='74c476'x, color='black', /overplot)
            endelse
            ;vitd_min = text(x(count_b-1), y(count_b-1)+3.2, strtrim(round(vitd_exp_time),1)+'~'+strtrim(round(crit_exp_time),1)+'분', font_name='NanumGothic', /data, font_size=18, font_style=0, alignment=0.5)
            vitd_min = text(x(count_b-1), y(count_b-1)+3.2, strtrim(round(vitd_exp_time/10)*10,1) +'분', font_name='NanumGothic', /data, font_size=18, font_style=0, alignment=0.5)
            vitd_arrow = arrow([x(count_b-1), x(count_b-1)],[y(count_b-1)+2.4, y(count_b-1)+0.5], /data, /current, head_size=0.7, thick=2)
          endif
          if vitd_exp_time gt 9999 then begin
            vitd_dt_x = [x(count_b-2)-0.053, x(count_b+2)+0.043, x(count_b+2)+0.043, x(count_b-2)-0.053]
```

```
            vitd_dt_y = [y(count_b-1)+2.4, y(count_b-1)+2.4, y(count_b-1)+4.6, y(count_
b-1)+4.6]
            a=x(count_b+7)
            b=x(count_b-9)
            c=(x(count_b+15)-x(count_b+14))/20
            d=y(count_b-1)
            e=2.5
            f=4.8
            g=0.02
            vitd_box = polygon([a+5*c, a+2*c, a+c, a, a-c, a-2*c, a-5*c, b+5*c, b+2*c,
b+c, b, b-c, b-2*c, b-5*c, b-5*c, b-2*c, b-c, b, b+c, b+2*c, b+5*c, a-5*c, a-2*c,
a-c, a, a+c, a+2*c, a+5*c], [d+e+5*g, d+e+4*g, d+e+2*g, d+e, d+e-2*g, d+e-4*g,
d+e-5*g, d+e-5*g, d+e-4*g, d+e-2*g, d+e, d+e+2*g, d+e+4*g, d+e+5*g, d+f-5*g,
d+f-4*g, d+f-2*g, d+f, d+f+2*g, d+f+4*g, d+f+5*g, d+f+5*g, d+f+4*g, d+f+2*g, d+f,
d+f-2*g, d+f-4*g, d+f-5*g], /data, fill_color='F03B20'x, color='black', thick=1.8)
            ;vitd_box = polygon(vitd_dt_x, vitd_dt_y, /data, fill_color='F03B20'x, thick=2)
            ;vitd_ellipse = ellipse(x(count_b-1), y(count_b-1)+3.7, major=0.075, minor=1.3,
/data, fill_color='F03B20'x, thick=2)
            vitd_leg_b = text(x(count_b-1), y(count_b-1)+3.3, '노출자제', font_name='Nanum
Gothic', /data, font_size=16, font_style=1, alignment=0.5)
            vitd_arrow = arrow([x(count_b-1), x(count_b-1)],[y(count_b-1)+2.4, y(count_b-1)+
0.5], /data, /current, head_size=0.8, thick=2)
          endif
        endif
        endif

        ; draw graph title
        if l eq 0 then begin
          if fix(mn_obs) lt 10 then begin
            fixed_mn_obs = '0' + strtrim(string(fix(mn_obs)), 1)
          endif else begin
            fixed_mn_obs = strtrim(string(fix(mn_obs)), 1)
```

```
    endelse

    if i eq 0 then uvi.title = '고산 ' + strtrim(string(year_tday), 1) + '년 ' + strtrim(string
(month_tday), 1) + '월 ' + strtrim(string(day_tday), 1) + '일(' + dow_week + ') ' + strtrim
(string(fix(h_obs)), 1) + ':' + fixed_mn_obs + ' 관측'
    if i eq 1 then uvi.title = '강릉 ' + strtrim(string(year_tday), 1) + '년 ' + strtrim(string
(month_tday), 1) + '월 ' + strtrim(string(day_tday), 1) + '일(' + dow_week + ') ' + strtrim
(string(fix(h_obs)), 1) + ':' + fixed_mn_obs + ' 관측'
    if i eq 2 then uvi.title = '울릉도 ' + strtrim(string(year_tday), 1) + '년 ' + strtrim
(string(month_tday), 1) + '월 ' + strtrim(string(day_tday), 1) + '일(' + dow_week + ') ' +
strtrim(string(fix(h_obs)), 1) + ':' + fixed_mn_obs + ' 관측'
    if i eq 3 then uvi.title = '안면도 ' + strtrim(string(year_tday), 1) + '년 ' + strtrim
(string(month_tday), 1) + '월 ' + strtrim(string(day_tday), 1) + '일(' + dow_week + ') ' +
strtrim(string(fix(h_obs)), 1) + ':' + fixed_mn_obs + ' 관측'
    if i eq 4 then uvi.title = '포항 ' + strtrim(string(year_tday), 1) + '년 ' + strtrim(string
(month_tday), 1) + '월 ' + strtrim(string(day_tday), 1) + '일(' + dow_week + ') ' + strtrim
(string(fix(h_obs)), 1) + ':' + fixed_mn_obs + ' 관측'
    if i eq 5 then uvi.title = '목포 ' + strtrim(string(year_tday), 1) + '년 ' + strtrim(string
(month_tday), 1) + '월 ' + strtrim(string(day_tday), 1) + '일(' + dow_week + ') ' + strtrim
(string(fix(h_obs)), 1) + ':' + fixed_mn_obs + ' 관측'
    if i eq 6 then uvi.title = '서울 ' + strtrim(string(year_tday), 1) + '년 ' + strtrim(string
(month_tday), 1) + '월 ' + strtrim(string(day_tday), 1) + '일(' + dow_week + ') ' + strtrim
(string(fix(h_obs)), 1) + ':' + fixed_mn_obs + ' 관측'
  endif

  if l eq 1 then begin
    if i eq 0 then uvi.title = '고산 ' + strtrim(string(year_yday), 1) + '년 ' + strtrim
(string(month_yday), 1) + '월 ' + strtrim(string(day_yday), 1) + '일(' + dow_week_y + ')'
    if i eq 1 then uvi.title = '강릉 ' + strtrim(string(year_yday), 1) + '년 ' + strtrim
(string(month_yday), 1) + '월 ' + strtrim(string(day_yday), 1) + '일(' + dow_week_y + ')'
    if i eq 2 then uvi.title = '울릉도 ' + strtrim(string(year_yday), 1) + '년 ' + strtrim
(string(month_yday), 1) + '월 ' + strtrim(string(day_yday), 1) + '일(' + dow_week_y + ')'
```

```
        if i eq 3 then uvi.title = '안면도 ' + strtrim(string(year_yday), 1) + '년 ' + strtrim
(string(month_yday), 1) + '월 ' + strtrim(string(day_yday), 1) + '일(' + dow_week_y + ')'
        if i eq 4 then uvi.title = '포항 ' + strtrim(string(year_yday), 1) + '년 ' + strtrim
(string(month_yday), 1) + '월 ' + strtrim(string(day_yday), 1) + '일(' + dow_week_y + ')'
        if i eq 5 then uvi.title = '목포 ' + strtrim(string(year_yday), 1) + '년 ' + strtrim
(string(month_yday), 1) + '월 ' + strtrim(string(day_yday), 1) + '일(' + dow_week_y + ')'
        if i eq 6 then uvi.title = '서울 ' + strtrim(string(year_yday), 1) + '년 ' + strtrim
(string(month_yday), 1) + '월 ' + strtrim(string(day_yday), 1) + '일(' + dow_week_y + ')'
      endif

      uvi.title.position = [0.53, 0.97]

      if l eq 2 then begin
        if i eq 0 then uvi.title = '고산 ' + strtrim(string(year_tdby), 1) + '년 ' + strtrim
(string(month_tdby), 1) + '월 ' + strtrim(string(day_tdby), 1) + '일(' + dow_week_dby + ')'
        if i eq 1 then uvi.title = '강릉 ' + strtrim(string(year_tdby), 1) + '년 ' + strtrim
(string(month_tdby), 1) + '월 ' + strtrim(string(day_tdby), 1) + '일(' + dow_week_dby + ')'
        if i eq 2 then uvi.title = '울릉도 ' + strtrim(string(year_tdby), 1) + '년 ' + strtrim
(string(month_tdby), 1) + '월 ' + strtrim(string(day_tdby), 1) + '일(' + dow_week_dby + ')'
        if i eq 3 then uvi.title = '안면도 ' + strtrim(string(year_tdby), 1) + '년 ' + strtrim
(string(month_tdby), 1) + '월 ' + strtrim(string(day_tdby), 1) + '일(' + dow_week_dby + ')'
        if i eq 4 then uvi.title = '포항 ' + strtrim(string(year_tdby), 1) + '년 ' + strtrim
(string(month_tdby), 1) + '월 ' + strtrim(string(day_tdby), 1) + '일(' + dow_week_dby + ')'
        if i eq 5 then uvi.title = '목포 ' + strtrim(string(year_tdby), 1) + '년 ' + strtrim
(string(month_tdby), 1) + '월 ' + strtrim(string(day_tdby), 1) + '일(' + dow_week_dby + ')'
        if i eq 6 then uvi.title = '서울 ' + strtrim(string(year_tdby), 1) + '년 ' + strtrim
(string(month_tdby), 1) + '월 ' + strtrim(string(day_tdby), 1) + '일(' + dow_week_dby + ')'
      endif

      ; save img file
      imgpath = '/n0005/LOCAL_INTEGRATION/webapps/ccic/ROOT/data/graph/vitd/';
'/Users/WaveGuide/Documents/workspace/2014/UVI/graph/'; 'C:₩Users₩WaveGuide₩
```

```
Documents₩workspace₩2014₩KMA₩CCIC₩UVI₩';
        imgname = site_name(i)+'_'+day_index(l)
        uvi.Save, imgpath+imgname+'.png', bit_depth=2, /transparent, width=600, height=350
        uvi.Close

        ; exit to this FOR statement when it has already previous day graphs
        if  hour_tday le 4 && minute_tday ge 10 then begin
          print, 'THE PREVIOUS GRAPHS HAVE BEEN ALREADY MADE.'
          break
        endif
      endfor
      print, site_name(i)+ ' finished.'
  endfor
END
```

〈 참고문헌 〉

[1] 기후변화감시기술노트 2014-7, 총자외선지수 산출방법 및 프로그램

[2] 기상청(2015), 기후변화감시자료 산출기술 개발(II)

[3] 기상청(2015), 기후변화감시자료 품질관리 기술개발(II)

[4] Kazantzidis, A., Bais, A.F., Zempila, M. M., Kazadzis, S., Peter, N., Koskela, T., & Slaper, H. (2009). Calculations of the human vitamin D exposure from UV spectral measurements at three European stations. Photochemical & Photobiological Sciences, 8(1), 45-51

[5] Madronich, S., R. L. McKenzie, L. O. Björn, and M. M. Caldwell(1998), Changes in biologically active ultraviolet radiation reaching the Earth's surface, J. Photochem. Photobiol., B: Biology, 46, 1, 5-19

[6] McKenzie, R. L., Liley, J.B., & Björn, L. O. (2009). UV Radiation: Balancing Risks and Benefits†. Photochemistry and Photobiology, 85(1), 88-98

[7] 보건복지부(2015), 2015 한국인 영양소 섭취기준

기후변화감시 기술노트 발간 목록

2014년도

1. 기후변화감시 기술노트 2014-01 : 통합 지구대기화학관측(IGACO)의 오존 및 자외복사 이행 계획
2. 기후변화감시 기술노트 2014-02 : GC-μECD를 활용한 대기 농도 수준의 SF_6 분석 가이드라인
3. 기후변화감시 기술노트 2014-03 : 건조 공기 포집 및 안정도 평가를 위한 가이드라인
4. 기후변화감시 기술노트 2014-04 : 강수화학 국제비교실험 참가방법 및 결과
5. 기후변화감시 기술노트 2014-05 : WMO 지구대기감시(GAW) 전략계획(2012-2015) : WMO GAW 전략계획 (2008-2015) 보강
6. 기후변화감시 기술노트 2014-06 : 표준 에어로졸 샘플링 시스템
7. 기후변화감시 기술노트 2014-07 : 총자외선지수 산출방법 및 프로그램
8. 기후변화감시 기술노트 2014-08 : 강수화학 측정자료의 분석 절차

2015년도

1. 기후변화감시 기술노트 2015-01 : 오존층에 관한 질문과 답변 20가지 : 2014년판
2. 기후변화감시 기술노트 2015-02 : WMO 육불화황 세계표준센터의 3차 표준가스 제조 방법
3. 기후변화감시 기술노트 2015-03 : Analytical Methods for Atmospheric SF_6 Using GC-μECD (WMO/GAW Report NO. 222)
4. 기후변화감시 기술노트 2015-04 : 강수화학(대기침적) 측정 및 분석 매뉴얼
5. 기후변화감시 기술노트 2015-05 : 대기조성의 측정

2016년도

1. 기후변화감시 기술노트 2016-01 : 온실가스 관측업무 매뉴얼
2. 기후변화감시 기술노트 2016-02 : 인체 비타민D 생성을 위한 태양자외선복사 노출시간 산정 및 서비스

기상청
서울특별시 동작구 여의대방로16길 61 기후변화감시과
대표전화 : 02-2181-0647 www.climate.go.kr