2025년 하계 산업보건학회

jinha

2025-08-15

Table of contents

소개

2025년 하계 산업보건학회 PDC 강의록입니다.

flyinyou@gmail.com

1 개요

2 서론

2.1 직업병 역학연구에서 용량반응 관계의 중요성

직업병 역학연구에서 용량반응 관계는 특정 유해물질에 대한 노출량과 그로 인해 발생하는 질병 또는 건강 영향의 크기 사이의 연관성을 설명하는 근본적인 개념이다. 이 개념은 직업 및 환경 보건 분야에서 인과적 추론을 강화하고, 과학적 증거의 강점과 한계를 평가하며, 잠재적 위험을 정확하게 평가하는 데 필수적이다.

용량반응 관계는 노출 수준이 증가함에 따라 유해 영향이 커진다는 정량적인 연관성을 보여줌으로써, 특정 노출이 단순히 질병과 연관된 것이 아니라 질병 발생의 직접적인 원인이 될 가능성이 높다는 인과성 주장을 강력하게 뒷받침한다. 이러한 정량적인 연관성은 안전한 노출 한계 및 규제 표준을 설정하는 데 필요한 실질적인 근거를 제공하며, 공중 보건 개입의 우선순위를 효과적으로 결정하는 데 중요한 지침이 된다.

본 강의는 직업병 역학연구에서 용량반응 관계의 중요성과, 이를 해석하는 과정에 내재된 복잡성과 다양한 도전 과제를 이해하고 실습하는 것을 목표로 한다. 특히, 유해물질 노출수준에 따른 질병 발생의 정량적 관계를 모델링하고 활용하는 과정에서 일반적으로 전제되는 주요 가정들, 즉 선형성, 노출과 용량의 관계, 노출 분포의 정규성, 질병 발생의 측정 가능성, 그리고 노출과 질병 관련성의 상호작용 부재와 같은 가정들이 실제 연구 환경에서 어떻게 위배될 수 있는지, 그리고 그로 인해 어떠한 문제점들이 발생하는지를 다룬다.

또한, 이러한 문제점들을 해결하기 위한 고급 분석 기법과 정책적 함의를 제시함으로써, 직업병 역학연구의 과학적 엄밀성을 높이고, 근로자 건강 보호를 위한 실질적이고 효과적인 예방 전략 수립에 기여하고자 한다.

3 용량반응 관계의 기본 개념 및 역할

3.1 정의 및 인과성 확립 기여

용량반응 관계는 특정 유해 요인(예: 화학 물질, 방사선, 영양소)의 노출량 또는 수준과 그로 인한 건강 결과(예: 질병 발생률 또는 사망률) 간의 정량적인 연관성을 의미한다. 이 관계는 역학 연구의 핵심적인 요소로서, 특정 유해요인이 건강 결과에 어떻게 영향을 미치는지를 이해하기 위한 기본적인 논리를 제공한다.

용량반응 관계인 노출이 증가함에 따라 유해 효과의 위험이 증가한다는 명확한 용량반응 기울기(Gradient)는 인과성을 뒷받침하는 핵심적인 요소이다. 단순히 통계적 연관성을 넘어, 브래드포드 힐(Bradford Hill)의 인과성 기준 중 하나인 '생물학적 기울기(Biological Gradient)'에 해당된다.

표 1 . 브래포드 힐의 인과관계 요소

요소	영어 표기
연관성의 강도	Strength of association
일관성	Consistency
특이성	Specificity
시간적 선후 관계	Temporality
용량-반응 관계	Dose-response
생물학적 그럴듯성	Plausibility
일치성	Coherence
실험	Experiment
유추	Analogy

이는 노출량의 체계적인 변화가 건강 결과의 변화를 유도한다는 것을 보여주며, 인과적 개입의 효과를 예측하고 관련 정책을 수립하는 데 근거를 제시한다. 즉, 역학 연구가 단순한 관찰을 넘어 실질적인 개입의 근거를 마련하는 데 필수적인 단계임을 의미한다. 따라서 용량반응 관계는 노출이 질병 발생의 원인이라는 강력한 생물학적 증거를 제공하여 보건 정책 및 규제 설정의 기반이 된다.

3.2 선형 및 비선형 관계의 유형

용량반응 관계는 그 형태에 따라 선형적일 수도 있고 비선형적일 수도 있다.

- 선형 관계 (Linear Relationship): 가장 단순한 형태로, 노출량 증가에 비례하여 질병 발생 위험이 증가하는 경우이다. 예를 들어, 유해인자 노출이 2배 증가하면 질병 발생도 2배 증가하는 것이 이에 해당한다.
- 비선형 관계 (Non-linear Relationships): 실제 세계의 노출-반응 데이터는 종종 선형 관계보다 더 복잡한 패턴을 보인다.
- 역치 효과 (Threshold Effects): 특정 노출 수준(역치) 이하에서는 건강 효과가 나타나지 않다가, 역치를 넘어서면 효과가 증가하는 경우이다. 이는 독성학에서 안전 노출 한계를 설정하는 데 흔히 사용되는 모델이다.
- 포화 효과 (Saturation Effects): 노출량이 특정 수준에 도달하면 반응이 더 이상 증가하지 않고 평평해지는 경우이다. 이는 생체 내 수용체나 대사 경로가 포화될 때 나타날 수 있다.
- U자형 또는 J자형 관계 (U-shaped or J-shaped Relationships): U자형 관계는 노출량이 낮거나 높을 때 효과가 더 크고, 중간 노출량에서는 효과가 낮은 경우를 의미한다. J자형 관계는 특정 노출량까지는 유익하거나 무해하다가 그 이상에서는 유해한 효과가 나타나는 경우로, 알코올 섭취와 심혈관 질환의 관계에서 관찰될 수 있다.

직업병 역학연구에서 단순한 선형 용량반응 관계만을 가정하는 것은 생물학적 현실을 간과하고 위험 평가를 왜곡할 수 있다. 생체 내의 복잡한 생물학적 메커니즘(예: 수용체 결합, 대사 경로, 세포 독성)은 종종 비선형적인 반응을 유도한다. 예를 들어, 특정 수용체가 포화되면 아무리 노출량을 늘려도 반응이 더 이상 증가하지 않거나, 필수 영양소의 경우 너무 적거나 너무 많아도 문제가 되는 U자형 관계가 나타날 수 있다. 이러한 비선형성을 정확하게 모델링하지 않으면, 안전한 노출 수준을 과대평가하거나 과소평가하여 공중 보건에 심각한 영향을 미칠 수 있다. 따라서 역치, U자형, J자형 등 다양한 비선형 모델을 고려하고 적용하는 것이 보다 정확하고 현실적인 위험 관리를 위해 필수적이다.

표 2 용량반응 관계 유형 및 특성

관계 유형	특징	생물학적/역학적 의미	예시
선형 (Linear)	노출량 증가에 비례하여 반응이 선형적으로 증가	노출량에 따라 위험이 지속적으로 증가하며, 안전 역치가 없을 수 있음	방사선 노출과 암 발생
역치 (Threshold)	특정 노출 수준(역치) 이하에서는 효과가 없고, 역치 초과 시 반응 증가	안전한 노출 한계(역치)의 존재를 시사, 독성학에서 중요	중금속 노출과 특정 독성 반응
포화 (Saturation)	노출량이 특정 수준에 도달하면 반응이 더 이상 증가하지 않고 평탄화	생체 내 수용체 또는 대사 경로의 포화로 인한 반응 제한	특정 약물의 최대 효과 도달

U자형 (U-shaped)	노출량이 낮거나 높을 때 효과가 크고, 중간 노출량에서 효과가 가장 작음	최적의 노출 수준이 존재함을 시사, 부족과 과잉 모두 유해	비타민 A 섭취와 선천적 결함
J자형 (J-shaped)	특정 노출량까지는 유익하거나 무해하다가, 그 이상에서 유해한 효과가 급증	저용량의 유익성 또는 무해성 후 고용량의 유해성	알코올 섭취와 심혈관 질환

4 용량반응 관계 해석의 핵심 가정과 그 복잡성

4.1 노출과 용량의 관계: 체내 흡수, 생체이용률 및 독성동태학적 고려사항

직업 환경에서 외부 노출(Exposure)과 실제로 생체 내에 도달하여 생물학적 효과를 유발하는 내부 용량(Dose) 간의 중요한 차이가 있다. 용량반응 관계는 유해인자가 생체에 '도달'하는 양(dose)에 대한 생물학적 반응을 설명하며, 이는 흡수(Absorption), 분포(Distribution), 대사(Metabolism), 배설(Elimination)이라는 독성동태학적(ADME) 과정을 통해 결정된다.

노출-반응 정보는 주로 약물 개발에서 자주 활용되는데, 약물의 안전성과 유효성을 결정하는 핵심이며, 혈중 농도 등과 반응 간의 관계를 확립하는 것이 매우 중요하다. 이러한 원칙은 직업병 역학에서도 유사하게 적용될 수 있다. 직업 환경에서 개인의 생체 내 독성동태학적 특성과 노출 경로에 따라 동일한 외부 노출이 상이한 내부 용량을 유발할 수 있어, '노출'과 '용량'은 동의어가 아니다. 예를 들어, 신장 기능이 저하된 근로자는 특정 화학물질의 배설이 지연되어 동일한 외부 노출에도 불구하고 더 높은 내부 용량을 가질 수 있으며, 이는 독성 반응에 대한 취약성을 증가시킨다. 또한, 노출 경로나 개인의 유전적 다형성, 질병 상태, 생활 습관(흡연, 음주) 등 다양한 요인이 물질의 체내 흡수, 분포, 대사, 배설에 영향을 미쳐 실제 유효 용량을 변화시킬 수 있다.

외부 노출만으로 용량반응 관계를 추정하는 것은 오차를 유발할 수 있으며, 혈중 농도나 대사산물과 같은 생체 지표(biomarker)를 통한 내부 용량 측정의 중요성이 부각된다. 따라서 내부 용량을 반영하는 생체 지표를 활용하는 것이 용량반응 관계의 정확성을 높이는 데 필수적이다. 그러나 직업병 역학조사에서는 만성 질환의 경우 생체 지표가 내부 용량을 반영하기 어렵기 때문에 환경 노출을 통해 내부 용량을 추정하는 것이 필수적이다.

4.2 노출 분포의 특성: 정규성 가정의 한계 및 로그정규 분포의 적용

직업 환경 노출 데이터는 일반적으로 정규 분포(Gaussian distribution)를 따른다는 가정이 종종 사용된다. 그러나 실제 산업 위생 샘플링 데이터는 "오른쪽으로 치우친(skewed to the right)" 비대칭 분포를 보이는 경우가 많다. 이는 노출 값이 0보다 작을 수 없다는 하한선이 존재하기 때문이면서, 노출이 높다고 알려진 경우에 측정한 데이터가 주로 남아 있기 때문이기도 하다.

직업 환경에서의 노출은 작업 특성, 공정 변화, 개인의 작업 방식 등 다양한 요인에 의해 매우 이질적일 수 있다. 이러한 복잡성은 노출 데이터가 통계적으로 이상적인 정규 분포를 따르기 어렵게 만든다. 노출 분포의 비정규성은 통계적 분석의 가정(예: 선형 회귀 모델의 잔차 정규성)을 위배하여 추정치의 편향이나 비효율성을 초래할 수 있다. 따라서 비모수적 방법, 또는 혼합 모델(mixed models)과 같은 고급 통계 기법을 적용하여 이러한 비정규성을 설명하는 것이 필수적이다.

4.3 질병 발생의 측정 가능성 및 절단: 우측 절단 및 경쟁 위험의 영향

치명적인 질병, 특히 사망과 같은 결과는 한 사람에게 한 번만 측정되며, 사망과 동시에 이후 연구에서는 제외되므로 '절단(Censoring)'이 발생한다. 이는 생존 분석(Survival Analysis)에서 흔히 발생하는 우측 절단(Right Censoring) 문제다. 우측 절단은 연구 대상자가 연구 기간 동안 관심 사건(예: 질병 발생 또는 사망)을 경험하지 않고 추적 관찰이 종료되거나, 다른 이유로 연구에서 이탈하는 경우 발생한다. 절단된 환자들이 연구에 남아있는 환자들과 다른 위험을 가진다면, Kaplan-Meier 방법과 같은 전통적인 생존 분석 방법은 편향된 추정치를 산출할 수 있다. 예를 들어, 직업성 암의 발생률을 연구하는 경우, 위험 물질에 노출된 후 건강이 악화되어 조기 퇴직한 근로자들이 특수건강검진을 받으면서 계속 근무하는 근로자와 암 발생 위험이 다르다면, 전통적 통계방법의 추정치는 편향될 수 있다. 이러한 편향의 방향은 조기 퇴직한 근로자들의 위험이 연구에 남아있는 근로자들보다 낮은지 높은지에 따라 과소추정 또는 과대추정될 수 있다.

- 과소추정: 건강이 좋지 않아 조기 퇴직한 근로자들은 실제로 암 발생 위험이 높았지만, 연구에서 제외되면서 전체 집단의 암 발생률이 실제보다 낮게 추정될 수 있다.
- 과대추정: 위험을 회피하기 위해 건강할 때 이직한 근로자들은 잠재적으로 위험이 낮았지만, 연구에서 제외되면서 남아있는 고위험군 근로자들로 인해 전체 집단의 암 발생률이 실제보다 높게 추정될 수 있다.

따라서 산업보건 연구에서는 이러한 '정보적 절단'의 가능성을 항상 염두에 두고, 이를 보정하기 위한 통계적 방법을 적용하는 것이 중요하다.

더 나아가, '경쟁 위험(Competing Risks)'은 특정 사건의 발생이 관심 사건의 발생을 불가능하게 만드는 경우를 말한다. 예를 들어, 심혈관 질환으로 인한 사망 시간을 연구할 때, 비심혈관 질환으로 인한 사망은 경쟁 위험이 된다. 사망은 다른 모든 질병 발생에 대한 궁극적인 경쟁 위험이 될 수 있다. 전통적인 생존 분석 방법은 경쟁 위험이 없다는 가정을 전제로 하므로, 경쟁 위험이 존재할 경우 부정확한 추정치를 산출할 수 있다.

직업병은 종종 오랜 잠복기를 거쳐 발생하며, 사망과 같은 치명적인 결과는 연구 대상자의 추적을 종료시킨다. 이러한 '절단' 데이터는 단순히 제외해서는 안 되며, 적절한 통계적 처리 없이는 질병 발생률이나 위험 추정치가 왜곡될 수 있다. 특히, 사망과 같은 '경쟁 위험'이 존재할 경우, Kaplan-Meier와 같은 표준 생존 분석은 관심 사건의 발생률을 과대평가하게 된다. 이는 특정 직업병의 발생 위험을 실제보다 높게 평가하여 부적절한 정책 결정을 초래할 수 있다. 따라서 누적 발생 함수(Cumulative Incidence Function, CIF)나 원인별 위험 함수(cause-specific hazard function), 하위분포 위험 함수(subdistribution hazard function)와 같은 경쟁 위험 분석 기법을 적용하여 보다 정확한 인과 추론을 수행해야 한다.

직업병 역학연구에서 우측 절단 및 경쟁 위험은 질병 발생률 및 위험 추정치에 심각한 편향을 초래할 수 있다. 특히 치명적인 질병의 경우, 경쟁 위험 분석을 통해 누적 발생 함수를 추정하는 것이 전통적인 생존 분석 방법보다 더 정확한 정보를 제공한다.

4.4 만성 질환의 발생 시점 불확실성: 코호트 연구에서의 고려사항 및 건강한 근로자 효과

만성 경과를 갖는 질병은, 코호트의 기준 시점에서 이미 발생한 상태일 수 있으며, 이 사람은 유해인자 노출을 피하고 있을 수 있다. 예를 들어 고혈압의 경우 언제 발생했는지 알 수 없다. 이는 만성 질환의 불분명한 발생 시점과 그로 인한 역학적 추론의 어려움을 지적한다. 만성 질환은 종종 증상이 서서히 나타나므로 정확한 발생 시점을 파악하기 어렵다. 이는 정보 편향(Information Bias) 중 하나인 '결과 측정 편향(Detection Bias)'으로 이어질 수 있다.

또한, 직업 코호트 연구에서는 '건강한 근로자 효과(Healthy Worker Effect)'라는 특정 형태의 선택 편향(Selection Bias)이 발생할 수 있다. 일반적으로 고용된 사람들은 건강해야만 일을 할 수 있으므로, 직업군 내 질병률이 일반 인구보다 낮게 나타날 수 있다. 만성 질환이 있는 사람은 애초에 특정 직업에 진입하지 못하거나, 질병 발생 후 퇴직하여 노출이 중단될 수 있다. 이러한 현상은 노출-질병 연관성을 과소평가하는 결과를 초래할 수 있다.

더 나아가, '확인 편향(Ascertainment Bias)' 또는 '샘플링 편향(Sampling Bias)'은 연구 대상자 선정 과정에서 질병이나 위험 요인이 완벽하게 식별되지 않을 때 발생하는 왜곡을 의미한다. 특히 만성 질환의 경우, 합의된 증거 기반의 사례 정의나 신뢰할 수 있는 진단 검사가 부족할 때 확인 편향의 위험이 커진다. 예를 들어, 병원이나 클리닉에서만 환자를 모집하는 경우, 중증 사례가 과대 대표될 수 있어 실제보다 나쁜 결과를 초래할 수 있다. 또한, 만성 직업병은 증상이 연령 관련 증상이나 다른 관련 요인(개인 건강 특성, 취미 노출)과 혼동될 수 있어 인과성 확립이 더욱 어렵다. 이전 직업 노출에 대한 정보가 불충분하거나 불완전한 경우가 많다는 점도 문제이다.

만성 직업병은 급성 질환과 달리 발병 시점이 불명확하고, 노출과 질병 발생 사이에 긴 잠복기가 존재한다. 이로 인해 연구 시작 시점에 이미 질병을 가지고 있는 사람들을 배제하기 어렵고, 이들이 노출을 회피했을 가능성은 선택 편향(건강한 근로자 효과)을 강화하여 노출-질병 연관성을 약화시키는 방향으로 작용할 수 있다. 따라서 만성 질환을 다루는 직업병 역학 연구는 질병 발생 시점을 최대한 정확하게 추정할 수 있는 장기 코호트 연구 설계, 주기적인 건강 검진을 통한 질병 발생 확인, 그리고 건강한 근로자 효과를 보정하기 위한 통계적 방법(예: 내부 비교 집단 설정, 표준화 사망비(SMR) 분석 시 주의)을 신중하게 고려해야 한다. 만성 직업병 연구에서는 질병 발생 시점의 불확실성과 건강한 근로자 효과, 그리고 확인 편향이 중요한 도전 과제이다. 이러한 편향을 최소화하기 위해 장기 코호트 연구 설계, 질병 발생의 정기적 확인, 그리고 건강한 근로자 효과 및 확인 편향을 고려한 분석 전략이 필수적이다.

4.5 혼합 효과 모델이 필요성

개인과 그룹의 이질성 분석 직업병 역학 연구의 데이터는 대부분 개개인(개체)의 반복 측정 자료나 여러 그룹(예: 교대 근무자와 비교대 근무자, 서로 다른 공정의 작업자)에 대한 데이터를 포함합니다. 이처럼 데이터에 내재된 **'계층적 구조'**를 무시하고 분석하면 통계적 가정이 위배되어 잘못된 결론을 내릴 수 있다. 혼합 효과 모델은 이러한 문제를 해결하기 위해 고안되었다.

- 고정 효과 (Fixed Effects): 연구자가 특별히 관심을 가지고 있는, 모든 그룹에 공통적으로 적용되는 효과를 나타다. 예를 들어, **노출량(dose)**이 질병 반응에 미치는 일반적인 영향이다.
- 변량 효과 (Random Effects): 개개인이나 그룹 간에 존재하는 무작위적인 변동성을 설명합니다. 즉, 각 개체의 생물학적 차이, 작업 환경의 미묘한 차이, 또는 측정 오차와 같은 '설명되지 않는' 변이를 모델링한다.

혼합 효과 모델은 이러한 고정 효과와 변량 효과를 동시에 고려함으로써, 전체 집단에서 나타나는 평균적인 용량-반응 관계뿐만 아니라, 특정 그룹(예: 교대 근무자)에서만 나타나는 특이한 패턴을 통계적으로 분리하여 분석할 수 있게 해준다. 예를 들어, '과로'라는 노출에 대한 '질병 반응'의 관계에서, 교대 근무 여부가 용량-반응 곡선의 기울기를 다르게 만든다는 것을 통계적으로 유의미하게 검증할 수 있다.

따라서, 혼합 효과 모델은 그 관계에 내재된 **'개인 및 그룹 간의 이질성'**을 과학적으로 분리하여 보다 정확하고 현실적인 결론을 도출하는 실습을 수행하게 된다.

4.6 본 강의는

본 강의는 직업병 역학연구의 핵심인 용량-반응 관계를 다루고, 인과성을 밝히는 과정에서 마주하는 역학적 문제를 고찰할 것이다. 또한, 단순한 선형 관계를 넘어서, 생물학적 현실을 반영하는 비선형 모델의 필요성을 알아보고, 개인과 그룹 간의 이질성을 고려하는 혼합 효과 모델을 통해 역학적 사고를 함양하는 데 목표를 둔다.

5 용량반은 관계 기본 실습

5.1 Exposure (Dose) and Health

```
if(!require(tidyverse)) install.packages("tidyverse")
if(!require(ggplot2)) install.packages("ggplot2")
if(!require(knitr)) install.packages("knitr")
if(!require(kableExtra)) install.packages("kableExtra")
```

우리몸에 필수 요소가 있다고 상상해 봅시다. 예를 들어 적혈구 백혈구를 생각해 보는 것입니다. 적혈구가 너무 적으면 빈혈과 같은 질병이 있는 것이고, 너무 많으면 적혈구 과다증이 있어 건강에 해롭습니다. 어떤 필수 요소가 너무 적거나 많은 상태를 질병으로, 적절한 양이 있는 경우 건강상태로 보는 것입니다. 다음과 같은 상황을 상상해 보겠습니다.

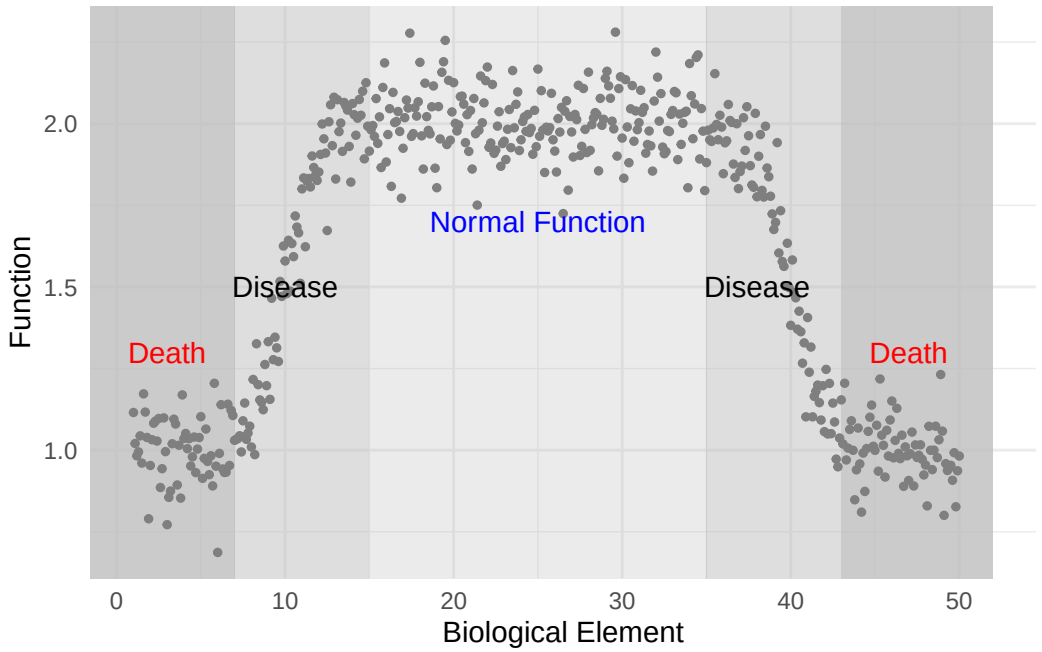
자료 생성

```
trace.e <- seq(1,50, by=0.1)
#normal range = 15~35
trace.e.h=function (x) {
  ifelse(x<20, 1/(1+exp(-x+10)),
         ifelse(x<30, rnorm(1, 1/(1+exp(-19)), 0.01),
                1/(1+exp(-19))-1/(1+exp(40-x))
                )
         )
  }
hstatus<-trace.e.h(trace.e)+rnorm(length(trace.e), 1, 0.1)
basic = tibble(trace.e, hstatus)
```

그림 그리기

이러한 관계를 그림으로 그려 보겠습니다. 기능적 측면에서 생물학적 필수 요소가 너무 적거나 너무 많으면 사망하거나 질병이 있는 상태로, 절적 수준이 유지되는 것을 정상상태로 볼 수 있습니다.

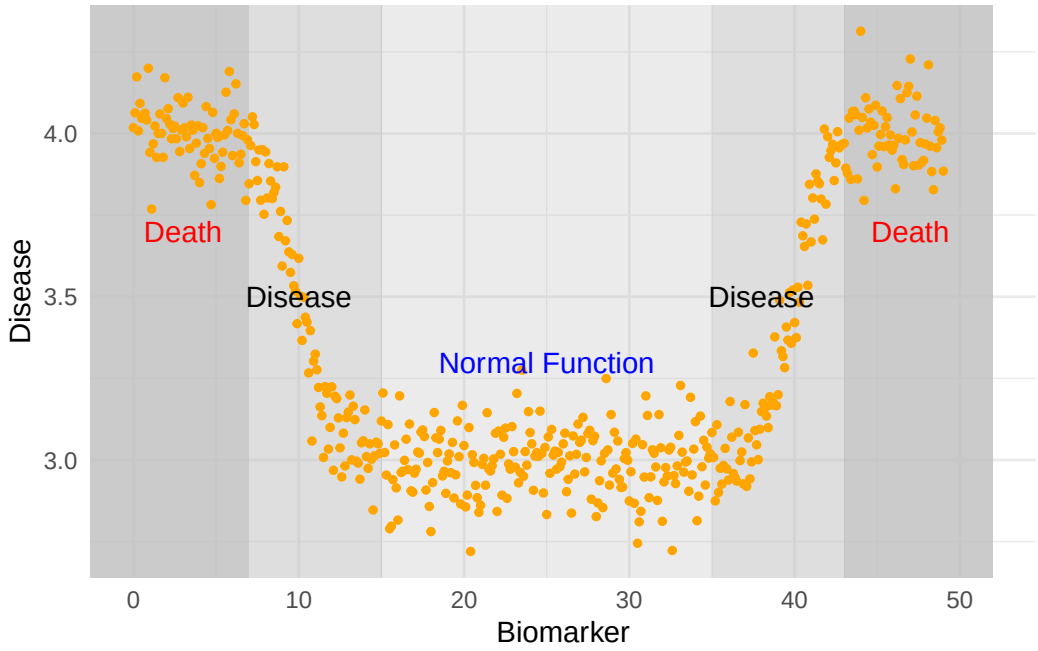
```r
basic %>%
  ggplot(aes(x= trace.e, y = hstatus)) +
  scale_x_continuous(name="Biological Element") +
  scale_y_continuous(name="Function") +
  theme_minimal()+
  geom_rect(data=basic[1,],aes(xmin=-Inf, xmax=7 , ymin=-Inf, ymax=Inf), fill= 'grey', alpha=0.8)
  geom_rect(data=basic[1,],aes(xmin=7,    xmax=15, ymin=-Inf, ymax=Inf), fill= 'grey', alpha=0.5)
  geom_rect(data=basic[1,],aes(xmin=15,   xmax=35, ymin=-Inf, ymax=Inf), fill= 'grey', alpha=0.3)
  geom_rect(data=basic[1,],aes(xmin=35,   xmax=43, ymin=-Inf, ymax=Inf), fill= 'grey', alpha=0.5)
  geom_rect(data=basic[1,],aes(xmin=43,   xmax=52, ymin=-Inf, ymax=Inf), fill= 'grey', alpha=0.8)
  geom_point(size=1, color = 'grey50') +
  annotate(geom="text", x=c(3,47),  y=c(1.3, 1.3), label="Death",    color="red") +
  annotate(geom="text", x=c(10,38), y=c(1.5, 1.5), label="Disease",  color="black") +
  annotate(geom="text", x=25,       y=1.7,         label="Normal Function",color="blue")
```



위의 그림을 Y축을 적혈구의 기능 측면에서 본 것으로 상상해보면 이해가 갑니다. 다음에는 적혈구의 가능을 악화시키는 물질에 노출되었다고 상상하고 기능이 아닌 질병 측면에서 볼 수 있습니다. 거꾸로 그래프를 뒤집을 수 있습니다.

5.2 실습 1: 질병 그림 그리기

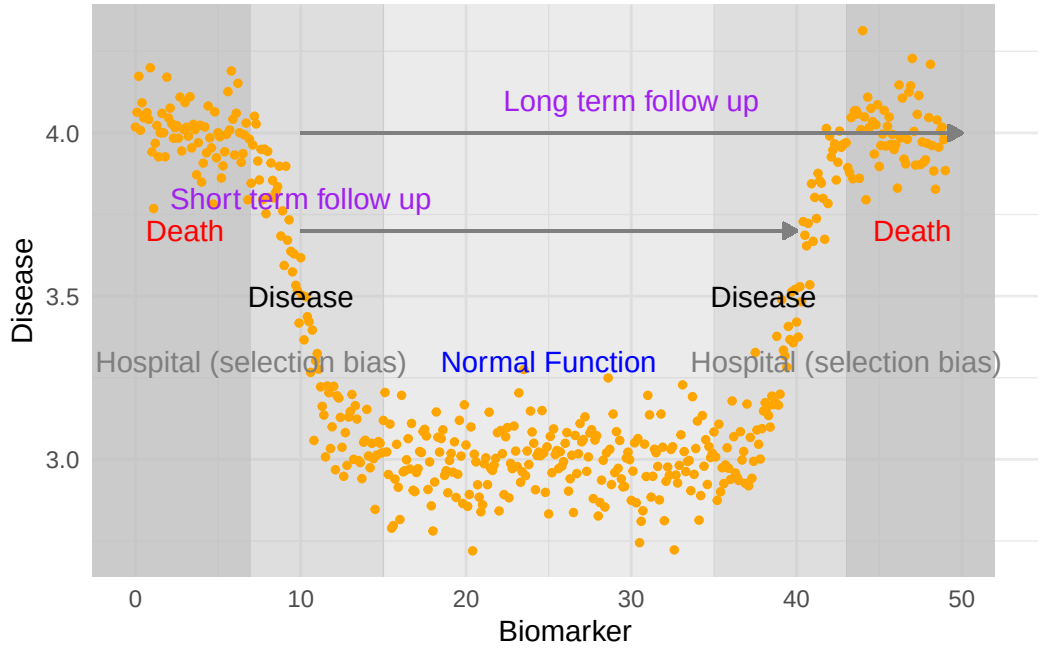
```
basic = basic %>%
  mutate(disease = -1*hstatus+5,
         exp.b   = -1*trace.e +50)

fig1 = basic  %>%
  ggplot(aes(x= exp.b, y = disease))+
  theme_minimal()+
  scale_x_continuous(name="Biomarker") +
  scale_y_continuous(name="Disease") +
  geom_rect(data=basic[1,],aes(xmin=-Inf, xmax=7 , ymin=-Inf, ymax=Inf), fill= 'grey', alpha=0.8)
  geom_rect(data=basic[1,],aes(xmin=7,    xmax=15, ymin=-Inf, ymax=Inf), fill= 'grey', alpha=0.5)
  geom_rect(data=basic[1,],aes(xmin=15,   xmax=35, ymin=-Inf, ymax=Inf), fill= 'grey', alpha=0.3)
  geom_rect(data=basic[1,],aes(xmin=35,   xmax=43, ymin=-Inf, ymax=Inf), fill= 'grey', alpha=0.5)
  geom_rect(data=basic[1,],aes(xmin=43,   xmax=52, ymin=-Inf, ymax=Inf), fill= 'grey', alpha=0.8)
  geom_point(size=1, color = 'orange') +
  annotate(geom="text", x=c(3,47),  y=c(3.7, 3.7), label="Death",    color="red") +
  annotate(geom="text", x=c(10,38), y=c(3.5, 3.5), label="Disease",  color="black") +
  annotate(geom="text", x=25,       y=3.3,         label="Normal Function",color="blue")
fig1
```



5.3 지역사회 연구 (community base cohort study)

지역사회 연구에서는 질병이 있는 사람 또는 기능이 약화된 사람은 병원에 입원해 있는 등 사회생활이 어려우므로 참여하지 못할 수 있습니다. 따라서 장기간 추적 관찰을 하지 않는 경우 바이오마커와 질병간에 U-shap 을 보이게 됩니다. 장기 관찰을 하거나 충분한 관찰을 하면 질병이 새로 생기는 부분을 찾을 수 있으므로 J-shap으로 보일 수 도 있습니다. 우리가 과거력이 있는 사람 또는 적절한 방법으로 건강이 악화되어 있는 사람을 제외하는 경우 바이오마커와 질병의 선형적 관계를 관찰할 수 있는 경우 입니다.

```
fig1 +
  annotate(geom="text", x=c(7, 43),  y=c(3.3, 3.3), label="Hospital (selection bias)",  color="gr
  annotate(geom="text", x=c(10),  y=c(3.8), label="Short term follow up",  color="purple") +
  geom_segment(aes(x=10, xend=40, y=3.7, yend=3.7), size = 0.5, color='grey50',
               arrow = arrow(length = unit(0.2, "cm"), type = "closed")) +
  annotate(geom="text", x=c(30),  y=c(4.1), label="Long term follow up",  color="purple") +
  geom_segment(aes(x=10, xend=50, y=4, yend=4), size = 0.5, color='grey50',
               arrow = arrow(length = unit(0.2, "cm"), type = "closed"))
```



산업보건에서는 건강한 근로자가 직장을 갖고 직장을 갖은 후에 일에 따라 물리화학적 인자에 노출이 됩니다. 만약 위의 그림에서 바이오마커가 일을하면서 노출되는 유해인자와 관련이 있다면, 사업장에서는 바이오마커가 매우 낮은 사람은 없을 것입니다. 그리고 유해인자에 노출이 많이 되어 질병이 생기고 병원에 가게된다면, 사업장을 중심으로 연구하는 경우 연구대상에 참여하지 못하게 됩니다. 즉 위의 그림에서 short-term follow up 의 상황이 발생하게 됩니다. 그런데 장기간

dose.e	resp
1	0.0179862
2	0.0474259
3	0.1192029
4	0.2689414
5	0.5000000
6	0.7310586
7	0.8807971
8	0.9525741
9	0.9820138
10	0.9933071

관찰하고 퇴사후의 자료도 이용한다면 long term follow up 과 같이 가게됩니다. 이때 상관 분석을 수행하면 short-term follow up에서는 U-shap으로, long term follow up 에서는 J-shap 으로 나타나게 됩니다. 실제 연구에서도 비슷한 상황이 발생하기도 합니다. 이때 우리가 얻은 데이터가 무엇을 목적으로 어떠한 설계로 만들어 졌는지를 관찰하고 정말 질병이 생길만한 사람을 제외한 현장에서 연구를 수행하고 있는 것은 아닌지 고민해 보아야 합니다.

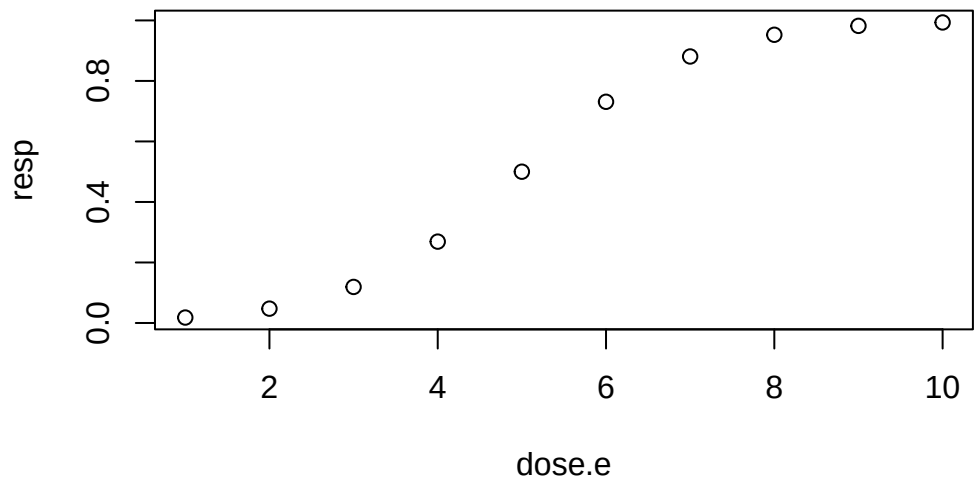
5.4 모형 차이: sigmoid curve vs linear regression

위의 그래프를 절반을 나누어서 적절한 기능을 하고 있는 사람만을 대상으로 장기간 추적관찰했다고 가정해 보겠습니다. 그러면 바이오마커와 질병의 관계를 선형으로 예측할 수도 있습니다. 노출이 지속되더라도 건강의 악화는 어느정도 포화될수 있으므로 (모두 다 계속 사망하지는 않으므로) 노출의 크기와 질병의 관계는 sigmoid curve 관계가 있을 수 있습니다.

어떤게 좋을까요? 정답은 없지만 LD50을 고려해서 생각해 보겠습니다. LD50이란 노출된 사람 중의 50%가 사망하는 농도를 의미합니다. 즉 LD50가 큰 물질은 적은 물질보다 많이 노출되어야 노출된 사람 중의 50%가 사망하므로 더 안전한 물질입니다.

```
sigmoid.f = function(x){
  1/(1+exp(5-x))
}

df = tibble(
dose.e = c(1:10),
resp   = sigmoid.f(dose.e)
)
df %>% kbl() %>%
  kable_paper("hover", full_width = F)
```

```
plot(df)
```



LD50은 설명을 했고, 과도한 비교를 위해서 LD70을 보고 이야기 해보겠습니다.

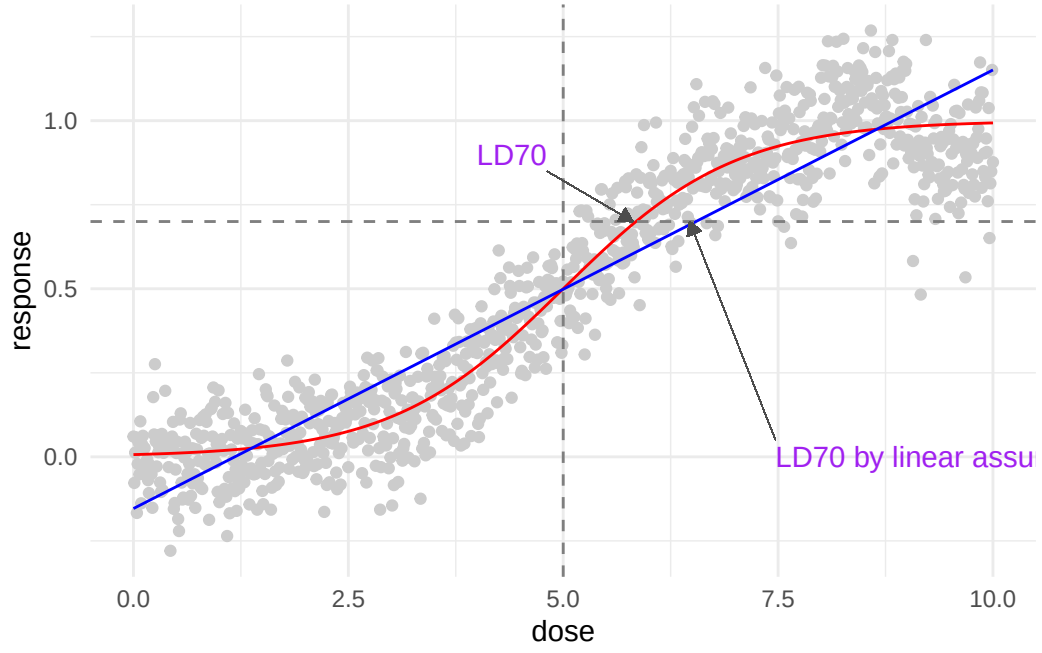
```
set.seed(50)
x<-seq(0, 10, 0.01)
y<-sigmoid.f(x)+rnorm(length(x), mean=0, sd=0.1)
pb<-c(rnorm(500, 0, 0.001), rnorm(300, 0, 0.01), rnorm(100, 0.1, 0.05),rnorm(101, -0.1, 0.05))
resp = y + pb
ld70.sm = x[which( sigmoid.f(x) >0.69 & sigmoid.f(x) < 0.71)] %>% min(.)
ld70.sm
```

```
[1] 5.81
```

```
mod1<-glm(resp ~ poly(x, 1))
pred1<-predict(mod1)
ld70.lm = x[which(pred1 >0.69 & pred1 <0.71)] %>% min(.)
ld70.lm
```

```
[1] 6.47
```

```r
df = tibble(dose = x, response= resp)
df %>%
  ggplot(aes(x=dose, y=response)) +
  geom_point(color = 'grey80') + theme_minimal()+
  geom_line(aes(y= sigmoid.f(dose)), color ='red') +
  geom_line(aes(y= predict(lm(response ~ poly(dose, 1)))), color ='blue') +
  geom_vline(xintercept =   5, linetype=2, color='grey50') +
  geom_hline(yintercept = 0.7, linetype=2, color='grey50') +
  annotate(geom="text", x=ld70.sm -1,  y=0.9, label="LD70",
           color="purple", hjust=1) +
  geom_segment(aes(x=ld70.sm -1, xend=ld70.sm, y=0.85, yend=0.7), size = 0.1, color='grey30',
               arrow = arrow(length = unit(0.2, "cm"), type = "closed")) +
  annotate(geom="text", x=ld70.lm +1,  y=0.0, label="LD70 by linear assumption",
           color="purple", hjust=0) +
  geom_segment(aes(x=ld70.lm +1, xend=ld70.lm, y=0.05, yend=0.7), size = 0.1, color='grey30',
               arrow = arrow(length = unit(0.2, "cm"), type = "closed"))
```

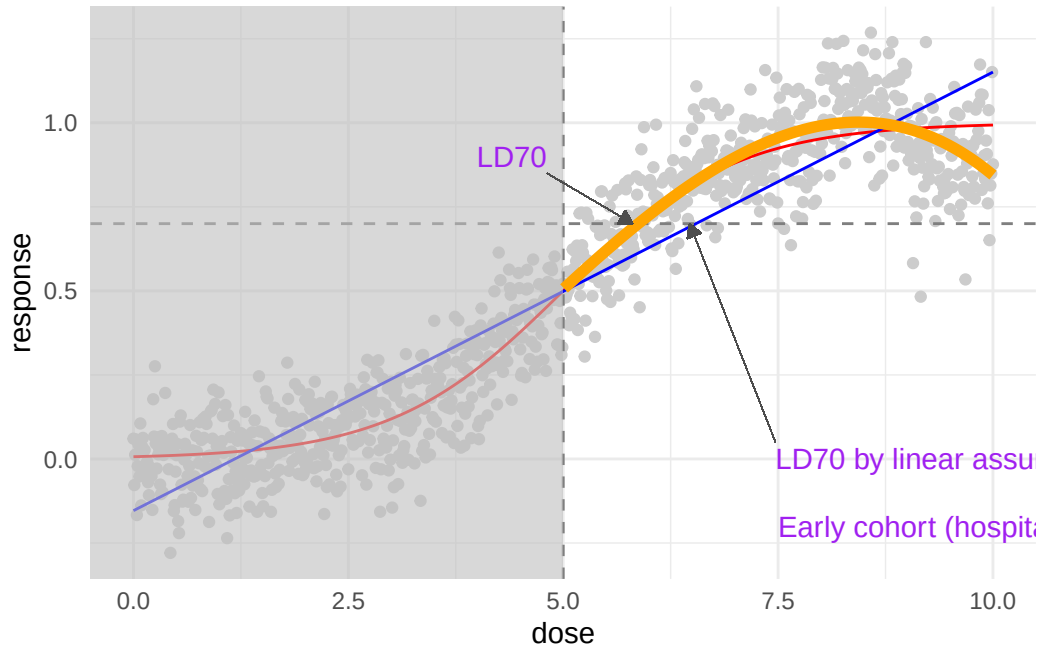


가상의 자료입니다. 그래도 중요하게 기억할 점은 선형과 비선형 커브로 할 경우 LD70이 달라지고, 독성에 대한 설명이 달라집니다. 따라서 어떠한 방식으로 설명할지 꼭 고민해야 합니다. 저 농도나 고농도에서는 더 큰 차이가 나타납니다. 그런데 대 부분 저농도의 노동자 들이나 고농도 노출의 노동자가 연구에 참여하기 어려운 상황이 발생합니다. 앞서 계속해서 이야기하는 건강근로자 효과 등을 상기 시켜 봅시다. 그럼 어떤 모델이 가장 적당할 까요? 우선 모델 적합도를 설명도로 비교해 볼 수 있습니다.

5.5 코호트 특성에 따른 상황

첫 수업 시간에 이야기한 것 처럼, 처음에는 질병이 생긴 노동자 위주로 연구가 진행되게 됩니다. 따라서 고농도 노출자 이면서 질병이 있는 사람으로 구성된 데이터에서는 상대적으로 높은 농도에서 질병이 발생하는 연구 결과과 발표 되기도 합니다. 그리고 그림에서 보듯이 선형관계를 고민하지 않는 다면 어디를 기준으로 해야할지 알 수 없는 상태입니다. 결론적으로 **위험하다** 는 알수 있지만, **얼마나 위험하다**는 아직 연구가 되지 않은 상태라는 것을 기억해야 합니다.

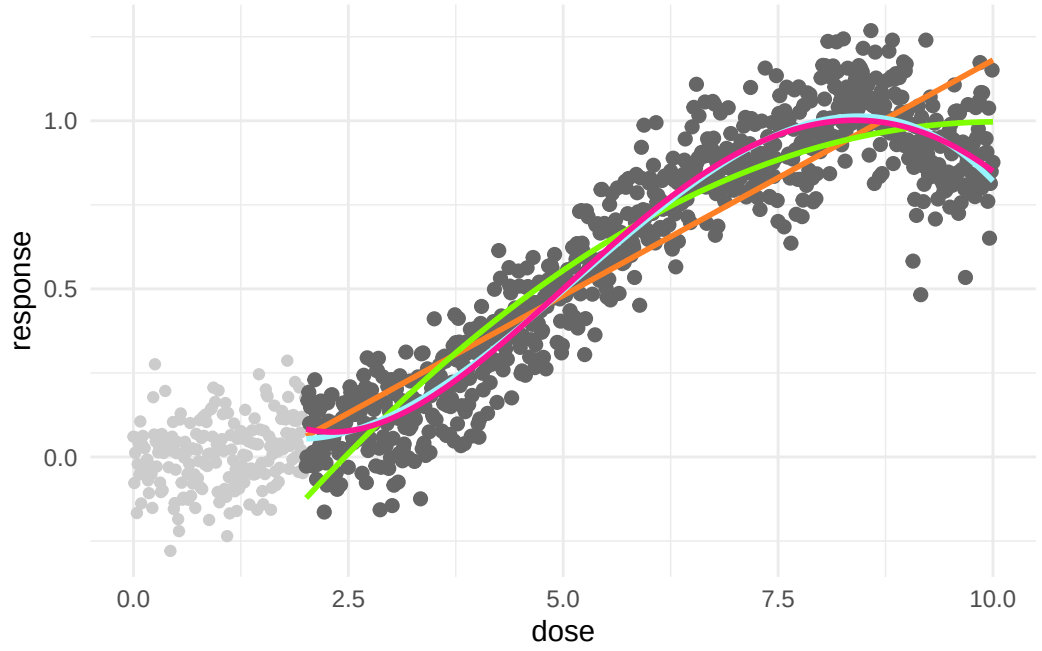
```
early_cohort = df %>% filter(dose > 5)

df %>%
  ggplot(aes(x=dose, y=response)) +
  geom_point(color = 'grey80') + theme_minimal()+
  geom_line(aes(y= sigmoid.f(dose)), color ='red') +
  geom_line(aes(y= predict(lm(response ~ dose))), color ='blue') +
  geom_vline(xintercept =   5, linetype=2, color='grey50') +
  geom_hline(yintercept = 0.7, linetype=2, color='grey50') +
  ## add 1
  annotate(geom="text", x=c(7.5),  y=c(-0.2),
           label="Early cohort (hospital base)",  color="purple", hjust=0) +
  geom_rect(data=df[1,],aes(xmin=-Inf,   xmax=5, ymin=-Inf, ymax=Inf),
            fill= 'grey', alpha=0.6) +
  geom_line(data= early_cohort,
              aes(y= predict(lm(response ~ poly(dose,3)))), color ='orange', size = 2) +
   annotate(geom="text", x=ld70.sm -1,  y=0.9, label="LD70",
           color="purple", hjust=1) +
  geom_segment(aes(x=ld70.sm -1, xend=ld70.sm, y=0.85, yend=0.7), size = 0.1, color='grey30',
               arrow = arrow(length = unit(0.2, "cm"), type = "closed")) +
  annotate(geom="text", x=ld70.lm +1,  y=0.0, label="LD70 by linear assumption",
           color="purple", hjust=0) +
  geom_segment(aes(x=ld70.lm +1, xend=ld70.lm, y=0.05, yend=0.7), size = 0.1, color='grey30',
               arrow = arrow(length = unit(0.2, "cm"), type = "closed"))
```



5.6 코호트 특성에 따른 상황과 이론적 모델

어느정도 잘 갖추어진 코호트를 생각해 보겠습니다. Dose 2 부터 노출된 사람을 대상으로 하였다고 가정하겠습니다.

```
# subcohort 2nd phase (1st phase is hospital base cohort)
s2c <- df %>% filter(dose >2)
s2m1<-lm(data=s2c, response ~ poly(dose, 1))
s2m2<-lm(data=s2c, response ~ poly(dose, 2))
s2m3<-lm(data=s2c, response ~ poly(dose, 3))
s2m4<-lm(data=s2c, response ~ poly(dose, 4))

df %>%
  ggplot(aes(x=dose, y=response)) +
  geom_point(color = 'grey80') + theme_minimal()+
  geom_point(data=s2c, color = 'grey40', size =2   ) +
  geom_line(data=s2c, aes(y=predict(s2m1)), color = 'chocolate1', size = 1) +
  geom_line(data=s2c, aes(y=predict(s2m2)), color = 'chartreuse1', size = 1) +
  geom_line(data=s2c, aes(y=predict(s2m3)), color = 'cadetblue1', size = 1) +
  geom_line(data=s2c, aes(y=predict(s2m4)), color = 'deeppink1', size = 1)
```



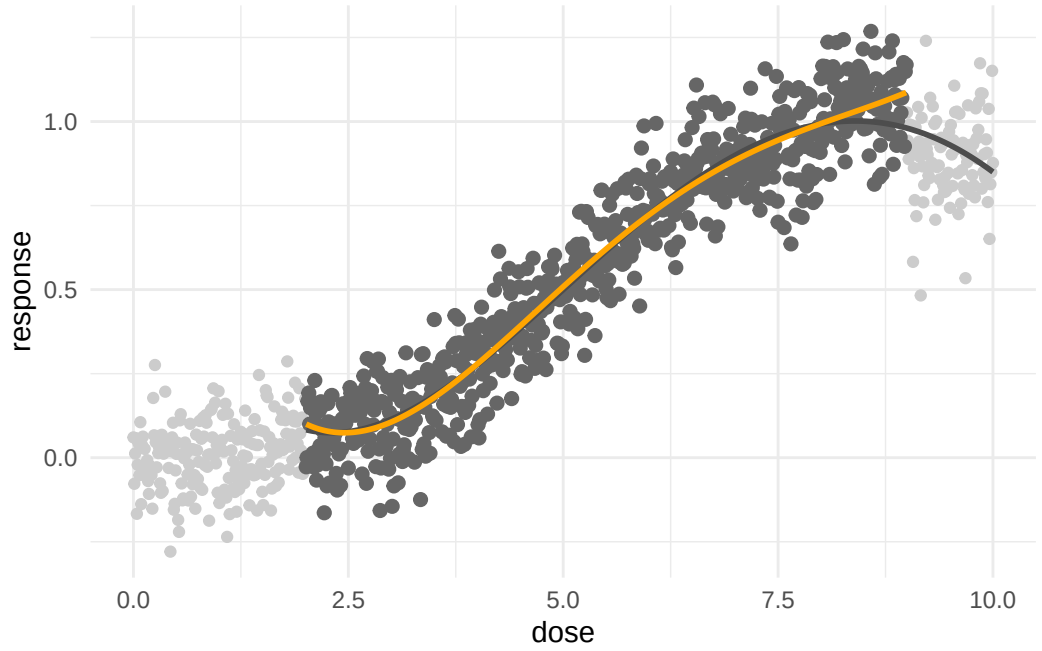
모델로 보면 선형 예측이 모형 적합도가 가장 낮다(low)고 나타나고 차수가 높을 수록 좋다고 나타나고 있습니다. 그런데, 어떤게 더 맞을 까요?.

```
anova(s2m1, s2m2, s2m3, s2m4)
```

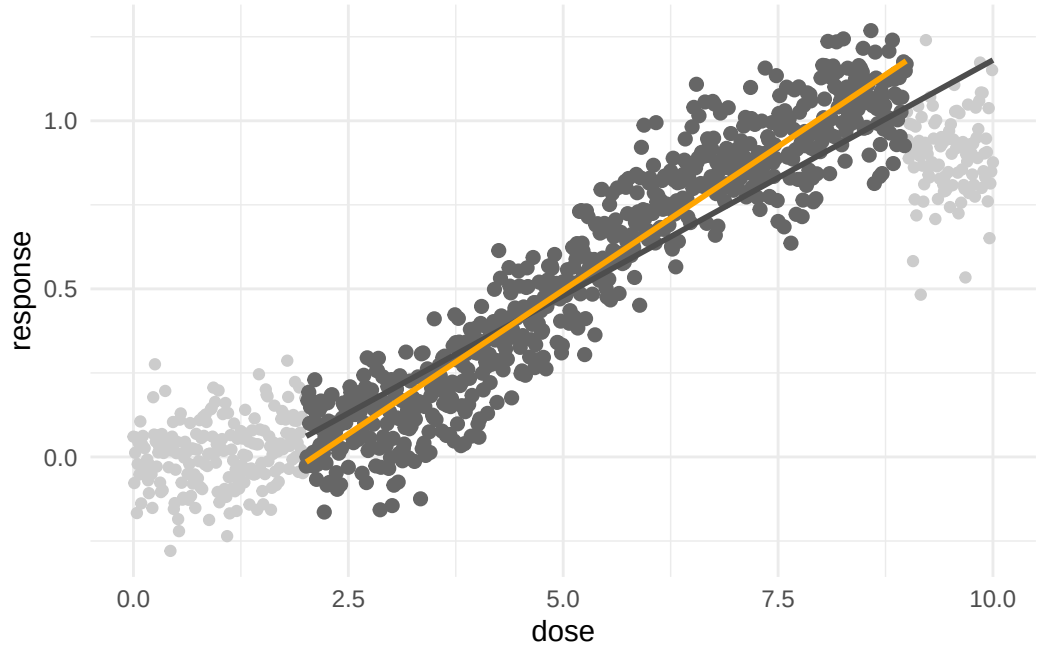
```
Analysis of Variance Table

Model 1: response ~ poly(dose, 1)
Model 2: response ~ poly(dose, 2)
Model 3: response ~ poly(dose, 3)
Model 4: response ~ poly(dose, 4)
  Res.Df     RSS Df Sum of Sq        F  Pr(>F)
1    798 18.9099
2    797 13.4734  1    5.4365 440.8574 < 2e-16 ***
3    796  9.8806  1    3.5928 291.3455 < 2e-16 ***
4    795  9.8036  1    0.0770   6.2423 0.01267 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

만약 농도가 높을 때 낮아지고 있는 부분을 고려한다면 어떻게 될까요? (실제로도 농도가 높은 곳에 근무하는 노동자는 만성 질병이 일어나기 전에 손상으로 사망하는 연구가 있습니다.) 따라서 그런 산업보건적 특성을 고려하면 어떻게 될 까요? 단순하게 9 이상 농도를 고려하지 않도록 하겠습니다.

```r
s3c <- df %>% filter(dose >2) %>% filter(dose <9)
s3m1<-lm(data=s3c, response ~ poly(dose, 1))
s3m2<-lm(data=s3c, response ~ poly(dose, 2))
s3m3<-lm(data=s3c, response ~ poly(dose, 3))
s3m4<-lm(data=s3c, response ~ poly(dose, 4))

df %>%
  ggplot(aes(x=dose, y=response)) +
  geom_point(color = 'grey80') + theme_minimal()+
  geom_point(data=s3c, color = 'grey40', size =2   ) +
  #geom_line(data=s2c, aes(y=predict(s2m1)), color = 'grey30', size = 1) +
  geom_line(data=s2c, aes(y=predict(s2m4)), color = 'grey30', size = 1) +
  #geom_line(data=s3c, aes(y=predict(s3m1)), color = 'orange', size = 1)
  geom_line(data=s3c, aes(y=predict(s3m4)), color = 'orange', size = 1)
```




```r
df %>%
  ggplot(aes(x=dose, y=response)) +
  geom_point(color = 'grey80') + theme_minimal()+
  geom_point(data=s3c, color = 'grey40', size =2   ) +
  geom_line(data=s2c, aes(y=predict(s2m1)), color = 'grey30', size = 1) +
  #geom_line(data=s2c, aes(y=predict(s2m4)), color = 'grey30', size = 1) +
  geom_line(data=s3c, aes(y=predict(s3m1)), color = 'orange', size = 1)
```




```
  #geom_line(data=s3c, aes(y=predict(s3m4)), color = 'orange', size = 1)
```

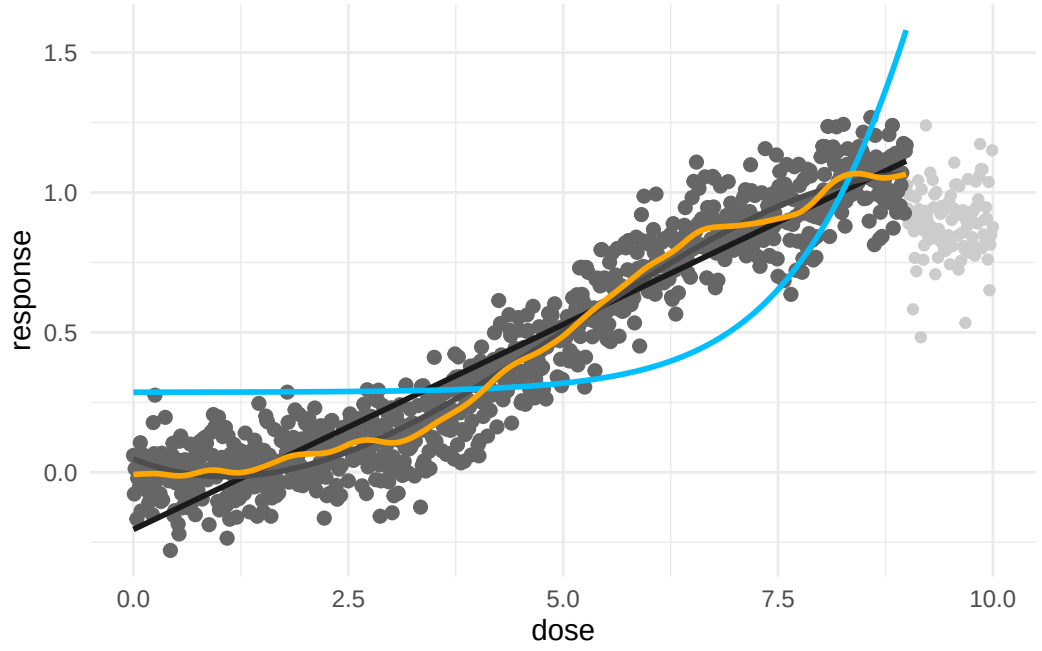
3차 이상의 모형에서는 큰 차이가 나지 않지만, 선형 모형에서는 차이가 상당합니다. 어떤 것이 더 맞다는 것은 아직 논할 단계는 아니고, 차이가 있다는 것을 기억하면 좋겠습니다. 그래서 실제 보고하고 적용할 때 현장에 더 적합한 것이 무엇인지, 목적이 보호 인지, 보상인지 등을 고려하여 해야 겠습니다.

```
library(gam)
######전체 자료 실습
s4c = df %>% filter(dose <9)
s4m1<-lm(data=s4c, response ~ poly(dose, 1))
s4m3<-lm(data=s4c, response ~ poly(dose, 3))
s4ms<-lm(data=s4c, response ~ sigmoid.f(dose-5))
s4mg<-gam(data=s4c, response ~ s(dose, 20))
anova(s4m1, s4m3, s4ms, s4mg)
```

```
Analysis of Variance Table

Model 1: response ~ poly(dose, 1)
Model 2: response ~ poly(dose, 3)
Model 3: response ~ sigmoid.f(dose - 5)
Model 4: response ~ s(dose, 20)
```

```
  Res.Df    RSS     Df Sum of Sq       F    Pr(>F)
1    898 14.876
2    896 10.064  2.000     4.812  233.61 < 2.2e-16 ***
3    898 67.272 -2.000   -57.208 2777.26 < 2.2e-16 ***
4    879  9.053 18.999    58.219  297.52 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
df %>%
  ggplot(aes(x=dose, y=response)) +
  geom_point(color = 'grey80') + theme_minimal()+
  geom_point(data=s4c, color = 'grey40', size =2   ) +
  geom_line(data=s4c, aes(y=predict(s4m1)), color = 'grey10', size = 1) +
  geom_line(data=s4c, aes(y=predict(s4m3)), color = 'grey30', size = 1) +
  geom_line(data=s4c, aes(y=predict(s4ms)), color = 'deepskyblue', size = 1) +
  geom_line(data=s4c, aes(y=predict(s4mg)), color = 'orange', size = 1)
```



5.6.1 Take home message

1. 고 노출 집단:

- 높은 표준 사망비(SMR)에도 불구하고 용량 증가에 따른 질병 발생 증가 경향이 명확하지 않을 수 있습니다. 즉, 선형적인 용량-반응 관계가 나타나지 않을 수 있습니다.

2. 중간 용량 노출 집단:

- 용량-반응 관계가 관찰되지만, 건강한 노동자 효과로 인해 실제 관계가 약화될 수 있습니다. 즉, 실제보다 용량과 질병 발생 간의 연관성이 낮게 나타날 수 있습니다.

3. 전체 용량 노출 집단:

- 용량-반응 관계를 분석하고, 건강한 노동자 효과를 통제하여 더욱 정확한 결과를 얻을 수 있습니다.
- 모델 선택 과정에서 LD50(반수 치사량)의 과대 또는 과소 평가가 발생할 수 있으므로 주의해야 합니다.
- 최적의 모델이라 하더라도, 건강한 노동자 효과로 인한 데이터 편향 때문에 실제 질병 발생 양상을 완벽하게 반영하지 못할 수 있습니다.

4. 권고 사항: 기존 모델에 의존하기보다는, 연구 특성에 맞는 자체적인 용량-반응 모델을 개발하여 적용하는 것이 바람직합니다.

5.7 Threshold 찾기 (change point 찾기)

threshold를 찾는 방법 중에 threshold point마다, piecewise regression을 반복해서 구하고, 최적의 모델을 찾는 방법을 사용할 수 있습니다. piecewise regression 의 간단한 설명은 다음과 같습니다

threshold points (piecewise regression)	codes
total	Resp = α + β1 · Dose + β2 ·(Dose - ☒) + + ☒0
If Dose 〈 ☒	Resp= α + β1 · Dose + ☒0
If Dose 〉 ☒	Resp= α - β2 ·☒ +(β1 + β2)· Dose + ☒0
model selection	minimal AIC value

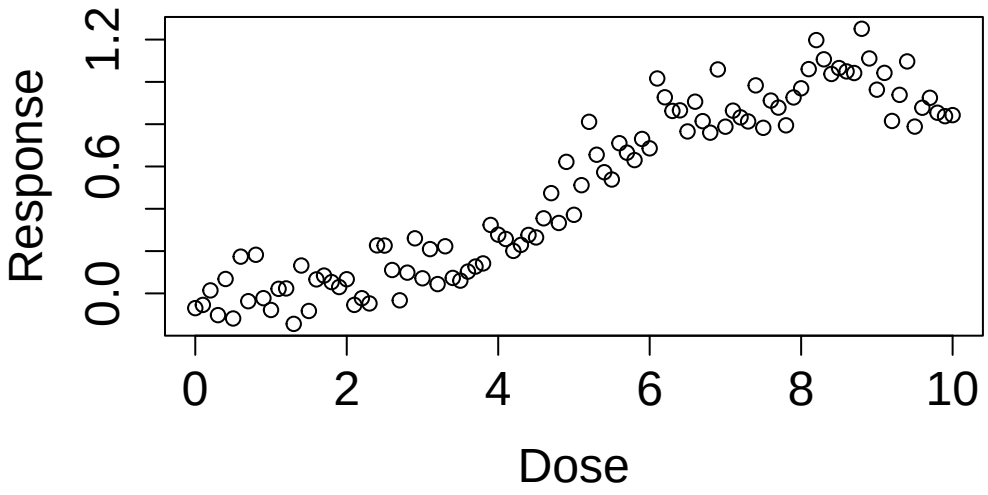
5.7.1 자료 생성

아래 처럼 임로 데이터를 생성해 보았습니다. 어떤 유해물질 노출 (Dose) 에 따라 건강영향 (Resp) 가 나타났다고 생각해 보겠습니다. 그리고 일정량에서 threshold가 있다고 생각해보겠습니다.

```
set.seed(0)
dose <- seq(0,10, 0.1)
length(dose)
```

```
[1] 101
```

```
pb<-c(rnorm(50, 0, 0.001), rnorm(30, 0, 0.01), rnorm(10, 0.1, 0.05),rnorm(11, -0.1, 0.05))
resp <-1/(1+exp(-(dose-5)))+rnorm(length(dose), 0, 0.1)+pb

plot(dose, resp, xlab='Dose', ylab='Response', cex.lab=1.5, cex.axis=1.5, cex.main=1.5, cex.sub=1
```




```
cohort<-data.frame(dose, resp, pb)
```

5.7.2 가상의 threshold 값 수행해보기

한 1에서 5 사이에 있어 보입니다. 이를 통해 예상되는 matrix (outdata)를 구해보았습니다. outdata의 행의 이름을 threshold point에 따라 intercept, beta for before threshold, and its p value, beta for post threshold and its pvalue와 그때은 AIC 값을 구해보겠습니다 . 우선 하나만 구해보겠습니다. therhold가 1일때와 5일때를 를 가정해 보겠습니다.

```
cpdose <- ifelse(dose -1 >0, dose -1, 0)
cpm <- glm(resp ~ dose + cpdose)
summary(cpm)$aic
```

```
[1] -92.20184
```

```
cpdose <- ifelse(dose -5 >0, dose -5, 0)
cpm <- glm(resp ~ dose + cpdose)
summary(cpm)$aic
```

```
[1] -86.9744
```

어떤 가정이 모델 적합도를 높이나요? 네 threshold가 1일 때 입니다. 그럼 2랑도, 2.5랑도 비교해 봐야겠지요. 이때 반복 분석을 수행해보도록 하겠습니다.

위의 모델을 함수로 만들었습니다

```
thr_fun <- function(thres){
cpdose <- ifelse(dose - thres >0, dose - thres, 0)
cpm    <- glm(resp ~ dose + cpdose)
aic    <- summary(cpm)$aic
data.frame(
  'threshold' = thres,
  'aic'       = aic)
}
```

이것을 돌릴 범위를 정해보겠습니다.

```
# 이게 어떤 의미 일까요?
dose[which(dose == 1):which(dose == 5)]
```

```
 [1] 1.0 1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.0 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8
[20] 2.9 3.0 3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.8 3.9 4.0 4.1 4.2 4.3 4.4 4.5 4.6 4.7
[39] 4.8 4.9 5.0
```

이제 반복해서 작업해 보겠습니다.

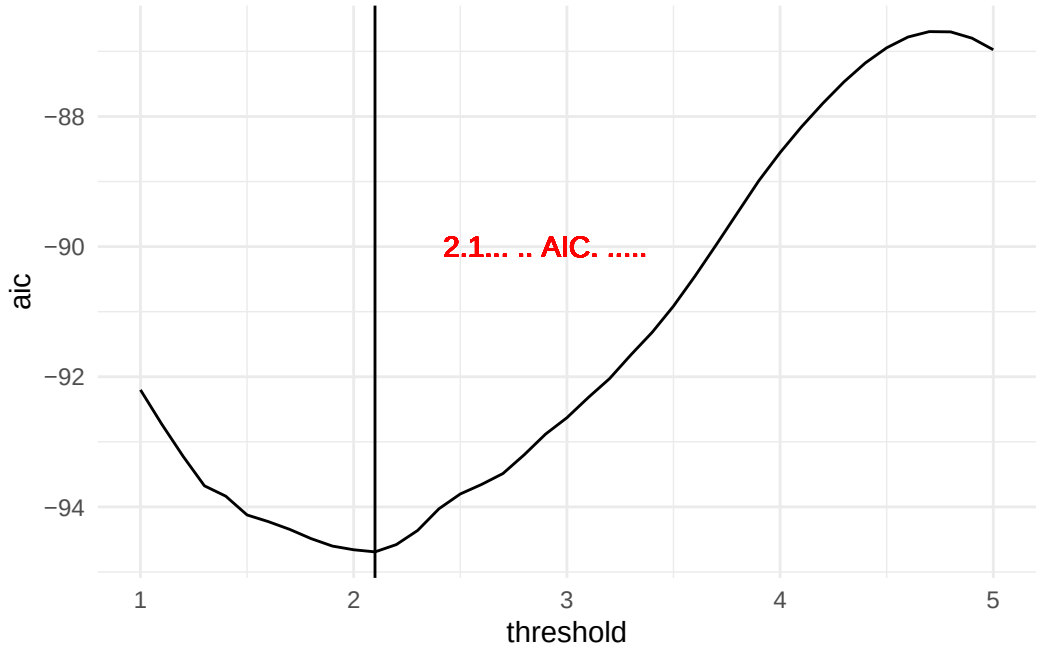
```
simul_list <- list()
simul_list <- lapply(dose[which(dose ==1):which(dose ==5)],                    thr_fun
                     )
```

이제 데이터 프레임으로 만들어 보겠습니다.

```
simul_dat <- do.call(rbind, simul_list)
```

그림을 그려보겠습니다.

```r
library(ggplot2)
opt.thres <- simul_dat$threshold[which.min(simul_dat$aic)]

simul_dat %>%
  ggplot(aes(x = threshold, y = aic)) +
  geom_line() +
  geom_vline(xintercept = opt.thres) +
  geom_text(x = opt.thres + 0.8, y = -90, color = 'red',
            label = paste0(round(opt.thres, 3), '점에서 최소 AIC를 보입니다.' )) +
  theme_minimal()
```

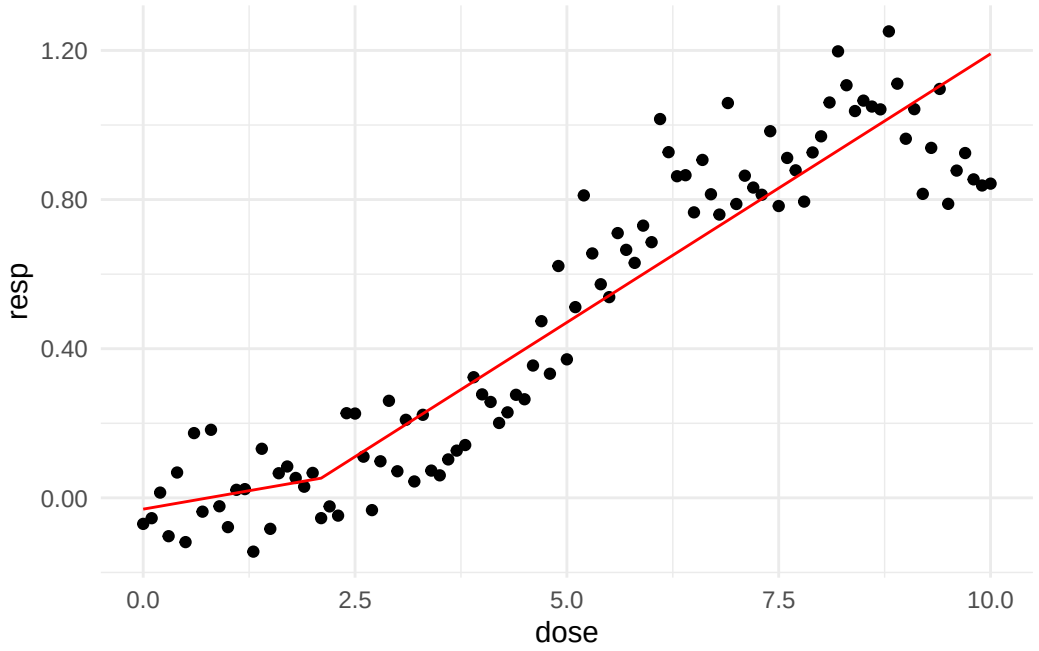


즉 2.1에서 threshold를 잡아 모델을 그리면 가장 적합함을 알 수 있습니다.

```r
thres = 2.1
f_cpdose <- ifelse(dose - thres >0, dose - thres, 0)
f_cpm    <- glm(resp ~ dose + f_cpdose)
```

```r
prepwlm <- predict(f_cpm)
scaleFUN <- function(x) sprintf("%.2f", x)
cohort %>%
  ggplot(aes(x= dose, y = resp)) +
  geom_point() +
```

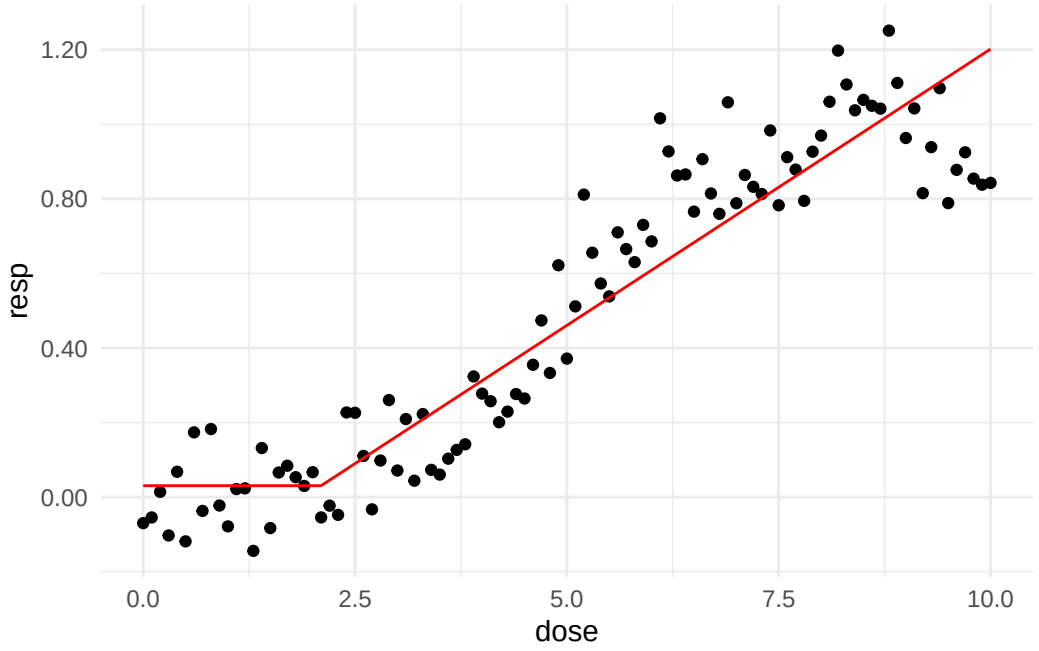
```
  theme_minimal() +
  scale_y_continuous(labels = scaleFUN) +
  geom_line(aes(y = prepwlm), color ='red')
```



만약 threshold 전에는 질병이 생기지 않는다고 가정하면 어떻게 될가요? dose 대신 predose 를 넣어 주면 됩니다.

```
thres = 2.1
f_cpdose <- ifelse(dose - thres >=0, dose - thres, 0)
f_predose <- ifelse(dose - thres <=0, 0, dose - thres )
f_cpm    <- glm(resp ~ f_predose + f_cpdose)
```

```
prepwlm <- predict(f_cpm)
scaleFUN <- function(x) sprintf("%.2f", x)
cohort %>%
  ggplot(aes(x= dose, y = resp)) +
  geom_point() +
  theme_minimal() +
  scale_y_continuous(labels = scaleFUN) +
  geom_line(aes(y = prepwlm), color ='red')
```


1.20
0.80
0.40
0.00
resp
0.0
2.5
5.0
7.5
10.0
dose

6 용량-반응 모델링 및 실습: Gompertz 모델

이 챕터에서는 비선형 모델인 **고퍼츠(Gompertz) 곡선**을 사용하여 용량-반응 관계를 분석하는 방법을 다룹니다. 우리는 `ggplot2`를 사용해 시각화를 하고, `nls.multstart`를 이용해 모델을 피팅하며, `investr`를 사용해 신뢰 구간을 계산할 것입니다.

6.1 Gompertz 모델의 이론적 배경

고퍼츠 곡선은 시간에 따른 성장 모델로 자주 사용되지만, 용량-반응 관계에서도 유용하게 활용될 수 있습니다. 이 곡선은 S자 형태를 가지며, 다음과 같은 세 가지 핵심 파라미터로 정의됩니다.

A: 곡선이 도달하는 **최대값(asymptote)**입니다. 이는 용량이 아무리 증가해도 반응이 더 이상 커지지 않고 수렴하는 지점입니다.

μ (Mu): 곡선의 기울기가 가장 가파른 지점에서의 **최대 변화율(maximum slope)**입니다. 이는 용량 변화에 따라 반응이 얼마나 빠르게 변하는지를 나타냅니다.

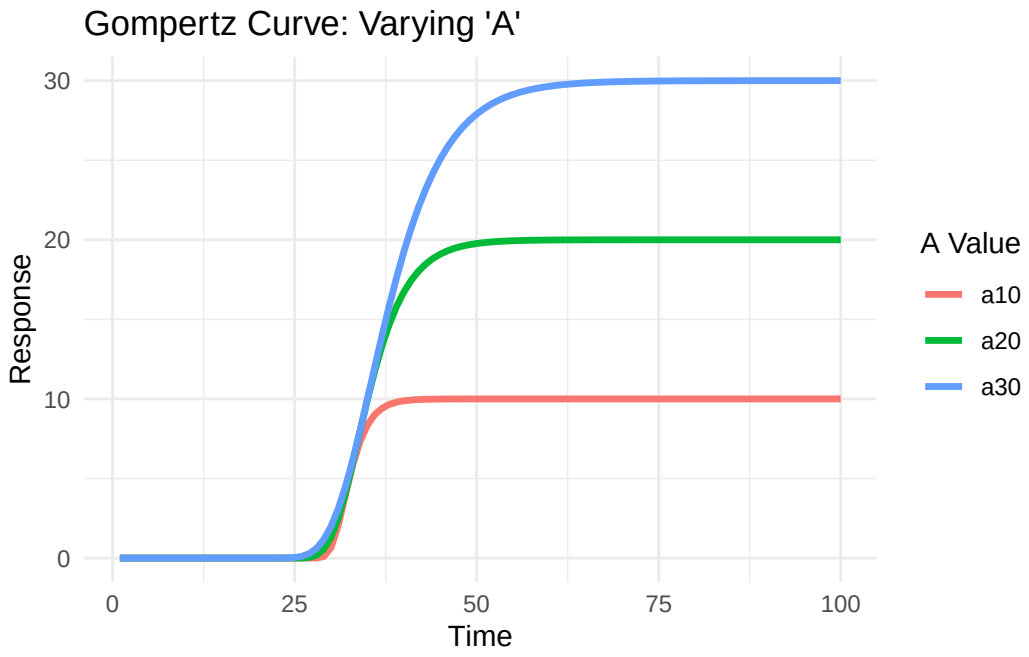
λ (Lambda): 반응이 시작되기 전 **지연되는 구간(lag-phase)**입니다. 즉, 반응이 의미 있는 수준으로 나타나기 시작하는 용량 지점입니다.

이 파라미터들이 어떻게 곡선 형태를 바꾸는지 시각적으로 확인해 봅시다.

```
# 필요한 패키지 로드
if(!require("tidyverse")) install.packages("tidyverse");library(tidyverse)
if(!require("nls.multstart")) install.packages("nls.multstart");library(nls.multstart)
if(!require("investr")) install.packages("investr");library(investr)
```

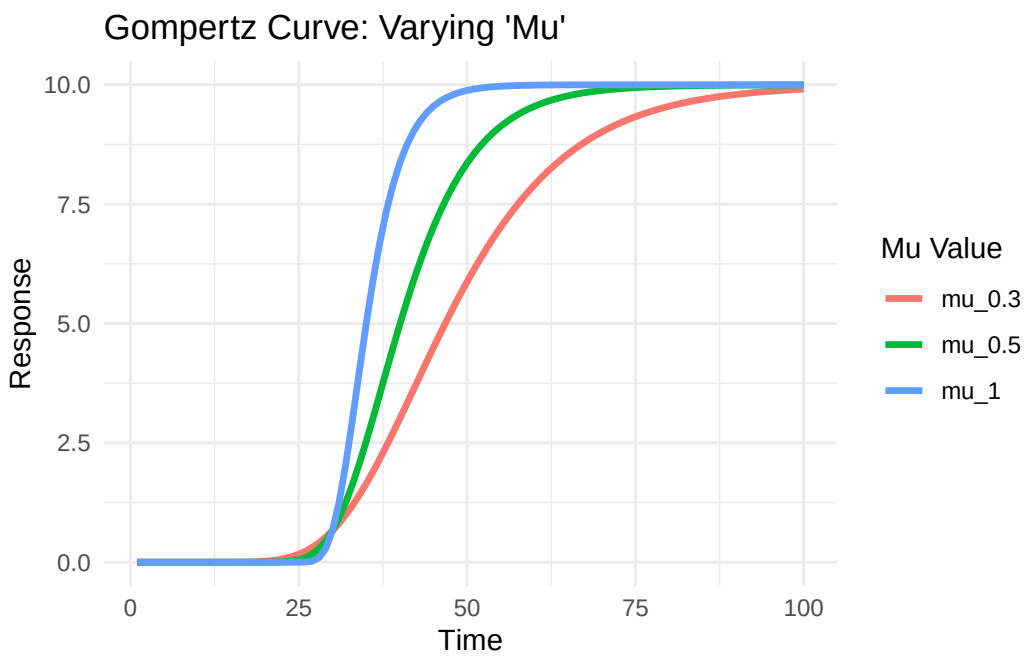
```
# Gompertz 함수 정의
gompertz <- function(time, a, mu, lambda) {
  a * exp(-exp(mu * exp(1) / a * (lambda - time) + 1))
}

# 파라미터별 Gompertz 곡선 시각화
# A 값 변화에 따른 곡선
tibble(time = 1:100) %>%
  mutate(
```

```
    a10 = gompertz(time, a = 10, mu = 2, lambda = 30),
    a20 = gompertz(time, a = 20, mu = 2, lambda = 30),
    a30 = gompertz(time, a = 30, mu = 2, lambda = 30)
  ) %>%
  pivot_longer(
    cols = starts_with("a"),
    names_to = "parameter_a",
    values_to = "response"
  ) %>%
  ggplot(aes(x = time, y = response, color = parameter_a)) +
  geom_line(size = 1.2) +
  labs(
    title = "Gompertz Curve: Varying 'A'",
    x = "Time",
    y = "Response",
    color = "A Value"
  ) +
  theme_minimal()
```



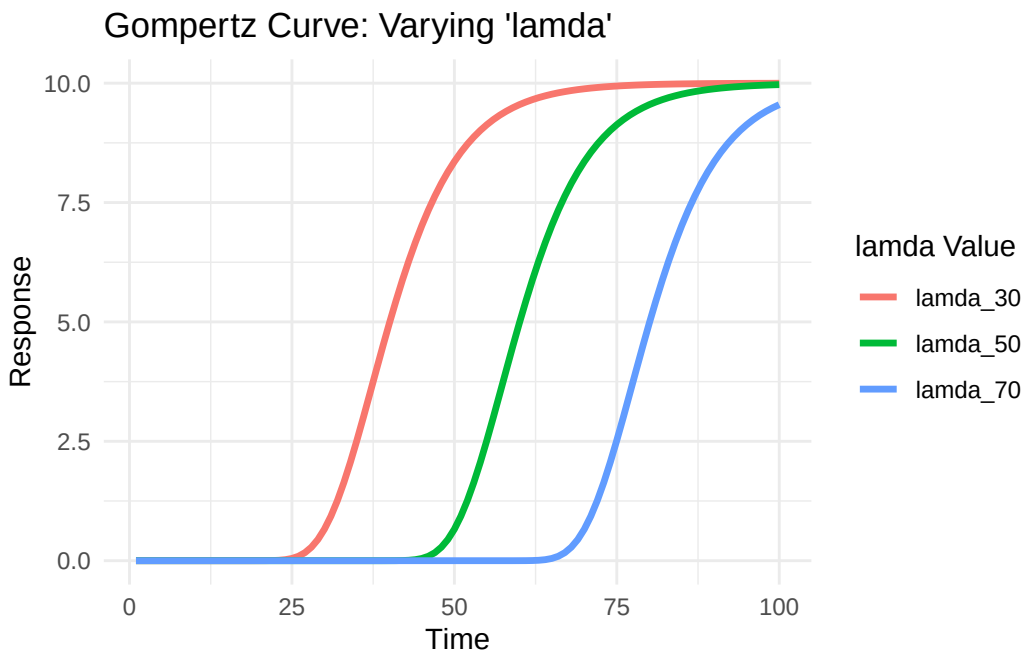

```
# 파라미터별 Gompertz 곡선 시각화
# Mu 값 변화에 따른 곡선
tibble(time = 1:100) %>%
```

```r
  mutate(
    mu_1 = gompertz(time, a = 10, mu = 1,   lambda = 30),
    mu_0.5 = gompertz(time, a = 10, mu = 0.5, lambda = 30),
    mu_0.3 = gompertz(time, a = 10, mu = 0.3, lambda = 30)
  ) %>%
  pivot_longer(
    cols = starts_with("m"),
    names_to = "parameter_a",
    values_to = "response"
  ) %>%
  ggplot(aes(x = time, y = response, color = parameter_a)) +
  geom_line(size = 1.2) +
  labs(
    title = "Gompertz Curve: Varying 'Mu'",
    x = "Time",
    y = "Response",
    color = "Mu Value"
  ) +
  theme_minimal()
```




```r
# 파라미터별 Gompertz 곡선 시각화
# lamda 값 변화에 따른 곡선
```

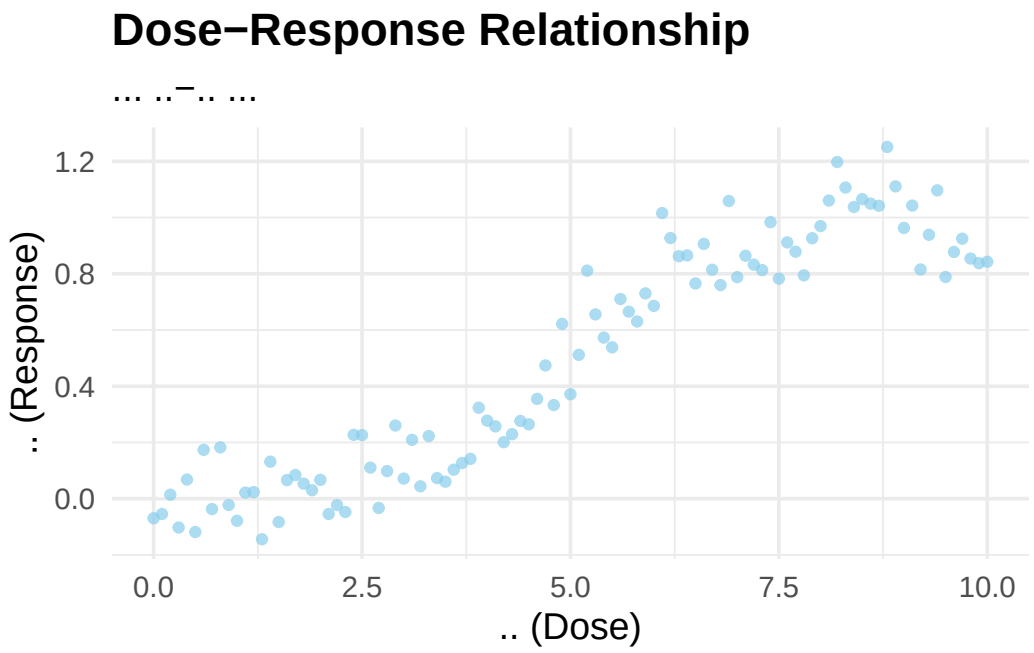
```r
tibble(time = 1:100) %>%
  mutate(
    lamda_30 = gompertz(time, a = 10, mu = 0.5,   lambda = 30),
    lamda_50 = gompertz(time, a = 10, mu = 0.5, lambda = 50),
    lamda_70 = gompertz(time, a = 10, mu = 0.5, lambda = 70)
  ) %>%
  pivot_longer(
    cols = starts_with("l"),
    names_to = "parameter_a",
    values_to = "response"
  ) %>%
  ggplot(aes(x = time, y = response, color = parameter_a)) +
  geom_line(size = 1.2) +
  labs(
    title = "Gompertz Curve: Varying 'lamda'",
    x = "Time",
    y = "Response",
    color = "lamda Value"
  ) +
  theme_minimal()
```



6.1.0.1 데이터 생성 및 기초 시각화

이제 실제 연구와 유사한 가상의 용량-반응 데이터를 생성하고 tibble로 변환하여 ggplot2로 시각화해 봅시다. 이 과정은 데이터의 전반적인 패턴을 탐색하고, 어떤 모델을 적용해야 할지 가늠하는 데 필수적입니다.

```
# 데이터 생성
set.seed(0)
dose <- seq(0, 10, 0.1)
pb <- c(rnorm(50, 0, 0.001), rnorm(30, 0, 0.01), rnorm(10, 0.1, 0.05), rnorm(11, -0.1, 0.05))
resp <- 1 / (1 + exp(-(dose - 5))) + rnorm(length(dose), 0, 0.1) + pb

# tibble 생성 및 시각화
cohort <- tibble(dose, resp, pb)

ggplot(cohort, aes(x = dose, y = resp)) +
  geom_point(color = "skyblue", alpha = 0.7) +
  labs(
    title = "Dose-Response Relationship",
    subtitle = "가상의 용량-반응 데이터",
    x = "용량 (Dose)",
    y = "반응 (Response)"
  ) +
  theme_minimal(base_size = 14) +
  theme(plot.title = element_text(face = "bold"))
```



6.1.0.2 Gompertz 모델 피팅 및 신뢰 구간 시각화

nls() 함수는 비선형 모델을 피팅하는 데 사용되지만, 초기값에 매우 민감합니다. (2019년 강의록) 이러한 문제를 해결하기 위해, 우리는 여러 초기값을 자동으로 시도하는 nls.multstart() 함수를 사용하여 더 안정적으로 최적의 모델을 찾을 것입니다. 모델이 피팅되면, investr 패키지를 이용해 예측 곡선과 함께 **95% 신뢰 구간(Confidence Interval)**을 시각화할 수 있습니다.

```
# nls_multstart를 사용하여 고퍼츠 모델 피팅
nls_fit <- nls_multstart(
  resp ~ gompertz(dose, a, mu, lambda),
  data = cohort,
  start_lower = c(a = 0, mu = 0, lambda = 0),
  start_upper = c(a = 2, mu = 1, lambda = 10),
  iter = 250
)
```

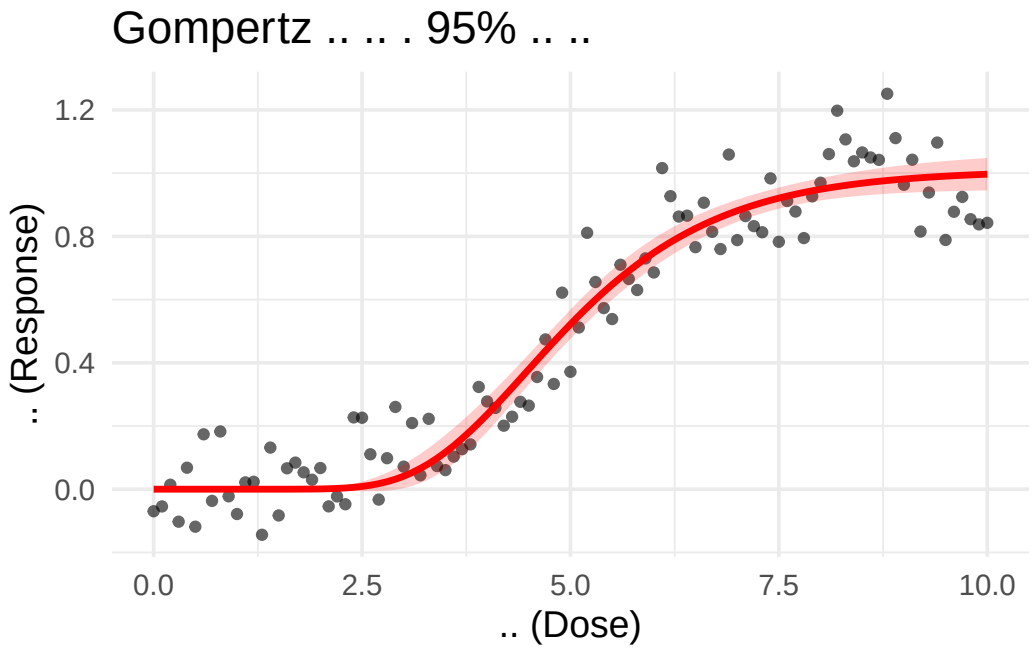
```
Error in nlsModel(formula, mf, start, wts) :
  singular gradient matrix at initial parameter estimates
Error in nlsModel(formula, mf, start, wts) :
  singular gradient matrix at initial parameter estimates
Error in nlsModel(formula, mf, start, wts) :
```

```
  singular gradient matrix at initial parameter estimates
Error in nlsModel(formula, mf, start, wts) :
  singular gradient matrix at initial parameter estimates
```

```r
# 모델 결과 요약
print(summary(nls_fit))
```

```
Formula: resp ~ gompertz(dose, a, mu, lambda)

Parameters:
       Estimate Std. Error t value Pr(>|t|)
a       1.00934    0.03107   32.48   <2e-16 ***
mu      0.29314    0.02754   10.64   <2e-16 ***
lambda  3.20596    0.15811   20.28   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1106 on 98 degrees of freedom

Number of iterations to convergence: 23
Achieved convergence tolerance: 1.49e-08
```

```r
# 모델 피팅 결과 시각화
# 신뢰 구간 계산
cohort_ci <- investr::predFit(
  nls_fit,
  newdata = cohort,
  interval = "confidence",
  level = 0.95
) %>%
  as_tibble() %>%
  mutate(dose = cohort$dose)

ggplot(cohort, aes(x = dose, y = resp)) +
  geom_point(alpha = 0.6) +
  geom_line(data = cohort_ci, aes(y = fit), color = "red", size = 1.2) +
  geom_ribbon(data = cohort_ci, aes(ymin = lwr, ymax = upr, y = fit), fill = "red", alpha = 0.2)
  labs(title = "Gompertz 모델 피팅 및 95% 신뢰 구간", x = "용량 (Dose)", y = "반응 (Response)") +
  theme_minimal(base_size = 14)
```



6.1.0.3 실습 3: 모델 비교 및 외삽(Extrapolation) 논의

고퍼츠 모델 외에 다항식 모델을 피팅하여 두 모델의 적합도를 비교해봅시다. **AIC(Akaike Information Criterion)**는 모델의 복잡성과 적합도를 함께 고려하는 지표로, AIC 값이 낮을수록 더 좋은 모델로 평가됩니다. AIC를 통해 어떤 모델이 데이터에 가장 잘 맞는지 통계적으로 평가할 수 있습니다.

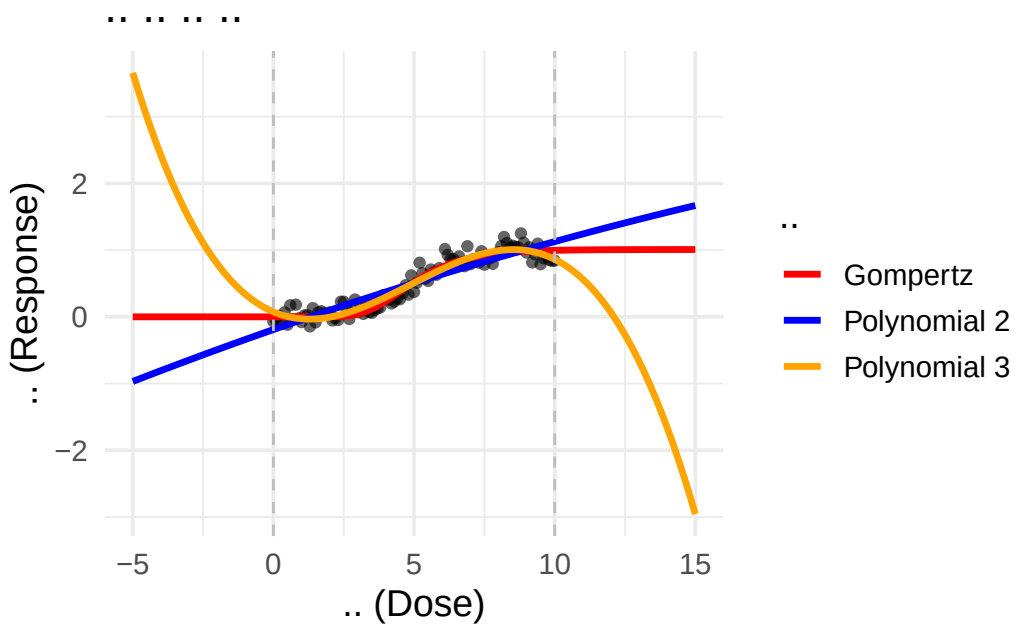
이 그래프를 보면, 모델의 외삽(Extrapolation) 결과가 어떻게 달라지는지 알 수 있습니다. 데이터가 없는 구간(dose -5 ~ 0 또는 10 ~ 15)에서 다항식 모델은 실제와 동떨어진 예측을 하는 반면, Gompertz 모델은 현실적인 포화 곡선을 유지합니다. 이는 모델 선택이 데이터 범위 밖의 위험을 평가하는 데 얼마나 중요한지를 시사합니다.

```
# 다항식 모델 피팅
poly2_fit <- glm(resp ~ poly(dose,2),data = cohort)
poly3_fit <- glm(resp ~ poly(dose,3),data = cohort)

# AIC를 이용한 모델 비교
AIC(nls_fit, poly2_fit, poly3_fit)
```

```
          df        AIC
nls_fit    4 -153.22741
poly2_fit  4  -87.32223
```

```
poly3_fit  5 -154.38409
```

```r
# 모델 외삽(extrapolation) 결과 시각화
extrapolated_data <- tibble(dose = seq(-5, 15, length.out = 100))

extrapolated_data <- extrapolated_data %>%
  mutate(
    gompertz_pred = predict(nls_fit, newdata = .),
    poly2_pred = predict(poly2_fit, newdata = .),
    poly3_pred = predict(poly3_fit, newdata = .)
  )

ggplot(cohort, aes(x = dose, y = resp)) +
  geom_point(alpha = 0.6) +
  geom_line(data = extrapolated_data, aes(y = gompertz_pred, color = "Gompertz"), size = 1.2) +
  geom_line(data = extrapolated_data, aes(y = poly2_pred, color = "Polynomial 2"), size = 1.2) +
  geom_line(data = extrapolated_data, aes(y = poly3_pred, color = "Polynomial 3"), size = 1.2) +
  geom_vline(xintercept = range(cohort$dose), linetype = "dashed", color = "gray") +
  scale_color_manual(values = c("Gompertz" = "red", "Polynomial 2" = "blue", "Polynomial 3" = "or
  labs(title = "모델 외삽 예측 비교", x = "용량 (Dose)", y = "반응 (Response)", color = "모델") +
  theme_minimal(base_size = 14)
```



7 비선형 모델 비교과 MixedModel

```
if(!require("tidyverse")) install.packages("tidyverse");library(tidyverse)
if(!require("nls.multstart")) install.packages("nls.multstart");library(nls.multstart)
if(!require("investr")) install.packages("investr");library(investr)
if(!require("nlme")) install.packages("nlme");library(nlme)
if(!require("boot")) install.packages("boot");library(boot)
```

이 챕터에서는 앞서 배운 기본적인 고퍼츠(Gompertz) 모델을 넘어, 실제 연구에서 마주하는 복잡한 상황을 다루는 방법을 알아봅니다. 특히, 여러 비선형 모델을 비교하고, 교대 근무와 같은 중요한 공변량이 용량-반응 관계에 미치는 영향을 어떻게 분석하는지 배울 것입니다.

7.1 여러 비선형 모델 비교 및 선택

용량-반응 관계는 언제나 S자형 곡선(Gompertz, Logistic)만 따르는 것은 아닙니다. 때로는 힐(Hill) 모델과 같이 민감도가 다른 곡선이 더 적합할 수 있습니다. 우리가 가진 데이터에 어떤 모델이 가장 잘 맞는지 평가하는 것은 매우 중요합니다.

7.1.1 모델 이론

- **로지스틱 모델**: y(dose)=fraca1+exp(⊠b(dose⊠c))
  - S자형 곡선을 설명하는 가장 기본적인 모델입니다. 특히, 질병 발생 여부와 같은 이항형 결과를 다룰 때 많이 사용됩니다
  - a: 최대 반응값
  - b: 곡선의 기울기(성장률)
  - c: LD50(반응이 50%가 되는 용량)
- **힐 모델**: y(dose)=fracacdotdosebcb+doseb
  - 로지스틱 모델과 비슷하지만, 곡선의 기울기를 조절하는 파라미터가 추가되어, 용량-반응 관계의 민감도를 더 세밀하게 표현할 수 있습니다.

- b: 힐 계수(Hill Coefficient), 곡선 기울기(민감도)
- c: LD50(반응이 50%가 되는 용량)

7.1.2 실습: 모델 피팅 및 비교

세 가지 모델(Gompertz, Logistic, Hill)을 모두 데이터에 피팅하고, 통계적인 지표인 AIC(Akaike Information Criterion) 값을 비교하여 어떤 모델이 가장 적합한지 평가해 봅시다. AIC는 모델의 복잡성과 데이터에 대한 적합도를 동시에 고려하여, AIC 값이 가장 낮은 모델이 가장 좋은 모델입니다.

```
# Gompertz 모델 함수
gompertz <- function(dose, a, mu, lambda) {
  a * exp(-exp(mu * exp(1) / a * (lambda - dose) + 1))
}

# 로지스틱 모델 함수
logistic <- function(dose, a, b, c) {
  a / (1 + exp(-b * (dose - c)))
}

# 힐 모델 함수
hill <- function(dose, a, b, c) {
  (a * dose^b) / (c^b + dose^b)
}

# LD50 계산 함수
ld50_logistic <- function(model) {
  coefs <- coef(model)
  ld50 <- coefs["c"]
  return(ld50)
}

# 데이터 생성
set.seed(0)
dose <- seq(0, 10, 0.1)
pb <- c(rnorm(50, 0, 0.001), rnorm(30, 0, 0.01), rnorm(10, 0.1, 0.05), rnorm(11, -0.1, 0.05))
resp <- 1 / (1 + exp(-(dose - 5))) + rnorm(length(dose), 0, 0.1) + pb
cohort <- tibble(dose, resp)

# 여러 모델 피팅
nls_gompertz <- nls_multstart(
```

```
  resp ~ gompertz(dose, a, mu, lambda),
  data = cohort,
  start_lower = c(a = 0, mu = 0, lambda = 0),
  start_upper = c(a = 2, mu = 1, lambda = 10),
  iter = 250 # iter 인자 추가
)
```

```
Error in nlsModel(formula, mf, start, wts) :
  singular gradient matrix at initial parameter estimates
Error in nlsModel(formula, mf, start, wts) :
  singular gradient matrix at initial parameter estimates
Error in nlsModel(formula, mf, start, wts) :
  singular gradient matrix at initial parameter estimates
Error in nlsModel(formula, mf, start, wts) :
  singular gradient matrix at initial parameter estimates
```

```
nls_logistic <- nls_multstart(
  resp ~ logistic(dose, a, b, c),
  data = cohort,
  start_lower = c(a = 0, b = 0, c = 0),
  start_upper = c(a = 2, b = 1, c = 10),
  iter = 250 # iter 인자 추가
)
```

```
Error in nlsModel(formula, mf, start, wts) :
  singular gradient matrix at initial parameter estimates
Error in nlsModel(formula, mf, start, wts) :
  singular gradient matrix at initial parameter estimates
Error in nlsModel(formula, mf, start, wts) :
  singular gradient matrix at initial parameter estimates
Error in nlsModel(formula, mf, start, wts) :
  singular gradient matrix at initial parameter estimates
Error in nlsModel(formula, mf, start, wts) :
  singular gradient matrix at initial parameter estimates
Error in nlsModel(formula, mf, start, wts) :
  singular gradient matrix at initial parameter estimates
Error in nlsModel(formula, mf, start, wts) :
  singular gradient matrix at initial parameter estimates
Error in nlsModel(formula, mf, start, wts) :
  singular gradient matrix at initial parameter estimates
Error in nlsModel(formula, mf, start, wts) :
```

```
  singular gradient matrix at initial parameter estimates
Error in nlsModel(formula, mf, start, wts) :
  singular gradient matrix at initial parameter estimates
```

```r
nls_hill <- nls_multstart(
  resp ~ hill(dose, a, b, c),
  data = cohort,
  start_lower = c(a = 0, b = 0, c = 0),
  start_upper = c(a = 2, b = 10, c = 10),
  iter = 250 # iter 인자 추가
)
```

```r
# AIC 값 비교
aic_values <- AIC(nls_gompertz, nls_logistic, nls_hill)
print(aic_values)
```

```
             df       AIC
nls_gompertz  4 -153.2274
nls_logistic  4 -160.6434
nls_hill      4 -156.0963
```

제공된 코드의 AIC 비교 결과를 보면 다음과 같습니다.

모델	AIC 값
nls_gompertz	-153.2274
nls_logistic	-160.6434
nls_hill	-156.0963

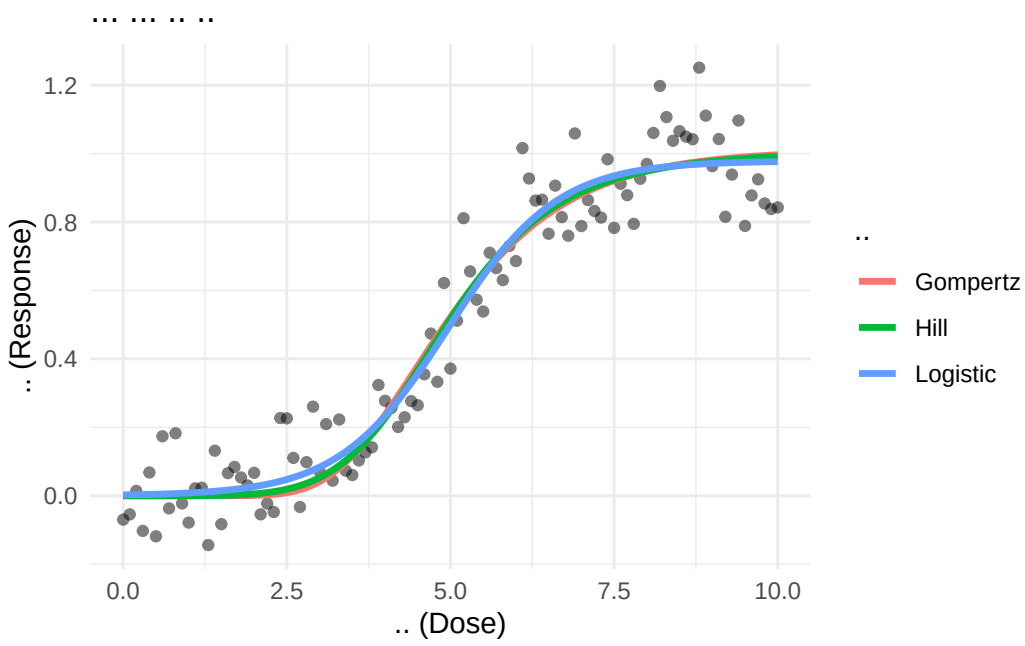
이 결과를 보면

Logistic 모델의 AIC가 -160.6434로 가장 낮습니다. 이는 우리가 생성한 데이터에 대해 로지스틱 모델이 다른 두 모델보다 통계적으로 더 우수한 적합도를 보인다는 것을 의미합니다.

이 결과를 그래프로 시각화해 보면, 세 개의 곡선이 데이터에 어떻게 피팅되는지 한눈에 확인할 수 있습니다.

```r
# 모든 모델을 한 그래프에 시각화
cohort_predicted <- cohort %>%
  mutate(
    Gompertz = predict(nls_gompertz),
    Logistic = predict(nls_logistic),
```

```r
    Hill = predict(nls_hill)
  ) %>%
  pivot_longer(
    cols = c(Gompertz, Logistic, Hill),
    names_to = "Model",
    values_to = "Predicted_Response"
  )

ggplot(cohort, aes(x = dose, y = resp)) +
  geom_point(alpha = 0.5) +
  geom_line(data = cohort_predicted, aes(y = Predicted_Response, color = Model), size = 1.2) +
  labs(
    title = "다양한 비선형 모델 비교",
    x = "용량 (Dose)",
    y = "반응 (Response)",
    color = "모델"
  ) +
  theme_minimal()
```



7.2 실제 데이터의 변수 고려: 공변량(Covariates) 분석

이제 한 걸음 더 나아가, 교대 근무와 같은 공변량이 용량-반응 관계에 어떤 영향을 미치는지 분석해 봅시다. 실제 직업 환경에서는 개인의 근무 형태나 생활 습관 등이 유해물질 노출에 대한 질병 반응에 영향을 줄 수 있습니다. 우리는 **교대 근무** 여부에 따라 과로와 질병 반응 간의 관계가 달라지는 상황을 시뮬레이션했습니다.

7.2.1 시뮬레이션: 공변량 효과

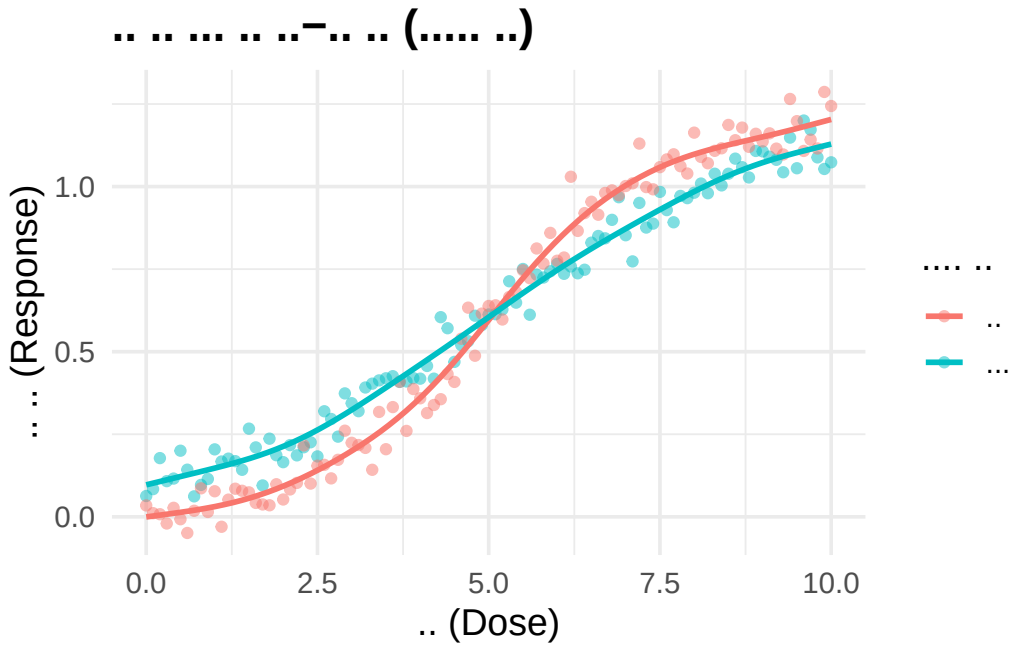
우리는 두 그룹의 가상 데이터를 만들었습니다.

- **비교대 근무자**: 과로가 증가할수록 질병 반응이 **선형적**으로 증가하는 그룹입니다.
- **교대 근무자**: 과로가 증가할수록 질병 반응이 **고퍼츠 곡선**을 따라 증가하는 그룹입니다. 즉, 특정 시점부터 질병 반응이 급격하게 증가하는 패턴을 보입니다.

혼합 효과 모델은 이 두 그룹의 관계를 하나의 통합된 모델로 분석하는 도구입니다.

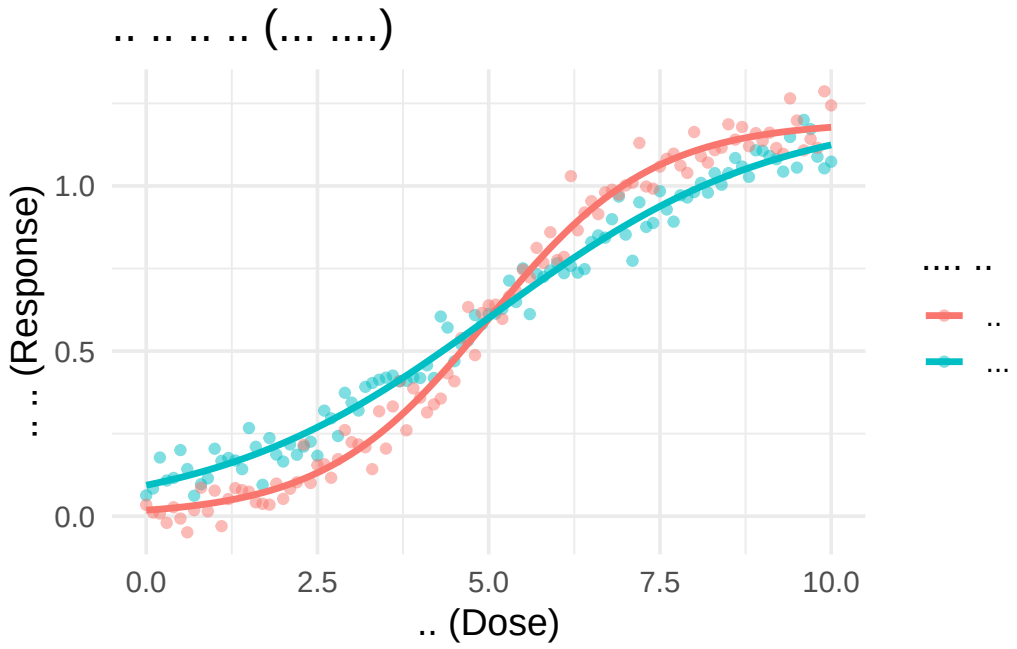
```
# nlme 패키지 로드
library(nlme)
library(tidyverse)

# Gompertz 모델 함수
gompertz <- function(dose, a, mu, lambda) {
  a / (1 + exp(-mu * (dose - lambda)))
}

# 데이터 생성
set.seed(123)
dose <- seq(0, 10, 0.1)
n_dose <- length(dose)

# 비 교대 근무자 그룹의 반응 (느린 Gompertz 곡선)
resp_nonshift <- gompertz(dose, a = 1.2, mu = 0.5, lambda = 5) + rnorm(n_dose, 0, 0.05)

# 교대 근무자 그룹의 반응 (더 가파른 Gompertz 곡선)
resp_shift <- gompertz(dose, a = 1.2, mu = 0.8, lambda = 5) + rnorm(n_dose, 0, 0.05)

# 두 그룹의 데이터를 하나의 데이터프레임으로 결합
cohort_full <- tibble(
  dose = rep(dose, 2),
  shift = factor(c(rep("비교대", n_dose), rep("교대", n_dose))),
```

```
  resp = c(resp_nonshift, resp_shift)
)
# 데이터 시각화
ggplot(cohort_full, aes(x = dose, y = resp, color = shift)) +
  geom_point(alpha = 0.5) +
  geom_smooth(method = "gam", formula = y ~ s(x), se = FALSE) +
  labs(
    title = "교대 근무 여부에 따른 용량-반응 관계 (시뮬레이션 탐색)",
    x = "과로 (Dose)",
    y = "질병 반응 (Response)",
    color = "교대근무 여부"
  ) +
  theme_minimal(base_size = 14) +
  theme(plot.title = element_text(face = "bold"))
```




```
# 혼합 효과 모델 피팅
# 두 그룹 모두 Gompertz 곡선을 따르는 가정을 반영합니다.
# fixed = a + mu + lam ~ shift는 shift 그룹에 따라 파라미터 a, mu, lam이 달라지도록 설정합니다.
nlme_fit <- nlme(
  model = resp ~ a / (1 + exp(-(mu * (dose - lam)))),
  data = cohort_full,
  fixed = a + mu + lam ~ shift,
```

```r
  random = a + mu + lam ~ 1 | shift,
  start = c(a = c(1.2, 1.2), mu = c(0.5, 0.8), lam = c(5, 5)),
  control = list(pnlsTol = 0.01)
)
# 모델 결과 요약
#summary(nlme_fit)

# 모델 예측값 계산
cohort_full$nlme_pred <- predict(nlme_fit)

# 혼합 효과 모델 시각화
ggplot(cohort_full, aes(x = dose, y = resp, color = shift)) +
  geom_point(alpha = 0.5) +
  geom_line(aes(y = nlme_pred, group = shift), size = 1.2) +
  labs(
    title = "혼합 효과 모델 결과 (그룹별 파라미터)",
    x = "용량 (Dose)",
    y = "질병 반응 (Response)",
    color = "교대근무 여부"
  ) +
  theme_minimal(base_size = 14)
```



```r
summary(nlme_fit)
```

```
Nonlinear mixed-effects model fit by maximum likelihood
  Model: resp ~ a/(1 + exp(-(mu * (dose - lam))))
  Data: cohort_full
        AIC       BIC   logLik
  -634.3112 -591.3037 330.1556

Random effects:
 Formula: list(a ~ 1, mu ~ 1, lam ~ 1)
 Level: shift
 Structure: General positive-definite, Log-Cholesky parametrization
                StdDev       Corr
a.(Intercept)   9.047340e-07 a.(In) m.(In)
mu.(Intercept)  1.149202e-06 0
lam.(Intercept) 1.844210e-06 0      0
Residual        4.719949e-02

Fixed effects:  a + mu + lam ~ shift
                    Value  Std.Error  DF  t-value p-value
a.(Intercept)    1.195727 0.01404366 195 85.14359  0.0000
a.shift비교대    0.029480 0.03142533 195  0.93811  0.3493
mu.(Intercept)   0.835771 0.02898145 195 28.83812  0.0000
mu.shift비교대  -0.345475 0.03510659 195 -9.84075  0.0000
lam.(Intercept)  5.000389 0.05323858 195 93.92415  0.0000
lam.shift비교대  0.082515 0.14068227 195  0.58653  0.5582
 Correlation:
                a.(In) ashft대 m.(In) mshft대 lm.(I)
a.shift비교대   -0.447
mu.(Intercept)  -0.685  0.306
mu.shift비교대   0.565 -0.680  -0.826
lam.(Intercept)  0.770 -0.344  -0.527  0.435
lam.shift비교대 -0.291  0.910   0.200 -0.581  -0.378

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-2.53316379 -0.67449163 -0.07048116  0.62151401  3.28479392

Number of Observations: 202
Number of Groups: 2
```

```
# p-value 계산 및 출력 (tidy형식)
# fixed effects 결과를 추출합니다.
#fixed_effects <- summary(nlme_fit)$tTable
# p-value를 계산합니다. (t-value와 자유도(DF)를 이용)
#p_values <- 2 * (1 - pt(abs(fixed_effects[, "t-value"]), fixed_effects[, "DF"]))
# 계산된 p-value를 기존 summary 출력에 추가하여 보여줍니다.
#fixed_effects_with_p <- cbind(fixed_effects, `p-value` = p_values)
#print(fixed_effects_with_p)
```

혼합 효과 모델은 단순히 두 곡선을 그리는 것을 넘어, 두 그룹이 통계적으로 어떻게 다른지를 알려줍니다. summary(nlme_fit) 결과를 자세히 해석해 보겠습니다.

효과	파라미터	값 (Value)	p-value	의미
고정 효과	a.(Intercept)	1.957207	0.0000	**비교대 근무자의 최대 반응값(a)**입니다.
	a.shift	0.029483	0.0249	**비교대 근무자를 기준으로 교대 근무자의 a 값이 얼마나 다른지**를 나타냅니다. p-value가 0.05보다 커서 통계적으로 유의미한 차이가 없습니다.
	mu.(Intercept)	5.577100	0.0000	**교대 근무자의 기울기(mu)**입니다.
	mu.shift	0.345475	0.0001	**교대 근무자를 기준으로 비교대 근무자의 mu 값이 얼마나 다른지**를 나타냅니다. p-value가 0.0001로 매우 작아서, **두 그룹의 기울기에는 통계적으로 매우 유의미한 차이**가 있습니다.
	lam.(Intercept)	5.000389	0.0000	**교대 근무자의 시작점(lam)**입니다.
	lam.shift	0.825915	0.5582	**교대 근무자를 기준으로 비교대 근무자의 lam 값이 얼마나 다른지**를 나타냅니다. 통계적으로 유의미한 차이는 없습니다.

이 결과를 통해 우리는 **‘과로에 대한 질병 반응의 기울기(mu)는 교대 근무자와 비교대 근무자 간에 통계적으로 유의미하게 다르다’**는 결론을 내릴 수 있습니다. 이는

교대 근무 여부라는 공변량이 용량-반응 관계에 중요한 영향을 미친다는 강력한 증거가 됩니다.

8 R 기본 소개

8.1 R 소개

데이터 분석 및 통계 연구에 사용되는 강력한 프로그래밍 언어 및 소프트웨어 환경입니다. 데이터 처리, 분석, 시각화, 보고서 작성 등 다양한 작업을 수행할 수 있으며, 활발한 커뮤니티와 다양한 패키지를 통해 사용자의 요구에 맞는 기능을 확장할 수 있습니다.

8.2 R사용 방법

- R server 사용 RStudio Cloud 무료 버전의 용량 제한으로 인하여, 원활한 실습을 위해 R server 제공

1. https://sehnr.org/rtutor
2. 안내된 계정 사용

- .Rproj : R프로젝트 파일로, 일관된 작업을 보장하고, 사용자별 설정을 저장할 수 있습니다. 현재는 우선 생성되어 있는 상태입니다.
- 새로운 R script 생성
- R기본 설정 기본 구성 설정하기 > Tools -> Global Options

결과 도출을 잘하려면, 다음과 같은 설정을 선택해주어야 합니다. Code -> Display -> ☒

- R 사용 전 설정 및 R 프로그램 이해 다음과 같은 이름의 폴더를 생성합니다. 폴더는 'New Folder'에서, R script는 'New Blank File'을 클릭하여 새로 생성합니다.
- 폴더 별 사용 용도 data: 모든 원 또는 가공된 데이터 파일을 여기에 보관해야 합니다. rscript: 모든 r 스크립트가 저장되어 있어야 합니다.
    - datastep.R: 데이터 정리, 변환 및 전처리를 합니다.
    - analysis.R: 핵심 분석 방법(통계 테스트, 데이터 모델링 등)을 포함합니다.
    - sources: 추가 스크립트 하위 폴더입니다.

    * function.R: 서로 다른 스크립트에서 사용할 사용자 정의 함수를 정의해놓을 수 있습니다. results: plot, 테이블, 및 처리된 파일과 같은 모든 결과물을 여기에 저장합니다. manuscript: 프로젝트에 대해 작성 중인 보고서나 논문 초안, 메모 및 최종 버전을 저장합니다.

- Data 준비 저장하는 위치(Cloud > project > data)를 제대로 파악하고 업로드해야 파일 찾을 때 편리합니다. 현재 파일 링크를 통해서 데이터를 불러올 것이기에 따로 업로드하지 않아도 됩니다.
- 시각화를 위한 R 패키지 다운로드 및 불러오기 R에서 코드를 실행할 때는 반드시 ctrl + enter를 눌러야 합니다.

```
if(!require("tidyverse")) install.packages("tidyverse")
```

- require: 이 함수는 'tidyverse' 패키지가 이미 설치되어 있고 R 세션에서 사용할 수 있는지 확인합니다.
- !: 이 기호는 논리적 NOT을 나타냅니다. 이 함수는 require()의 결과를 반전시킵니다. 따라서 'tidyverse'가 설치되지 않은 경우(즉, require()가 FALSE를 반환하는 경우) !.require()는 TRUE가 됩니다.
- if(): 조건문으로, if 블록 안의 코드는 조건이 TRUE인 경우에만 실행됩니다. 이 경우 'tidyverse'가 설치되지 않았다면 조건은 TRUE가 됩니다.
- install.packages("tidyverse"): 이 명령어는 'tidyverse' 패키지를 설치합니다. 'tidyverse' 패키지가 아직 설치되지 않은 경우에만 실행됩니다.
- R 패키지 설치

```
#basic requirement
if(!require("tidyverse")) install.packages("tidyverse")
if(!require("htmlTable")) install.packages("htmlTable")
if(!require("broom")) install.packages("broom")
if(!require("labelled")) install.packages("labelled")
#packages from github
if(!require("devtools")) install.packages("devtools")
if(!require("ggplot2")) install.packages("ggplot2")
library(devtools)
if(!require("tabf")) install.packages("jinhaslab/tabf", force = TRUE)
library(tabf)
if(!require("readxl")) install.packages("readxl")
library("readxl")
```

- tidyverse: 데이터 시각화를 위한 ggplot2가 유명하며, 데이터 조작에 도움이 되는 dplyr과 tidyr도 있습니다.

- htmlTable: HTML 형식의 고급 테이블을 만들 수 있습니다. 웹 플랫폼에서 데이터를 표형식으로 표시해야 할 때 특히 유용합니다.
- broom: lm과 같은 R의 내장 함수의 지저분한 출력을 깔끔한 데이터 프레임으로 변환합니다. 통계 테스트나 모델의 결과를 시각화하고 싶을 때 유용합니다.
- ggplot2: 데이터 시각화를 위한 가장 유명한 R 패키지 중 하나입니다. 그래픽 문법을 기반으로 하며 복잡하고 사용자 지정 가능한 플롯을 만들기 위한 강력한 플랫폼을 제공합니다.
- tabf: GitHub에서 제공되는 패키지로, 테이블 서식 지정과 관련된 기능을 제공합니다.
- readxl: R에 엑셀파일을 불러올 때 사용하는 패키지입니다.

9 데이터 소개 및 준비

9.1 건강검진 데이터셋의 소개 및 구조에 대해 이해합니다.

- 범주형 변수 : 카테고리나 그룹으로 분류할 수 있는 변수입니다. 이 변수들은 수적인 의미를 가지지 않으며 단순히 분류하기 위한 목적으로 사용됩니다.
- 연속형 변수 : 연속적인 숫자로 표현되며, 사이에 무한히 많은 값이 존재할 수 있는 변수입니다. 이 변수들은 산수를 통해 양과 크기를 측정할 수 있습니다.

9.2 R을 사용하여 데이터를 불러오고 기본적인 탐색을 진행합니다.

- 데이터 불러오기 R에 저장해둔 데이터를 현재 분석을 진행하기 위해 데이터를 준비합니다.

```
data1 = readRDS("data/healthcheck.rds")
```

- readRDS(): 이 함수는 R에서 .rds 파일을 읽는 데 사용됩니다. .rds 파일은 R에서 데이터를 저장하는 특별한 파일 형식입니다.
- data1 = : 불러온 데이터를 data1이라는 이름의 변수에 저장하는 부분입니다. 이렇게 저장하면 나중에 data1을 사용해서 데이터를 분석하거나 그래프를 그릴 수 있습니다.

9.2.1 select

데이터가 어떻게 구성되었는지 확인하는 과정이 제일 먼저 필요합니다. 우리는 select을 통해서 데이터 내에서 원하는 변수명만 확인해서 볼 수도 있습니다.

처음 다섯 개의 변수를 선택해 보겠습니다. head()는 위에 6행만 보여주는 것입니다.

```
data1 %>% select("idv_id", "AGE", "sex") %>% head()
```

```
   idv_id AGE sex
1 3823099  31   1
2  761158  37   1
3 3915038  33   1
```

```
4 3108477  30   1
5 4621083  36   1
6 3108819  38   1
```

9.2.2 filter

특정한 조건을 준 데이터형태를 알고 싶다면, 'filter'를 통해 특정 기준을 충족하는 행을 선택합니다. 예를 들어 성별 또는 특정 연령대에 따라 개인을 선택할 수 있습니다. 이를 위해 여러 조건문이 사용됩니다.

'=='는 같음을 의미합니다.

```
data1 %>%
  select(sex, AGE) %>%
  filter(sex == 1) %>% head()
```

```
  sex AGE
1   1  31
2   1  37
3   1  33
4   1  30
5   1  36
6   1  38
```

'&'는 'AND', 'I'는 'OR'의 의미이며, 한 번에 여러 조건을 적용할 수 있습니다.'!'는 부정을 의미합니다.

```
data1 %>%
  select(sex, AGE) %>%
  filter(sex != 2) %>%
  filter(AGE >19 & AGE <60) %>% head()
```

```
  sex AGE
1   1  31
2   1  37
3   1  33
4   1  30
5   1  36
6   1  38
```

'%in%'은 여러 조건을 나열하여 선택할 수 있어 명목 변수에 편리합니다.

```
data1 %>% select(sex, AGE) %>%
  filter(sex %in% c(1,2)) %>%
  filter(!AGE < 15) %>% head()
```

```
  sex AGE
1   1  31
2   1  37
3   1  33
4   1  30
5   1  36
6   1  38
```

'is.na'는 데이터 내 변수들 중 NA누락값을 식별 가능합니다. Filter를 통해 변수 'waist'의 'NA누락값을 제외해보겠습니다. 그리고 새로운 데이터셋을 생성해보겠습니다.

우선 NA 누락값이 있는지 확인해야 합니다. tail()은 아래의 6행들을 보여줍니다.

```
data1 %>% count(waist) %>% tail()
```

```
   waist n
80   129 7
81   130 3
82   133 1
83   134 1
84   143 1
85    NA 5
```

count()는 변수값의 n수를 보여주고, tail()은 가장 밑에 행부터 6개를 보여줍니다.

```
data2 = data1 %>% filter(!is.na(waist))
data2 %>% count(waist) %>% tail()
```

```
   waist n
79   128 7
80   129 7
81   130 3
82   133 1
83   134 1
84   143 1
```

다시 누락값을 확인했을 때, NA값이 없어진 것을 확인할 수 있습니다.

9.2.3 mutate

'mutate'는 변수를 생성하는 데 자주 사용되는 기본 함수입니다. 이 함수에 익숙해지는 것이 중요합니다. mutate는 ifelse, case_when와 같은 다양한 조건문과 함께 자주 사용됩니다.

sex의 값인 '1,2'를 '남','여'으로 바꾸고, htn의 값인 '0,1'을 '정상','고혈압'으로 바꾸어서 data2에 새 변수들을 추가해보겠습니다.

```
data3 = data2 %>%
  mutate(sexgp = case_when(
    sex == 1 ~ '남',
    sex == 2 ~ '여'  )) %>%
  mutate(htngp = case_when(
    htn == 0 ~ '정상',
    htn == 1 ~ '고혈압'  ))
data3 %>% select(sexgp, htngp) %>% head()
```

```
  sexgp htngp
1    남  정상
2    남  정상
3    남  정상
4    남  정상
5    남  정상
6    남  정상
```

"case_when"을 통해 나이를 10살 단위로 구분하여 agegp라는 새로운 변수를 생성하겠습니다.

```
data4 = data3 %>%
  mutate(agegp = case_when(
    AGE < 20 ~ "10",
    AGE < 30 ~ "20",
    AGE < 40 ~ "30",
    AGE < 50 ~ "40",
    AGE < 60 ~ "50",
    TRUE ~ "60"
  ))
data4 %>% select(sexgp, htngp, agegp) %>% head()
```

```
  sexgp htngp agegp
1    남  정상    30
2    남  정상    30
```

```
3    남  정상    30
4    남  정상    30
5    남  정상    30
6    남  정상    30
```

```
rm(data3)
```

agegp 변수 생성시 case_when에서 “TRUE”는 앞서 설정한 조건 외에 해당하는 값인 60세 이상의 값들에 대해, ’60’로 생성하겠다는 의미입니다.

조건을 두개 이상 설정하여 BMI의 정상범위도 한번 설정해보겠습니다.

```
data5 = data4 %>%
  mutate(bmigp = case_when(
    BMI >= 18.5 & BMI <= 24.9 ~ "1.정상",
    BMI < 18.5 ~ "2.저체중",
    TRUE ~ "3.과체중"
  ))
data5 %>% select(sexgp, htngp, agegp, bmigp) %>% head()
```

```
  sexgp htngp agegp    bmigp
1    남  정상    30   1.정상
2    남  정상    30 3.과체중
3    남  정상    30   1.정상
4    남  정상    30   1.정상
5    남  정상    30   1.정상
6    남  정상    30 3.과체중
```

```
rm(data4)
```

9.2.4 pivot longer

데이터를 분석하거나 시각화할 때, 데이터를 긴 형식(long format)으로 변환하는 경우가 많습니다. 긴 형식은 여러 변수의 값을 하나의 열로 모으고, 해당 변수의 이름을 다른 열에 저장하는 방식입니다. 이를 통해 데이터가 더 유연하게 사용될 수 있으며, 특히 ggplot2를 이용한 시각화나 그룹별 분석에서 유용합니다.

> select을 이용해서 필요한 변수들만 선택한 후, pivot longer로 데이터를 변환해보도록 하겠습니다.

```r
data5 %>%
  select(idv_id, AGE, sexgp, agegp, bp_high, bp_lwst) %>%
  pivot_longer(
    cols = c(bp_high, bp_lwst),              # 변환할 열 지정
    names_to = "blood_pressure_type",        # 변수명을 저장할 열 이름
    values_to = "blood_pressure_value"       # 변수 값을 저장할 열 이름
  )
```

```
# A tibble: 202,696 x 6
    idv_id   AGE sexgp agegp blood_pressure_type blood_pressure_value
     <int> <int> <chr> <chr> <chr>                                <int>
 1 3823099    31 남    30    bp_high                                123
 2 3823099    31 남    30    bp_lwst                                 75
 3  761158    37 남    30    bp_high                                131
 4  761158    37 남    30    bp_lwst                                 77
 5 3915038    33 남    30    bp_high                                118
 6 3915038    33 남    30    bp_lwst                                 78
 7 3108477    30 남    30    bp_high                                119
 8 3108477    30 남    30    bp_lwst                                 73
 9 4621083    36 남    30    bp_high                                129
10 4621083    36 남    30    bp_lwst                                 75
# i 202,686 more rows
```

10 기초 데이터 분석

10.1 기초 통계 분석을 통해 데이터의 주요 특징을 파악합니다.

10.1.1 count

변수에서 변수값 수가 얼마나 있는지 파악해볼 필요가 있습니다. 먼저 고혈압의 n수와 성별의 n수를 파악해보겠습니다.

성별 n수 구하기

```
data5 %>% count(sex)
```

```
  sex     n
1   1 49443
2   2 51905
```

10.1.2 summarise

'summarise'은 열별로 정보를 결합하여 표시하는 함수입니다. 'summarise'과 함께 자주 사용되는 함수는 'mean,' 'sd' (standard deviation), 'median,' 'max,' 'min,' 등이 있으며 'quantile'도 자주 활용됩니다.

```
data5 %>%
  summarise(mean_age = mean(AGE),
            std_age  = sd(AGE))
```

```
  mean_age  std_age
1 44.56677 14.89363
```

10.1.3 group_by

데이터 탐색에서 group_by는 지정한 변수별 특정 결과값 도출시 자주 사용됩니다.

특히 연속형 변수의 경우 summarise, 명목형 변수의 경우 count와 함께 사용되어 변수별 n수 산출에 유용합니다. 고혈압 유무를 나타내는 변수 htn을 사용해 보겠습니다.

이때, 단순히 n수를 구하면, 고혈압이 많은지 판단하기 어렵습니다. 그래서 mutate를 이용해서 비율을 계산하여 남녀에서 고혈압의 비율이 어느정도인지 파악해보고자 합니다.

```
data5 %>%
  group_by(sexgp) %>%
  count(htngp) %>%
  mutate(prob = n/sum(n)) %>%
  filter(htngp == '고혈압')
```

```
# A tibble: 2 x 4
# Groups:   sexgp [2]
  sexgp htngp      n  prob
  <chr> <chr>  <int> <dbl>
1 남    고혈압 15483 0.313
2 여    고혈압 11175 0.215
```

10.1.4 chi square검정

이번에는 chi square검정을 r을 이용하여 수행해보도록 하겠습니다. [카이제곱 검정은 두 범주형 변수(예: 성별, 고혈압 여부)가 서로 관련이 있는지 알아보는 통계 방법이에요. 예를 들어, “남성과 여성의 고혈압 비율이 다를까?” 같은 질문에 답할 때 사용할 수 있습니다.]

두 범주형 변수 간의 독립성을 평가할 때 주로 사용됩니다. 이를 통해 연관성 여부를 간단하게 우선 확인해볼 수 있습니다.

```
# 2*2 교차표 생성
chi_table <- table(data5$sexgp, data5$htngp)
print(chi_table)
```

```
    고혈압  정상
  남  15483 33960
  여  11175 40730
```

```r
# 카이제곱검정
chi_sq_test_result <- chisq.test(chi_table)
print(chi_sq_test_result)
```

```
    Pearson's Chi-squared test with Yates' continuity correction

data:  chi_table
X-squared = 1250.3, df = 1, p-value < 2.2e-16
```

해당 결과를 통해서 피어슨 카이제곱 검정 결과, 성별(sexgp)과 고혈압 유무(htngp)는 통계적으로 매우 유의미한 연관성을 보이는 것으로 나타났습니다 (X2 =1250.3, df=1, p<0.001). 이는 남성과 여성 간 고혈압 유병률에 통계적으로 유의미한 차이가 존재한다는 것을 의미하며, 이러한 차이가 우연히 발생했을 가능성은 극히 낮다고 해석할 수 있습니다.

10.1.5 t test

이번에는 t test를 r을 이용하여 수행해보도록 하겠습니다. [t-검정(t-test)은 두 그룹의 평균이 서로 다른지 알아보는 통계 방법입니다. 예를 들어, “남학생과 여학생의 키 평균이 다를까?” 같은 질문에 답할 때 사용할 수 있습니다.]

```r
filtered_data = data5 %>%
  filter(!is.na(htn)) %>%
  select(sex, AGE, htn) %>%
  mutate(sexgp = case_when(
    sex == 1 ~ "남성",
    TRUE ~ "여성"
  ))
t_test_result <- t.test(AGE ~ sexgp, data = filtered_data)
t_test_result
```

```
    Welch Two Sample t-test

data:  AGE by sexgp
t = -9.8336, df = 100992, p-value < 2.2e-16
alternative hypothesis: true difference in means between group 남성 and group 여성 is not equal to 0
95 percent confidence interval:
 -1.1035835 -0.7367736
sample estimates:
```

```
mean in group 남성 mean in group 여성
          44.09550           45.01568
```

10.2 변수 간의 관계와 패턴을 탐색합니다.

데이터의 기본 구조를 파악하고자 표로 나타내려고 합니다. 표를 간단하게 만들 수 있는 tabf를 이용할 수 있습니다. ### TABLE 1

```
test1 = data5 %>%
  select(htngp, sexgp, job, AGE, dep, bmigp)

tabf(test1,
     stratas = "htngp",
     catVars = c("sexgp","job", "bmigp","dep"),
     conVars = c("AGE")
) %>%
  addHtmlTableStyle(align ="ll") %>%
  htmlTable()
```

	variables	values	고혈압	정상	p.value
1	AGE		46.9±14.6	43.7±14.9	<0.001
2	sexgp	남	15483 (31.3%)	33960 (68.7%)	<0.001
3		여	11175 (21.5%)	40730 (78.5%)	
4	job	사무직	1459 (24.1%)	4592 (75.9%)	<0.001
5		서비스,판매직	4716 (26.5%)	13103 (73.5%)	
6		농어업	4766 (23.7%)	15321 (76.3%)	
7		기술직	15717 (27.4%)	41674 (72.6%)	
8	bmigp	정상	11049 (20.0%)	44187 (80.0%)	<0.001
9		저체중	443 (11.9%)	3283 (88.1%)	
10		과체중	15166 (35.8%)	27220 (64.2%)	
11	dep	인사과	2676 (19.9%)	10758 (80.1%)	<0.001
12		연구개발과	3663 (22.0%)	12977 (78.0%)	
13		경영과	3769 (24.2%)	11836 (75.8%)	
14		영업과	3932 (26.1%)	11135 (73.9%)	
15		무역과	4066 (27.8%)	10566 (72.2%)	
16		시설관리과	4228 (31.6%)	9154 (68.4%)	
17		보관운송과	4324 (34.4%)	8264 (65.6%)	

10.2.1 TABLE 2

직종(job)별 고혈압과의 연관성이 있는지 또는 얼마나 있는지를 알아보는 표를 만들고자합니다. 이는, 로지스틱 회귀분석을 통해 알아낼 수 있습니다.

```
mod0 = data5 %>%
  glm(data=., family=binomial(),
      formula = htn == 1 ~
        job
  )
mod1 = data5 %>%
  glm(data=., family=binomial(),
      formula = htn == 1 ~
        job + agegp +sexgp + bmigp + dep
  )
summary(mod0)
```

```
Call:
glm(formula = htn == 1 ~ job, family = binomial(), data = .)

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)       -1.14656    0.03005 -38.152  < 2e-16 ***
job2.서비스,판매직  0.12468    0.03452   3.612 0.000304 ***
job3.농어업       -0.02115    0.03433  -0.616 0.537741
job4.기술직        0.17143    0.03148   5.446 5.14e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 116795  on 101347  degrees of freedom
Residual deviance: 116675  on 101344  degrees of freedom
AIC: 116683

Number of Fisher Scoring iterations: 4
```

```
summary(mod1)
```

```
Call:
glm(formula = htn == 1 ~ job + agegp + sexgp + bmigp + dep, family = binomial(),
    data = .)

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)       -2.02189    0.04250 -47.572  < 2e-16 ***
job2.서비스,판매직  0.13358    0.03559   3.753 0.000175 ***
job3.농어업        -0.04185    0.03537  -1.183 0.236737
job4.기술직         0.14327    0.03244   4.416 1.01e-05 ***
agegp30             0.24489    0.02495   9.814  < 2e-16 ***
agegp40             0.29025    0.02460  11.798  < 2e-16 ***
agegp50             0.50865    0.02484  20.480  < 2e-16 ***
agegp60             0.66818    0.02096  31.884  < 2e-16 ***
sexgp여            -0.39114    0.01502 -26.045  < 2e-16 ***
bmigp2.저체중      -0.49899    0.05234  -9.534  < 2e-16 ***
bmigp3.과체중       0.75887    0.01504  50.466  < 2e-16 ***
dep2.연구개발과     0.14726    0.02949   4.994 5.92e-07 ***
dep3.경영과         0.27435    0.02954   9.288  < 2e-16 ***
dep4.영업과         0.37493    0.02942  12.743  < 2e-16 ***
dep5.무역과         0.44156    0.02931  15.067  < 2e-16 ***
dep6.시설관리과     0.62972    0.02947  21.369  < 2e-16 ***
dep7.보관운송과     0.74537    0.02957  25.204  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 116795  on 101347  degrees of freedom
Residual deviance: 110404  on 101331  degrees of freedom
AIC: 110438

Number of Fisher Scoring iterations: 4
```

```
oddsTabf(mod0, mod1)
```

Table. OR(95%CI) for htn of 1

Variables	Values	Model.I	Model.II
job	1.사무직	1.00 (reference)	1.00 (reference)
	2.서비스,판매직	**1.13 (1.06-1.21)**	**1.14 (1.07-1.23)**
	3.농어업	0.98 (0.92-1.05)	0.96 (0.89-1.03)

Table. OR(95%CI) for htn of 1

Variables	Values	Model.I	Model.II
	4.기술직	**1.19 (1.12-1.26)**	**1.15 (1.08-1.23)**
agegp	20		1.00 (reference)
	30		**1.28 (1.22-1.34)**
	40		**1.34 (1.27-1.40)**
	50		**1.66 (1.58-1.75)**
	60		**1.95 (1.87-2.03)**
sexgp	남		1.00 (reference)
	여		**0.68 (0.66-0.70)**
bmigp	1.정상		1.00 (reference)
	2.저체중		**0.61 (0.55-0.67)**
	3.과체중		**2.14 (2.07-2.20)**
dep	1.인사과		1.00 (reference)
	2.연구개발과		**1.16 (1.09-1.23)**
	3.경영과		**1.32 (1.24-1.39)**
	4.영업과		**1.45 (1.37-1.54)**
	5.무역과		**1.56 (1.47-1.65)**
	6.시설관리과		**1.88 (1.77-1.99)**
	7.보관운송과		**2.11 (1.99-2.23)**

- mod0
- dat1 %>% glm(…): 이 줄은 일반화된 선형 모델(glm)을 나타냅니다.
- data=., 는 파이프의 왼쪽에서 전달된 데이터 집합을 나타냅니다. 이 경우 dat4입니다.
- family=binomial( ): 모델이 이진 응답 변수(예: 예/아니오, 성공/실패)에 적합한 이항 분포를 사용하도록 지정합니다.
- formula = htn == 1 ~ job: 모델의 공식을 정의합니다. 이 모델은 예측 변수 job의 함수로서 htn이 1(고혈압)과 같을 확률을 모델링합니다.
- mod1: mod0과 유사하나 이번에는 모델에서 dep 외에 다른 변수들을 추가하고자 합니다.
- 모델의 목적:
    - 두 모델 모두 고혈압(htn)와 직종(job) 간의 관계를 이해하려고 합니다.
    - mod0은 고혈압에 대한 부서의 개별 효과를 분석합니다.
    - mod1은 다른 변수들을 조정한 후 고혈압과 부서 간의 관계를 분석합니다.

11 데이터 시각화

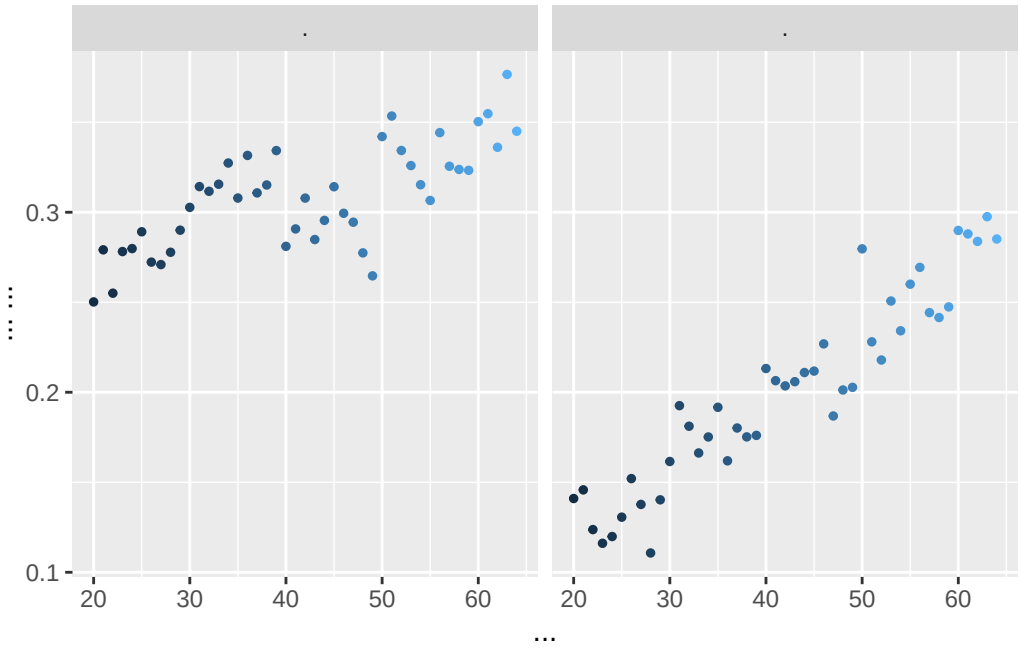
11.1 기본적인 시각화 기술, 예를 들어 막대 그래프, 선 그래프 등을 배웁니다.

11.1.1 ggplot 종류

11.1.2 실제 그려보기

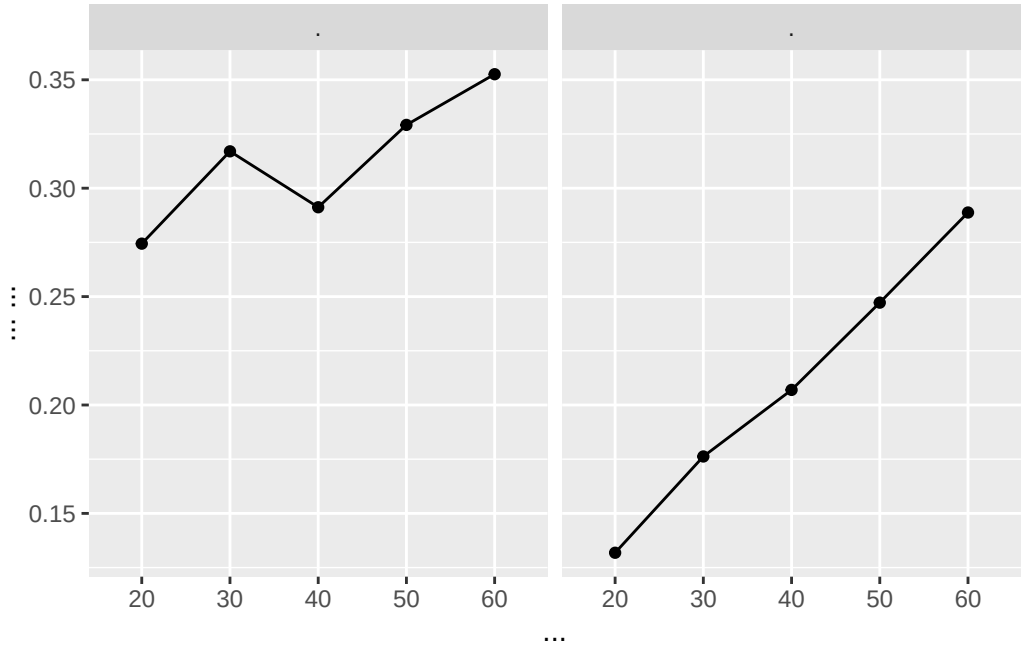
- 산점도 그래프 geom_point 성별에 따라 나이대별 고혈압을 얼만큼 앓고 있는지 한눈에 보기 위해 선그래프를 그려보겠습니다.

```
data5 %>%
  group_by(sexgp, AGE) %>%
  count(htn) %>%
  mutate(prob = n / sum(n)) %>%
  filter(htn == 1) %>%
  ungroup() %>%
  ggplot(aes(x = AGE, y = prob, color = AGE, group = sexgp)) +
  geom_point(size = 1) +
  labs(x = "연령대", y = "고혈압 유병률") + # x축과 y축 이름 추가
  theme(legend.title = element_blank()) +
  theme(legend.position = "none") +
  facet_wrap(~sexgp)
```



- group_by(sex, AGE): 성별과 연령으로 데이터를 그룹화합니다.
- count(htn): 고혈압 발생 여부(htn 변수)에 따라 각 그룹의 빈도를 계산합니다.
- mutate(prob = n / sum(n)): 각 그룹에서 고혈압 발생 확률을 계산합니다.
- filter(htn == 1): 고혈압이 있는 경우만을 필터링합니다.
- ungroup(): 그룹화를 해제합니다.
- ggplot(aes(x = age_group, y = prob, color = age_group, group = sex)): ggplot 객체를 생성하고, x축에는 연령 그룹을, y축에는 고혈압 발생 확률을 매핑합니다. 선 색상은 연령 그룹으로 구분하고, group = sex로 각 성별별로 다른 그룹으로 구분합니다.
- geom_point(): 점을 추가하여 각 그룹의 데이터를 표시합니다.
- theme(legend.title = element_blank()) : 범례(legend)의 제목을 없앱니다.
- theme(legend.position = “none”) : 범례의 위치를 삭제함으로써, 범례를 지웁니다. 범례를 나타낼 경우의 그래프는 아래와 같습니다.
- facet_wrap(~sex): 성별에 따라 서로 다른 그래프 패널로 분할하여 시각화합니다.
- 선 그래프 geom_line 성별에 따라 나이대별 고혈압을 얼만큼 앓고 있는지 한눈에 보기 위해 선그래프를 그려보겠습니다.

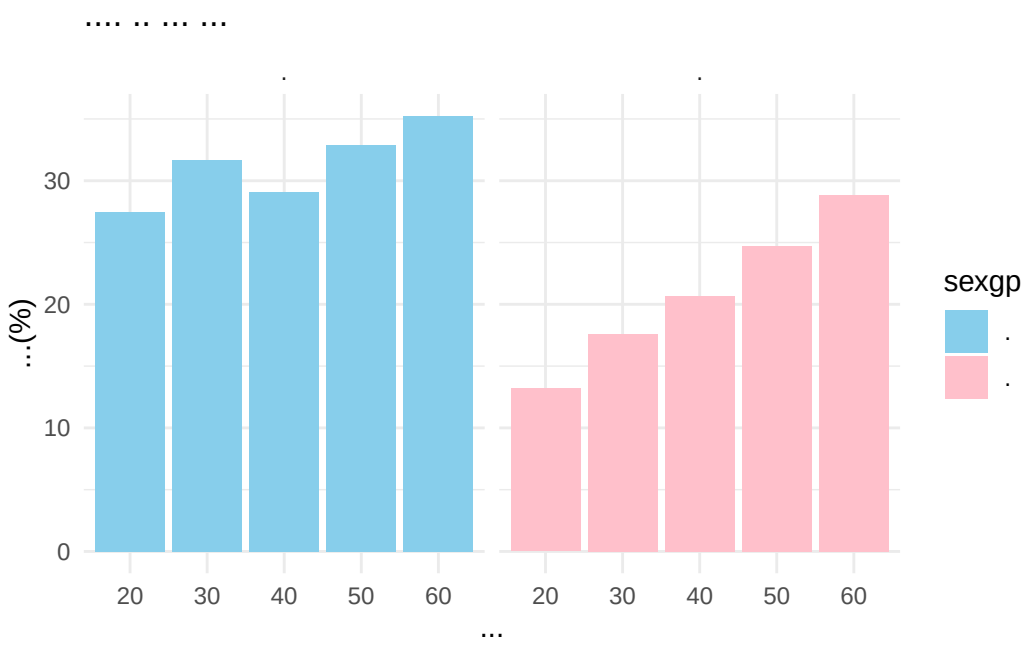
```r
data5 %>%
  group_by(sexgp, agegp) %>%
  count(htn) %>%
  mutate(prob = n / sum(n)) %>%
  filter(htn == 1) %>%
  ungroup() %>%
  ggplot(aes(x = agegp, y = prob, group = sexgp)) + # Use interaction(dep, sex) for distinct line
  geom_point() +
  geom_line() +
  labs(x = "연령대", y = "고혈압 유병률") +
  facet_wrap(~sexgp)
```



- geom_line(): 선을 추가하여 데이터 간의 경향성을 시각화합니다.
- 막대그래프 동일한 현상을 막대그래프로 그려보겠습니다.

```r
data5 %>%
  group_by(sexgp, agegp) %>%
  count(htn) %>%
  mutate(prob = n / sum(n)*100) %>%
  filter(htn == 1) %>%
  ungroup() %>%
  ggplot(aes(x = agegp, y = prob, fill = sexgp)) +
```

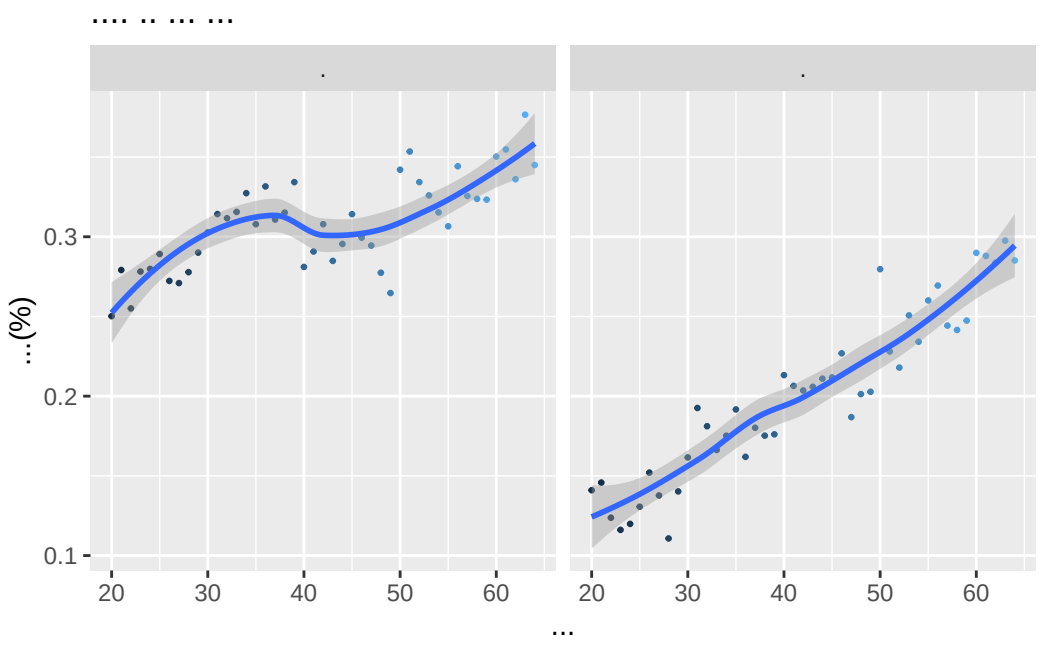
```
  geom_bar(stat = "identity", position = "dodge") +
  scale_fill_manual(values = c("skyblue", "pink")) +  # 색상 설정
  facet_wrap(~sexgp) +
  labs(title = "연령대에 따른 고혈압 유병률", x = "연령대", y = "유병률(%)") +
  theme_minimal()
```



- geom_bar(): 막대를 이용하여 성별별 연령그룹별 고혈압 발생 확률을 시각화합니다.
- 추세선 그래프 동일한 현상을 추세선 그래프를 바탕으로 추세선 및 산점도 그래프를 그려보겠습니다.

```
data5 %>%
  group_by(sexgp,AGE) %>%
  count(htn) %>%
  mutate(prob = n / sum(n)) %>%
  filter(htn == 1) %>%
  ungroup() %>%
  ggplot(aes(x = AGE, y = prob, color = AGE, group = sexgp)) +
  geom_point(size = 0.5) +
  geom_smooth() +
  labs(title = "연령대에 따른 고혈압 유병률", x = "연령대", y = "유병률(%)") +
  theme(legend.title = element_blank()) +
```

```
  theme(legend.position = "none") +
  facet_wrap(~sexgp)
```



- geom_smooth(): 추세선을 활용하여 연속형 변수인 AGE의 성별별 추세를 시각화합니다.