데이터 과학을 위한 통계 1강

1. 탐색적 데이터 분석(EDA)

개론: 20세기 초 ''실험계획법''과 ''최대우도추정''의 핵심 개념을 소개한 로널드 피셔는 현대 통계학의 대표적 선구자이다. 이 두 개념을 비롯한 여러 통계 개념은 데이터 과학 곳곳에 깊숙이 자리하고 있다. 이 책의 중요한 목표는 이러한 개념들을 분명히 이해하고, 데이터 과학과 빅데이터라는 측면에서 그것들이 왜 중요한지 동시에 부족한 것은 무엇인지를 정확하게 파악하는데 있다.

본론 : 이 장에서는 모든 데이터 과학 프로젝트의 첫 걸음이라고 할 수 있는 자료 탐색에 대해 알아본다. 탐색적 데이터 분석(Exploratory data analysis)(EDA)는 통계학에서 비교적 새로운 영역이다. 이전의 통계학에서는 추론, 즉 적은 표본(샘플)을 통해 더 큰 모집단에 대한 결론을 도출하기 위한 일련의 복잡한 과정에 관해 주로 다뤘다.

하지만 존 투키의 ''데이터 분석의 미래''라는 대표 논문을 통해 통계학의 개혁을 요구했다. 그는 통계적 추론을 하나의 구성 요소로 보는 데이터 분석이라는 새로운 과학적 학문을 제안했다.

1.1 정형화된 데이터의 요소

데이터 과학에서 가장 중요한 도전은 폭발적인 양의 원시 데이터를 활용 가능한 형태의 정보로 변환하는 것이다. 이책에서 다룰 통계적 개념들을 활용하기 위해서는 정형화 되지 않은 원시 데이터를 가공하여 (마치 관계형 DB에서 뽑은 듯한) 정형화된 형태로 변환하거나 혹은 애초에 특정한 연구 목적을 가지고 수집해야 한다.

연속형, 이산, 범주형, 이진. 순서형 등 다양한 종류의 데이터 들이 있다.

데이터 종류를 분휴하는 이 귀찮은 일을 왜 하는 걸까? 데이터를 분석하고 예측을 모델링할 때, 시각화, 해석, 통계 모델 결정 등에 데이터가 중요한 역활을 하기 때문이다. R이나 파이썬 같은 데이터 과학 SW들은 실제로 계산 성능을 향상시키기 위해 이러한 데이터 종류 정보를 활용한다. 더 중요한 것은 SW는 변수의 종류에 따라 해당 변수에 관련된 계산을 어떤 식으로 수행할지 결정한다는 점이다.

아마 SW엔지니어나 DB프로그래머라면, 범주형이니 순서형이니 하는 이러한 구분이 분석에 왜 필요한가 궁금할 수도 있겠다. 범주라는 것은 결국 문자나 숫자의 집합일 것이고, 기본적으로 DB는 이러한 내부적으로 자동 처리해 주기 때문이다.

데이터가 문자열인지 아니면 일정한 범위가 주어진 범주형인지 확실히 구분할 경우 다음과 같은 이점이 생긴다.

1. 데어터가 범주형이라는 정보는 SW가 차트 생성이나 모텔 피팅 등 통계분석을 수행하는 방식을 결정하는데 큰 도움을 준다. 예를 들어 R이나 파이썬에서는 순서형 데이터를 ordered.factor라고 구분하여 표현하고 이를 차트, 테이블, 통계 모델에서 사용자가 원하는 수너를 유지하는데 사용한다.
2. 관계형 DB에서처럼, 저장소와 인덱싱을 최적화하는데 사용한다.
3. 범주형 변수가 취할 수 있는 값들은 SW적으로 처리가 가능하다.

세번 째 이점은 뜻밖의 예상치 못한 결과를 가져온다. R에서 데이터를 읽어오는 데 사용하는 함수(read.csv)는 기본저긍로 텍스트 형태의 열 데이터를 factor로 자동 변환한다. 따라서 이 열에는 처음 읽은 문자열 값만 허용되고, 만약 새로운 텍스트 값을 할당하면 경고와 함께 NA(결측값)가 할당된다.

주요개념

1. 일반적으로 SW에서는 데이터를 종류별로 구분한다.
2. 데이터 종류에는 연속, 이산, 범주. 순서 형태가 있다.
3. SW에서 데이터 종류를 정하는 것은 해당 데이터를 어떻게 처리할지를 정하는 것과 같다.

1.2 테이블 데이터

테이블 데이터 : (ex 엑셀 스프레드시트 or 데이터베이스의 테이블)

cf) 용어 정리

데이터 프레임 : 통계와 머신러닝 모델에서 가장 기본이 되는 (스프레드시트와 같은) 테이블 형태의 데이터 구조

feature : 일반적으로 테이블의 각 열이 하나의 피처를 의미한다 (유의어 : 특징, 속성, 입력. 예측변수. 변수)

레코드 : 일반적으로 테이블의 각 행은 하나의 레코드를 의미한다. (유의어 : 기록값, 사건, 사례, 예제, 관측값)

테이블 데이터는 기본적으로 각 레코드(사건)를 나타내는 행과 피처(변수)를 나타내는 열로 이뤄진 이차원 행렬이라고 할 수 있다. 앞에서도 언급했지만, 데이터가 항상 이런 형태로 얻어지지 않는다. 예를 들어 문자열 같은 비정형 데이터는 테이블 데이터의 피처 형태로 표현되도록 처리해야 한다. 데이터 분석이나 모델링을 하기 위해 관계형 DB에 있는 데이터를 불러올 때도 역시 마찬가지로 하나의 테이블 형태로 변환해야 한다.

1.2.1 데이터 프레임과 인덱스

보통 DB에서 하나 혹은 그 이상의 열을 인덱스로 지정한다. 이를 통해 SQL쿼리 성능을 크게 향상시킬 수 있다. 파이썬의 pandas와 같은 라이브러리에서는 기본 테이블형 데이터 구조를 위해 DataFrame 객체를 제공한다. 기본적으로 DataFrame에서는 각 행마다 순차적으로 정수인 값을 붙여 이를 인덱스로 사용한다. 또한 pandas는 다중/계층적 인덱스를 설정할 수 있도록 되어 있어 좀 더 복잡한 동작도 효과적으로 처리할 수 있다.

이와 유사하게 R에서도 data.frame이라는 객체를 제공한다. data.frame 역시 내부적으로 행 번호에 따라 정수로 된 인덱스를 갖고 있다. row.names 속성을 조정하면 사용자가 원하는 키를 만들 수도 있다.

하지만 data.frame은 기본적으로 다중 인덱스를 지원하지 않는다.

이러한 단점을 보완하기 위해 나온 두 가지 패키지 **data.table**과 **dplyr**가 널리 사용되고 있다. 모두 다중 인덱스를 지원하고 data.frame을 다루는데 상당한 속도 개선을 가져왔다.

용어차이

통계학자들은 **응답변수**나 **종속변수**를 예측하는 모델에서 **예측변수**라는 용어를 사용한다.

데이터 과학자들은 **목표**를 예측하는데 **피처**를 사용한다, 컴퓨터 과학을 하는 사람은 보통 각각의 행을 하나의 **샘플**

이라고 부르는 반면 통계학자는 여러 행의 집한을 하나의 **샘플**이라고 한다.

1.2.2 테이블 형식이 아닌 데이터 구조

테이블 형식이 아닌 다른 형태의 데이터 구조도 있다. 시계열 데이터는 동일한 변수 안에 연속적인 측정값을 갖는다. 이는 통계적 예측 기법드릉ㄹ 위한 원재료가 되며, 사물 인터넷과 같이 다양한 디바이스에서 생산되는 데이터들에서 중요한 요소이다.

지도 제작과 위치 정보 분석에 사용되는 공간 데이터의 경우, 테이블 데이터보다 좀 더 복잡하고 다양하다.

객체(ex 주택)를 표현할 때는, 어떤 객체와 그것의 공간 좌표가 데이터의 중심이된다.

반면 **필드**정보는 공간을 나타내는 작은 단위들과 적당한 측정 기준값(ex 픽셀의 밝기)에 중점을 둔다.

주요개념

1. 데이터 과학에서 기본이 되는 데이터 구조는 행과 열이 각각 레코드와 변수(피처)를 의미하는 테이블 모양의 행렬이다.
2. 용어가 혼란스러울 수 있으니 주의하자 . 데이터 과학에 관련된 서로 다른(통계학, 컴퓨터 과학, 정보 공학)은 저마다 다양한 용어를 사용한다.

1.3 위치 추정

데이터를 표현하는 변수들은 보통 수천 가지 다른 값을 갖는다. 데이터가 주어졌을 때, 데이터를 살펴보는 가장 기초적인 단계는 각 피처(변수)의 '대푯값'을 구하는 것이다. 이는 곧 대부분의 값이 어디쯤에 위치하는지(중심경향성)를 나타내는 추정값이다.

용어정리

가중평균 : 가중치를 곱한 값의 총합을 가중치의 총합으로 나눈값***

중간값 : 데이터에서 가장 가운데 위치한 값

가중 중간값 : 데이터를 정렬한 후, 각 가중치 값을 위에서부터 더할 때, 총합의 중간이 위치하는 데이터 값

절사평균 : 정해진 개수의 극단값을 제외한 나머지 값들의 평균

로버스트하다 : 극단값들에 민감하지 않다는 것을 의미한다. (유의어 저항성 있다)

특잇값 : 대부분의 값과 매우 다른 데이터 값

데이터를 요약하려면 그냥 데이터의 평균을 구하기만 하면 되지 않느냐고 생각 할지도 모르겠다. 사실 평균이 계산하기도 쉽고 사용하기도 편리하다. 하지만 평균이 데이터의 중간을 대표하는 가장 좋은 방법은 아니다. 몇 가지 이유로, 통계학자들은 평균을 대체할 만한 다른 값들을 개발해냈다.

NOTE 측정 지표와 추정값

통계학자들은 보통 데이터로부터 얻은 값과 실제 상태를 나타내는 이론적인 참값을 구분하기 위해, 데이터로부터 계산된 값들을 보통 **추정값(estimate)**이라는 용어를 사용한다. 반면 데이터 과학자나 비즈니스 분석가들은 이러한 값들을 **측정 지표(metric)**라고 부른다. 이러한 차이는 곧 통계학과 데이터 과학의 접근법 차이를 반영한다.

통계학이라는 분야는 궁극적으로 불확실성을 이해하고자 하는 반면,

데이터 과학은 구체적인 비즈니스나 조직의 목표치에 관심을 둔다.

그러므로, 통계학자들은 추정한다고 하고,

> 데이터 과학자들은 측정한다고 한다.

1.3.1 평균

1) 절사평균

평균을 조금 변형한 것 중 하나로 **절사평균**이 있다. 절사평균은 값들을 크기 순으로 정렬한 후, 양끝에서 일정 개수의 값들을 삭제한 뒤 남은 값들을 가지고 구한 평균을 말한다.

정렬한 값들 $x_{(1)}, x_{(2)}, \cdots, x_{(n)}$ 이라고 할 때, 즉 $x_{(1)}$이 가장 작은 값 , $x_{(n)}$이 가장 큰 값을 의미한다.

p개의 가장 크고 작은 값들을 제외한 뒤 절사평균을 계산하는 수식은 다음과 같다.

$$\text{절사평균} = \overline{x} = \sum_{i=p+1}^{n=p} x_{(i)} / (n - 2p)$$

절사평균은 극단값의 영향을 제거한다.

2) 가중평균

각 데이터값 x_i 값에 가중치 w_i를 곱한 값들의 총합을 다시 가중치의 총합으로 나눈 것이다. 가중평균은 다음과 같은 수식으로 나타낼 수 있다.

$$\text{가중평균} = \overline{x}_w = \sum_{i=1}^{n} w_i x_i / \sum_{i}^{n} w_i$$

가중평균을 사용하는 두 가지 중요한 이유가 있다.

1. 어떤 값들이 본래 다른 값들에 비해 큰 변화량을 갖을 때, 이러한 관측값에 대해 더 작은 가중치를 줄 수 있다. 예를 들어 여러 개의 센서로부터 평균을 구한다고 할 때, 한 센서의 정확도가 떨어진다면 그 센서에서 나온 데이터에는 낮은 가중치를 주는 것이 합리적일 것이다.
2. 데이터를 수집할 때, 우리가 관심 있는 서로 다른 대조군에 대해 항상 똑같은 수가 얻어지지는 않는다. 예를 들어 온라인 실험을 진행했을 때, 모든 사용자 그룹에 대해 정확히 같은 비율을 반영하는 데이털르 수집하기는 참 어렵다. 이를 보정하기 위해, 데이터가 부족한 소수 그룹에 대해 더 높은 가중치를 적용할 필요도 있다.

1.3.2 중간값과 로버스트 추정

가중평균을 사용하는 이유와 마찬가지로, 가중중간값을 사용할 수도 있다. 중간값을 계산할 때와 마찬가지로 먼저 데이터를 정렬한다. 각 데이터 값은 이에 해당하는 가중치를 가지고 있다. 가중 중간값은 단순히 가운데 위치한 값이 아닌, 어떤 위치를 기준으로 상위 절반의 가중치의 합이 하위 절반의 가중치의 합과 동일한 위치의 값이 된다. 중간값과 마찬가지로 가중 중간값 역시 특잇값에 로버스트 하다.

특잇값

중간값은 결과를 왜곡할 수도 있는 특잇값(극단값)들의 영향을 받지 않으므로 로버스트한 위치 추정 방법이라고 알려져 있다. 특잇값은 어떤 데이터 집합에서 다른 값들과 매우 멀리 떨어져 있는 값들을 말한다. 다양한 데이터 요약과 시각화 방법에서 관습적으로 사용하는 특잇값에 대한 정의가 있기는 하지만, 사실 특잇값의 정확한 정의는 다소 주관적일 수 있다.

이상 검출

전형적인 데이터 분석에서 특잇값들은 가끔 유익한 정보를 제공하기도 하고, 때로는 골칫거리가 되기도 한다. 하지만 이와 반대로, 이상 검출에서는 대부분의 정상적인 데이터보다는 예외적으로 측정된 특잇값들이 바로 주된 관심의 대상이 된다.

중간값만이 로버스트한 위치를 추정하는 유일한 방법은 아니다. 사실 절사평균 역시 특잇값의 영향을 줄이기 위해 많이 사용된다. 예를 들어 데이터 상위 하위 10% (일반적인 선택)를 잘ㄹ라내는 방법은 데이터가 너무 작지만 않다면, 특잇값으로부터 데이터를 보호할 수 있다. 절사 평균은 중간값과 평균의 절충안이라고 볼 수 있다. 데이터의 특잇값들로부터 로버스트하지만 위치 추정을 위해 더 많은 데이터를 사용한다.

TIP 다른 로버스트한 위치 추정 방법

통계학자들은 원래 평균보다 좀 더 로버스트하면서 동시에 더 효율적인(즉, 데이터 집합들 사이의 작은 위치차이도 더 잘 분별할 수 있는)추정법을 개발하려는 목표를 가지고, 정말 다양한 위치 추정법을 개발해왔다. 이러한 방법들은 아마도 작은 데이터에 유용할 수 있다. 하지만 큰 데이터, 아니 어느 정도 크기가 되는 데이터에는 장점이 있다고 보기 어렵다.

1.3.3 예제

```
state <- read.csv(file="C:/Users/김대현/Desktop/data/chap1/state.csv")
> mean(state[["Population"]]) #평균
[1] 6162876
> mean(state[["Population"]], trim=0.1)  #절사평균(각 끝에서 10%를 제외한다.)
[1] 4783697
> median(state[["Population"]])  #중간값
[1] 4436370
```

만약 미국 전체의 평균적인 살인율을 계산하려면, 주마다 다른 인구를 고려하기 위해 가중평균이나 가중 중간값을 사용해야 한다. 기본 R에서는 이러한 함수르 제공하지 않기에 matrixStats라는 패키지를 설치해야 한다.

```
> mean(state[["Murder.Rate"]])
[1] 4.066    #평균
> library("matrixStats")
> weighted.mean(state[["Murder.Rate"]], w=state[["Population"]])
[1] 4.445834   #가중평균
> weightedMedian(state[["Murder.Rate"]], w=state[["Population"]])
[1] 4.4         #가중 중간값
```

1.4 변이추정

위치는 데이터의 특징을 요약하는 다양한 요소들 중 하나이다. 두 번째 요소인 **변이**는 데이터 값이 얼마나 밀집해 있는지 혹은 퍼져 있는지를 나타내는 **산포도**를 나타낸다. 변이를 측정하고, 이를 줄이고, 실제 변이와 랜덤을 구분하고, 실제 변이의 다양한 요인들을 알아보고, 변이가 있는 상황에서 결정을 내리는 등, 통계의 핵심에 이 변이가 있다.

용어정리

편차 : 관측값과 위치 추정값사이의 차이(유의어: 오차, 잔차)

분산 : 평균과의 편차를 제곱한 값들의 합을 n-1로 나눈 값, n은 데이터의 개수(유의어 : 평균제곱오차)

표준편차 : 분산의 제곱근(유의어 : IS 노름, 유클리드 노름)

평균절대편차 : 평균과의 편차의 절댓값 평균(유의어 : ㅣ1노름, 맨하탄 노름)

중간값의 중위절대편차 : 중간값과의 편차의 절댓값 중간값

범위 : 데이터의 최댓값과 최솟값의 차이

순서통계량 : 최소에서 최대까지 정렬된 데이터 값에 따른 계량형(유의어 : 순위)

백분위수 : 어떤 값들의 P퍼센트가 이 값 혹은 더 작은 값을 갖고, (100-P) 퍼센트가 이 값을 혹은 더 큰 값을 갖도록 하는 값(유의어 : 분위수)

사분위범위 : 75번째 백분위수와 25번째 백분위수 사이의 차이 (유의어 : IQR)

위치를 추정하는 데 다양한 방법(평균, 중간값 등)이 있었던 것 처럼, 변이를 추정하는 데도 다양한 방법이 있다.

1.4.1 표준편차와 관련 추정값들

가장 대표적으로 사용되는 변위 추정들은 관측 데이터와 위치 추정값 사이의 차이, 즉 편차를 기본으로 한다.

{1,4,4}라는 데이터가 있다고 할 때, 평균은 3이고 중간값은 4이다. 평균에서의 편차는 1-3=-2, 4-3=1, 4-3=1 이다. 이러한 편차는 데이터가 중앙값을 주변으로 얼마나 퍼져 있는지 말해준다.

변이를 측정하는 한 가지 방법은 바로 이러한 편차들의 대푯값을 추정하는 것이다. 그렇다고 편차 자체의 평균을 구하는 것은 바람직하지 않다. 음의 편차는 양의 편차를 상쇄해버리기 때문이다. 사실, 평균을 기준으로 편차의 합은 항상 0이 된다. 대신 사용할 수 있는 간단한 방법은 편차의 절댓값의 평균을 구하는 것이다. 앞의 예에서 편차의 절댓값은 {2,1,1}과 같고 그 평균은 (2+1+1)/3 = 1.33이다. 이는 **평균절대편차** 라고 하며 식은 다음과 같다.

$$\text{평균절대편차} = \sum_{i=1}^{n} |x_i - \overline{x}|/n$$

여기서 $\overline{x}$는 표본평균을 의미한다.

가장 유명한 변이 추정 방법은 제곱 편차를 이용하는, **분산**과 **표준편차**이다. 분산은 제곱 편차의 평균이고 표준편차는 분산의 제곱근이다.

$$\text{분산} = s^2 = \sum (x_i - \overline{x})^2 / (n-1)$$

$$\text{표준편차} = s = \sqrt{\text{분산}}$$

표준편차는 원래 데이터와 같은 척도에 있기에 분산보다 훨씬 해석하기가 쉽다. 그렇다 해도 수식이 복잡하고 한눈에 들어오지 않기에, 통계학에서 표준편차를 평균절대편차보다 선호하는게 이상할 수도 있다. 수학적으로 제곱한 값이 절댓값보다 통계 모델을 다루는게 더 편리하다는 통계 이론이 이를 뒷받침한다.

cf) n 아니면 n-1?

통계학 책을 보다 보면, 자유도라는 개념을 설명하기 위해, 분산을 구할 때 왜 n이 아닌 n-1을 분모로 사용하는지에 대한 논의가 등장한다. 만약 n이 충분히 커서 n으로 나누든 n-1로 나누든 차이가 거의 없다면 이러한 구분이 별로 중요하지 않을 것이다. 하지만 여기 관심 갈만한 이야기가 있다. 표본을 통해 모집단을 추정하고자 한다는 전제를 기본으로 하자

*만약 분산 수식에 n을 분모로 사용한다면 모집단의 분산과 표준편차의 참값을 과소평가하게 된다. 이를 **편향**추정이라고 부른다. 하지만 만약 n대신 n-1로 나눈다면, 분산은 **비편향**추정이 된다.*

왜 n을 사용하는 게 편향 추정이 되는지를 설명하려면, 추정값을 계산할 때 제약조건의 개수를 의미하는 자유도에 대해 언급할 필요가 있다. 이 경우, 표준편차는 표본의 평균에 따른다는 하나의 제약 조건을 가지고 있기에 n-1의 자유도를 갖는다. 보통 대부분의 경우 데이터 과학자들은 자유도에 크게 신경 쓰지 않아도 되지만 때로는 이 개념이 중요할 때가 있다.

분산, 표준편차, 평균절대편차 모두 특잇값과 극단값에 로버스트하지 않다. 분산과 표준편차는 제곱 편차를 사용하기에 특히 특잇값에 민감하다.

로버스트한 변위 추정값으로는 중간값으로부터의 **중위절대편차(MAD)**가 있다.

$$\text{중위절대편차} = \text{중간값}(|x_i - m|, |x_i - m|, \ldots, |x_N - m|)$$

여기서 m은 데이터의 중간값을 의미한다. 중간값의 특징을 따라 MAD는 극단값의 영향을 받지 않는다. 절사평균과 유사하게 절사 표준편차를 계산하는 것 역시 가능하다.

cf)분산, 표준편차, 평균절대편차, 중위절대편차 모두 동일한 추정은 아니지만, 모두 데이터가 정규분포에서 왔다고 가정하고 있다. 사실 표준편차는 항상 평균절대편차보다 크다. 그리고 평균절대편차는 중위절대편차보다 크다. 중위절대편차에 척도인자상수(약 1.4826)를 곱하면 정규분포의 경우에 표준편차와 같은 척도에서 MAB를 사용할 수 있다.

1.4.2 백분위수에 기초한 추정

변위를 추정하는 또 다른 접근은 정렬된 데이터가 얼마나 퍼져 있는지를 보는 것이다. 정렬(순위) 데이터를 나타내는 통계량을 **순서통계량**이라고 부른다. 여기서 가장 기본이 되는 측도는 가장 큰 값과 작은 캆의 차이를 나타내는 **범위**이다. 최대 최솟값 자체가 특잇값을 분석하는 데 큰 도움을 준다. 그렇지만 이 범위는 특잇값에 매우 민감하며 데이터의 변위를 측정하는 데 유용하지 않다.

특잇값에 민감한 것을 피하기 위해, 범위의 양 끝에서 값들을 지운 후, 범위를 다시 알아 볼 수도 있다. 좀 더 구체적으로는, **백분위수**사이의 차이를 가지고 이런 식의 추정을 하는 방법들이 있다. 데이터에서 P번째 백분위수는 P퍼센트의 값이 그 값 혹은 그보다 작은 값을 갖고 (100-P)퍼센트의 값이 그 값 혹은 그 보다 큰 값을 갖는 것을 의미한다.

변위를 측정하는 가장 대표적인 방법은 **사분위범위**라는, 25번째 백분위수와 75번째 백분위수의 차이를 보는 것이다.

예시로써, 3,1,5,3,6,7,2,9가 있다고 하자. 이것을 정렬하면 1,2,3,3,5,6,7,9가 된다. 25번째 2.5, 75번째 6.5이다. 6.5-2.5=4이다. SW마다 방법이 조금씩 달라 결과가 다를 수 있다. 하지만 일반적으로 그 차이는 작다.

대이터 집합이 매우 클 경우, 정확한 백분위수를 계산하기 위해 모든 값을 정렬하는 것은 매우 많은 연산을 필요로 한다. 머신러닝과 통계 SW에서는 백분위수의 근삿값을 사용한다. 이러한 근사 방법은 계산이 매우 빠르고 어느 정도의 정확도를 보장한다.

cf)백분위수 : 명확한 정의

데이터의 개수가 짝수라면 (n이 짝수), 앞선 정의에 따를 경우, 백분위수 값을 정하기 애매하다. 사실 아래 식을 만족하는 순서통계량 $x_{(j)}$ 와 $x_{(j+1)}$ 사이의 어떤 값도 택할 수 있다.

$$100 * \frac{j}{n} \leq P < 100 * \frac{j+1}{n}$$

보통은 백분위수는 아래 수식과 같은 가중평균이다.

$$\text{백분위수}(P) = (1-w)x_{(j)} + wx_{(j+1)}$$

가중치 w는 0과 1사이의 값이다. 통계 SW마다 이 w를 선택하는 방법이 조금씩 다르다. R의 quantile 함수의 경우, 분위수를 계산하는 9가지 다른 방법들을 제공한다. 데이터 개수가 너무 작지만 않다면, 백분위수를 계산할 때 정확도를 걱정할 필요는 없다.

1.4.3 예제 주별 인구의 변위 측정

R의 기본 함수를 이용해, 표준편차, 사분위범위 (IQR), 중위절대편차(MAD)와 같은 주별 인구의 변위 추정값들을 계산해보자

```
> sd(state[["Population"]])
[1] 6848235
> IQR(state[["Population"]])
[1] 4847308
> mad(state[["Population"]])
[1] 3849870
```

표준편차는 MAD의 거의 두 배가 된다.(R에서는 기본적으로 MAD의 척도가 평균과 같은 척도를 갖도록 보정된다.) 표준편차는 특잇값에 민감하므로 이 결과는 놀라운 것이 아니다.

주요개념

- *분산과 표준편차는 가장 보편적으로 널리 사용되는 변위 측정 방법이다.*
- *이들 모두 특잇값에 민감하다.*
- *평균과 백분위수(분위수)에서의 절대편차 평균과 중간값을 구하는 것이 좀 더 로버스트하다.*

1.5 데이터 분포 탐색하기

지금까지 알아본 추정들은 모두 데이터의 위치 혹은 변이를 나타내기 위한 하나의 수치로 데이터를 요약하고 있다. 이와 더불어 데이터가 전반적으로 어떻게 분포하고 있는지 알아보는 것 역시 유용하다.

용어 정리

상자 그림: 투키가 데이터의 분포를 시각화하기 위한 간단한 방법으로 소개한 그림

도수분포표 : 어떤 구간^{interval}(빈^{bin})에 해당하는 수치 데이터 값들의 빈도를 나타내는 기록

히스토그램 : x축은 구간들을, y축은 빈도수를 나타내는 도수 테이블의 그림

밀도 그림 : 히스토그램을 부드러운 곡선으로 나타낸 그림, 커널밀도추정을 주로 사용한다.

1.5.1 백분위수와 상자그림

1.4.2절에서 어떻게 백분위수를 활용해 데이터의 흩어진 정도를 측정하는지 알아봤다. 마찬가지로 전체 분포를 알아보는 데에도 백분위수가 유용하다, 주로 사분위수 (25,50,75번째 백분위수) 나 십분위수 (10,20,30, $\cdots$, 90번쨰 백분위수)를 공식적으로 사용한다. 특히 백분위수는 분포의 꼬리 부분(외측범위)을 묘사하는 데 제격이다. 대중문화에서는 상위 90번째 백분위수에 있는 부자를 지칭하기 위해 상위 1% 라는 말을 만들어내기도 했다.

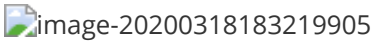
다음은 R로 사분위수를 표현한 방식이다.

```
> quantile(state[["Murder.Rate"]], p=c(.05, .25, .5, .75, .95))
   5%   25%   50%   75%   95%
1.600 2.425 4.000 5.550 6.510
```

5% 백분위수는 1.6에 불과한 반면, 95% 백분위수는 6.51에 달하는 등 약간의 변동폭이 있긴 하지만 중간값은 10만 명당 4건의 살인이 있다고 알려준다.

투키에 의해 처음 소개된 상자그림은 이 백분위수를 이용해 분산을 쉽게 시각화 하는 방법이다.

다음은 R을 이용해 만든 주별 인구 분포를 나타내는 상자그림이다.



```
boxplot(state[["Population"]]/1000000, ylab="Population (millions)")
```

상자부분의 위쪽과 아래쪽은 각각 75%, 25% 백분위수를 나타낸다. 중간값은 상자 안에 있는 굵은 수평선으로 표시한다. 구레나루처럼 위아래로 나 있는 점선이 바로 수염으로 데이터 전체의 범위 나타내주는 위 아래 선들과 연결되어 있다. 다양한 종류의 상자 그림들이 존재한다. R의 boxplot 함수 문서를 참고하자. 기본 설정상, R함수는 수염 부분이 사분위범위의 1.5배 이상 더 멀리 나가지 않도록 한다.(이것은SW마다 다를 수 있다). 수염 부분 보다 더 바깥쪽에 위치한 데이터는 각자 하나의 점으로 표시한다.

```
## Code for Figure 2
png(filename=file.path(PSDS_PATH, "figures", "psds_0102.png"), width = 3,
height=4, units='in', res=300)
?par
par(mar=c(0,4,0,0)+.1)
#par(mar)
#A numerical vector of the form c(bottom, left, top, right) which gives the
number of lines of margin to be specified on the four sides of the plot. The
default is c(5, 4, 4, 2) + 0.1.
boxplot(state[["Population"]]/1000000, ylab="Population (millions)")
dev.off()
```

1.5.2 도수분포표와 히스토그램

도수분포표는 변수의 범위를 동일한 크기의 구간으로 나눈 다음 , 각 구간마다 몇개의 변수 값이 존재하는지를 보여주기 위해 사용된다.

```
breaks <- seq(from=min(state[["Population"]]), to=max(state[["Population"]]),
length=11)
?cut
pop_freq <- cut(state[["Population"]], breaks=breaks, right=TRUE, include.lowest
= TRUE)
state['PopFreq'] <- pop_freq
table(pop_freq)
```

```
pop_freq
[5.64e+05,4.23e+06]  (4.23e+06,7.9e+06]
                 24                   14
 (7.9e+06,1.16e+07] (1.16e+07,1.52e+07]
                  6                    2
(1.52e+07,1.89e+07] (1.89e+07,2.26e+07]
                  1                    1
(2.26e+07,2.62e+07] (2.62e+07,2.99e+07]
                  1                    0
(2.99e+07,3.36e+07] (3.36e+07,3.73e+07]
                  0                    1
```

cf)데이터 가공을 위한 cut함수 활용

- 기능 : 연속형 변수를 기준에 따라 분할하여 범주화

- 형태 : cut(data, breaks, labels = NULL, include.lowest = FALSE, right = TRUE...)

- 내부인자

  1) breaks : 데이터 분할 기준

  2) labels : 분할된 데이터의 명칭

  3) include.lowest : 좌측 데이터 기준 포함 여부

  default값은 FALSE

if) right = FALSE일때,

① include.lowest = TRUE, a <= x < b

② include.lowest = FALSE, a < x < b

4) right : 우측 데이터 기준 포함 여부

default값은 TRUE

if) include.lowest = FALSE일때,

① right = TRUE, a < x <= b

② right = FALSE, a < x < b

[출처] [R함수] 데이터 가공*cut 함수*연속형 데이터 범주화 | **작성자** 쿠쓸

Note 도수분포표와 백분위수 모두 구간을 나눠서 데이터를 살펴보는 접근 방법이다. 일반적으로, 사분위수와 십분위수는 각 구간에 같은 수의 데이터가 포함되도록, 즉 서로 크기가 다르게 구간을 나누는 것이라고 할 수 있다. 반면에 도수분포표는 구간의 크기가 같도록, 즉 구간 안에 다른 개수의 데이터가 오도록 한다고 볼 수 있다.

히스토그램은 바로 이 도수분포표를 시각화하는 방법이라고 할 수 있다. x축에는 구간들을 표시하고 y축에는 해당 구간별 데이터의 개수를 표시한다. 표에 대한 히스토그램을 그리기 위해, hist 함수와 breaks 변수를 사용한다.

```
hist(state[["Murder.Rate"]], freq=FALSE )
```

히스토그램을 다음과 같은 정보들을 담고 있다.

- 그래드에 빈 구간들이 있을 수 있다.
- 구간은 동일한 크기를 갖는다.
- 구간의 수(혹은 구간의 크기)는 사용자가 결정할 수 있다.
- 빈 구간이 있지 않은 이상, 막대 사이는 공간 없이 서로 붙어 있다.

TIP 통계학에서 말하는 모멘트

통계학 이론에서, 위치와 변이는 각각 분포의 일차 및 이차 적률(모멘트)이라고 한다. 사차 모멘트는 각각 왜도, 첨도라고 부른다.

왜도는 데이터가 큰 값이나 작은 값 쪽으로 얼마나 비스듬히 쏠려 있는지를 나타내고,

첨도는 데이터가 극단값을 갖는 경향성을 나타낸다. 보통은 이러한 모멘트 값들을 직접 구하기보다는 위의 히스토그램과 같이 시각화해서 직접 확인한다.

1.5.3 밀도추정

히스토그램과 관련한 밀도 그림에 대해 알아보자. 밀도 그림은 데이터의 분포를 연속된 선으로 보여준다. 다시 말해서 좀 더 부드러운 히스토그램이라고 생각해볼 수 있겠다. **커널밀도추정**을 통해 데이터로부터 직접 계산한다. 다음 그림은 히스토그램에 겹쳐지게 밀도추정 결과를 표시하고 있다.

```
hist(state[["Murder.Rate"]], freq=F )
lines(density(state[["Murder.Rate"]]), lwd=3, col="blue")
```

주요 개념

- 도수 히스토그램은 y축에 횟수를 x축에 변수 값들을 표시하고 한눈에 데이터의 분포를 볼 수 있다.

- 도수분포표는 히스토그램에 보이는 횟수들을 표 형태로 나타낸 것이다.
- 상자그림에서 상자의 위 아래 부분은 각각 75%,25% 백분위수를 의미하며, 이것 역시 데이터의 분포를 한눈에 파악할 수 있도록 돕는다. 주로 분포들을 서로 비교하기 위해 사용한다.
- 밀도 그림은 히스토그램의 부드러운 버전이라고 할 수 있다. 데이터로부터 이 그림을 얻기 위해서는 어떤 함수를 구해야 하는데 여러 가지 가능한 추정 방법이 있다.

1.6 이진 데이터와 범주 데이터 탐색하기

범주형 데이터에서는 간단한 비율이나 퍼센트를 이용해 데이터에 관해 논할 수 있다.

용어정리

- ***최빈값: 데이터에서 가장 자주 등장하는 범주 혹은 값***
- ***기댓값: 범주에 해당하는 어떤 수치가 있을 때, 범주의 출현 확률에 따른 평균***
- ***막대도표: 각 범주의 빈도수 혹은 비율을 막대로 나타낸 그림***
- ***파이그림: 각 범주의 빈도수 혹은 비율을 원의 부채꼴 모양으로 나타낸 그림***

이진변수나 혹은 범주가 몇 개 안되는 범주형 변수를 분석하는 것은 그렇게 어렵지 않다. 예를 들어 이진변수의 경우 1과 같이 중요한 범주의 비율이 어느 정도 되는지 알아보면 된다. 예를 들어 2010년 이후 댈러스 포트워스 공항에서 항공기 운행이 지연된 원인의 퍼센트 비율을 보여준다. 운행 지연의 원인은 다음과 같은 요인들로 분류할 수 있다.

코드는 다음과 같다.

```
barplot(as.matrix(dfw)/4, cex.axis = 0.8, cex.names = 0.7)
```

막대도표는 히스토그램과 매우 유사하다는 점을 기억하자. 다만 닥대도표에서 x축은 각 요인변수의 서로 다른 범주들을 나타내는 반면, 히스토그램의 x 축은 수치적으로 나타낼 수 있는 하나의 변수값을 의미한다. 히스토그램에서 막대들은 일반적으로 서로 붙어 있고, 중간에 틈이 있다는 것은 그 부분에 해당하는 값들이 존재하지 않는다는 것을 의미한다. 이와 달리, 막대도표에서 막대들은 서로 떨어져있다.

막대도표 대신 파이그림을 사용하지도 하지만 통계학자나 데이터 시각화 전문가들은 파이그림이 시각적으로 효과적이지 않다는 이유로 잘 사용하지 않는다.

용어 정리

- ***상관계수: 수치적 변수들 간에 어떤 관계가 있는지를 나타내기 위해 사용되는 측정량(-1~+1)***
- ***상관행렬: 행과 열이 변수들을 의미하는 표를 말하며, 각 셀은 그 행과 열에 해당하는 변수들 간의 상관관계를 의미한다.***
- ***산점도: x,y축이 서로 다른 두 개의 변수들 나타내는 도표***

아래 두 변수를 고려해보자. 모두 작은 값에서 큰 값으로 점차 커지는 식으로 상관관계를 갖고 있다.

- v1 :{1,2,3}
- v2 :{4,5,6}

벡터곱의 합은 4+10+18 = 32이다. 이 값들의 순서를 섞어서 다시 계산한다면 벡터곱의 합은 절대 32를 넘을 수 없다. 이 곱의 합을 측정량으로 사용할 수 있다. 즉 32라는 합을 랜덤으로 섞었을 때 나오는 값들과 비교해볼 수 있을 것이다. 하지만 이렇게 얻은 값들은 재표본분포에 대한 레퍼런스로써의 의미밖에는 없다.

이러한 방법보다는 **상관계수(피어슨 상관계수)**라는 표준화된 방식이 훨씬 더 유용하다. 두 변수 사이의 상관관계를 항상 같은 척도에 놓고 추정하는 것이다. 피어슨 상관계수를 계산하기 위해, 변수 1과 변수 2 각각의 평균으로부터의 편차들을 서로 곱한 값들의 평균을 각 변수의 표준편차 곱으로 나눠준다.

$$Corr(X,Y) = \frac{Cov(X,Y)}{\sqrt{Var(X)Var(Y)}}$$

$$r = \sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})/(n-1)s_x s_y$$

$$= \sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})/\sqrt{(\sum_{i=1}^{n}(x_i - \overline{x})^2)(\sum_{i=1}^{n}(y_i - \overline{y})^2)}$$

R의 corrplot 패키지를 사용하여 상관계수 표를 그려보자.

```
etfs <- sp500_px[row.names(sp500_px)>"2012-07-01",
                 sp500_sym[sp500_sym$sector=="etf", 'symbol']]
library(corrplot)
corrplot(cor(etfs), method = "ellipse")
```

평균과 표준편차와 같이, 상관계수는 데이터의 특잇값에 민감하다. 이러한 클래식한 상관계수를 대체할 수 있는 로버스트한 방법들을 제공하는 SW들이 있다. 예를 들어 R의 cor함수 같이, 절사평균을 계산할 때아 같은 trim이라는 인수를 제공한다.

Note 상관계수를 추정하는 다른 방법

오래전부터 통계학자들은 스피어맨의 로(rho)나 켄들의 타우(tau)와 같은 다른 형태의 상관계수들을 만들어냈다. 이들은 데이터의 순위를 기초로 하는 상관계수이다. 값 자체보다 값의 순위를 이용하기에 이러한 추정법은 특잇값에 좀 더 로버스트하며 비선형 관계도 다룰 수 있다. 하지만 데이터 과학자들은 일반적으로 피어슨 상관계수, 혹은 이것의 로버스트한 다른 버전들을 탐색 분석에 주로 사용한다. 순위 기반 추정법은 보통 데이터의 크기가 작고 어떤 특별한 가설검정이 필요할 때 주로 사용된다.

1.7.1 산점도

두 변수 사이의 관계를 시각화하는 가장 기본적인 방법은 **산점도**를 그려보는 것이다. x,y축은 각각의 변수들을 의미하고 그래프의 각 점은 하나의 레코드를 의미한다. R로 plot예제를 사용해보자.

```
plot(telecom$T, telecom$VZ, xlab="T", ylab="VZ")
```

주요개념

- 상관계수는 두 변수 사이에 서로 어떤 관계가 있는지를 측정한다

1.8 두 개 이상의 변수 탐색하기

평균과 분산과 같이 익숙한 추정값들은 한 번에 하나의 변수를 다룬다(**일변량분석**). 상관분석은 두 변수(**이변량분석**)를 비교할 때 중요한 방법이다. 이번 절에서는 이에 관한 추정법과 도표를 살펴보고 셋 이상의 변수(**다변량 분석**)를 다루는 법도 살펴보겠다.

일변량분석과 마찬가지로, 이변량분석 역시 요약통계를 계산하고 시각화하는 것을 기본으로 한다. 이변량분석 혹은 다변량분석의 형태는 데이터가 수치형인지 아니면 범주형인지, 데이터의 특성에 따라 달라진다.

용어 정리

- ***분할표: 두 가지 이상의 범주형 변수의 빈도수를 기록한 표***
- ***육각형 구간: 두 변수를 육각형 모양의 구간으로 나눈 그림***

- *동고 도표: 지도상에 같은 높이의 지점을 등고선으로 나타내는 것처럼, 두 변수의 밀도를 등고선으로 표시한 도표*
- *바이올린 도표: 상자그림과 비슷하지만 밀도추정을 함께 보여준다.*

1.8.1 육각형 구간과 등고선(수치형 변수 대 수치형 변수를 시각화)

산점도는 데이터의 개수가 상대적으로 적을 때는 괜찮다. 하지만 수십, 수백만의 레코드를 나타내기에는 산점도의 점들이 너무 밀집되어 알아보기 어렵다. 따라서 이러한 관계를 나타내는 다른 방법이 필요하다. 예시로 워싱턴 주에 위치한 킹 카운티의 주택 시설에 대한 과세 평가 금액 정보를 담고 있는 데이터 집합 kc_tax를 살펴보고자한다. 데이터의 주요 부분에 집중하기 위해, 아주 비싸거나, 너무 작은 혹은 너무 큰 주택들은 subset함수를 이용해서 제거한다.

```
kc_tax0 <- subset(kc_tax, TaxAssessedValue < 750000 & SqFtTotLiving>100 &
                  SqFtTotLiving<3500)
nrow(kc_tax0)
[1] 432693
```

```
ggplot(kc_tax0, (aes(x=SqFtTotLiving, y=TaxAssessedValue))) +
  stat_binhex(colour="white") +
  theme_bw() +
  scale_fill_gradient(low="white", high="black") +
  labs(x="Finished Square Feet", y="Tax Assessed Value")
```

두 수치형 변수 사이의 관계를 시각화하기 위해 산점도 위에 등고선을 사용한다. 이 등고선은 이 두 변수로 이루어진 지형에서의 등고선을 의미한다. 등고선 위의 점들은 밀도가 같다. '꼭대기' 쪽으로 갈수록 밀도는 높아진다. 이는 ggplot2의 geom_density2d 함수를 사용해서 그릴 수 있다.

```
ggplot(kc_tax0, aes(SqFtTotLiving, TaxAssessedValue))+
  theme_bw() +
  geom_point( alpha=0.1) +
  geom_density2d(colour="white") +
  labs(x="Finished Square Feet", y="Tax Assessed Value")
```

두 수치형 변수의 관계를 나타내는 다른 도표로 **히트맵**이 있다. 히트맵, 육각 구간, **등고 도표** 모두 이차원상의 밀도를 시각화하는 데 사용된다. 히스토그램이나 밀도 그림과 유사성을 찾을 수 있다.

1.8.2 범주형 변수 대 범주형 변수

분할표는 두 범주형 변수를 요약하는 데 효과적인 방법이다. 범주별 빈도수를 기록한 표를 뜻한다.

```
CrossTable(lc_loans$grade, lc_loans$status,
                   prop.c=FALSE, prop.chisq=FALSE, prop.t=FALSE)
```

```
          lc_loans$status
lc_ln$    Chrg O   Currnt   Flly P     Late    Total
-------------------------------------------------------
A           1562    50051    20408      469    72490
           0.022    0.690    0.282    0.006    0.161
```

```
-------------------------------------------------------
B            5302    93852    31160     2056   132370
           0.040    0.709    0.235    0.016    0.294
-------------------------------------------------------
C            6023    88928    23147     2777   120875
           0.050    0.736    0.191    0.023    0.268
-------------------------------------------------------
D            5007    53281    13681     2308    74277
           0.067    0.717    0.184    0.031    0.165
-------------------------------------------------------
E            2842    24639     5949     1374    34804
           0.082    0.708    0.171    0.039    0.077
-------------------------------------------------------
F            1526     8444     2328      606    12904
           0.118    0.654    0.180    0.047    0.029
-------------------------------------------------------
G             409     1990      643      199     3241
           0.126    0.614    0.198    0.061    0.007
-------------------------------------------------------
Total       22671   321185    97316     9789   450961
=======================================================
>
```

1.8.3 범주형 변수 대 범주형 변수

상자그림은 범주형 변수에 따라 분류된 수치형 변수의 분포를 시각화 하여 비교하는 간단한 방법이다. barplot를 이용하여 알아보자.

예를 들어 항공사별 비행 지연 정도를 비교하고 싶다고 하자. 다음 그림은 항공사별 비행 지연 정도를 비교하려 한다.

```
boxplot(pct_carrier_delay ~ airline, data=airline_stats, ylim=c(0,50))
```

바이올린 도표는 상자그림을 보완한 형태로, y측을 따라 밀도추정 결과를 동시에 시각화한다. 밀도 분포 모양을 좌우대칭으로 서로 겹쳐지도록 해놓고 보면 바이올린을 닮은 모양이 된다. 바이올린 도표의 장점은 바로 상자그림에서는 보이지 않는 데이터의 분포를 볼 수 있다는 점이다. 한편, 상자그림은 데이터의 특잇값들을 좀 더 명확하게 보여준다. ggplot2에서 violin함수를 이용해 다음과 같이 바이올린 도표를 생성할 수 있다.

```
ggplot(data=airline_stats, aes(airline, pct_carrier_delay))  +
  ylim(0, 50) +
  geom_violin() +
  labs(x="", y="Daily % of Delayed Flights")
```

이 바이올린 도표를 통해 알래스카 항공, 그리고 그보다는 적지만 델타항공이 거의 0근처에 데이터가 0근처에 데이터가 집중되어 있는 것을 볼 수 있다. 상자그림에서는 명확하게 드러나지 않는 현상이었다. geom_boxplot 함수를 추가한다면 바이올린 도표에 상자그림을 결합할 수도 있다.

(색상을 추가하는 것도 좋다.)

1.8.4 다변수 시각화하기

조건화라는 개념을 통해 두 변수 비교용 도표(산점도, 육각형 구간, 상자그림)를 더 여러 변수를 비교하는 용도로 확장하여 활용할 수도 있다. 일례로, 주택 크기와 과세 평가액 간의 관계를 보여줬던 그림으로 가보자. 단위 넓이(제곱피트) 당 더 높은 과세 평가 금액을 보였던 한 무리의 주택들을 알 수 있었다. 좀 더 깊이 들어가기 위해 다음그림을 제시하고자 한다. 지리적 요인을 살펴보기 위해 우편번호별로 데이터를 묶어서 도식화했다. 이제 그림이 좀 더 명확해졌다.

```
ggplot(subset(kc_tax0, ZipCode %in% c(98188, 98105, 98108, 98126)),
         aes(x=SqFtTotLiving, y=TaxAssessedValue)) +
  stat_binhex(colour="white") +
  theme_bw() +
  scale_fill_gradient( low="white", high="black") +
  labs(x="Finished Square Feet", y="Tax Assessed Value") +
  facet_wrap("ZipCode")
```

어떤 우편번호(98105,98126)에서의 평가액이 다른 두 군데(98108,98188)보다 훨씬 더 높다는 것을 볼 수 있다. 이러한 불균형이 위의 산점도 그림보다 보다 상관관계를 명확히 보여준다.

조건화 변수라는 개념은 이후 lattifce, ggplot2 같은 R패키지, 그리고 Seaborn, Bokeh 같은 파이썬 패키지 등 다양한 최신 그래픽스 시스템에 영향을 주었다. 또한 조건화 변수는 태블로나 스폿파이어 같은 비즈니스 지능형 플랫폼에도 없어서는 안 될 중요한 요소가 되었다. 최근 어마어마한 계산 능력의 등장과 더불어, 최신 시각화 플랫폼들은 탐색적 데이터 분석의 초라했던 시작에 비해 엄청난 발전을 이뤘다. 하지만 다년간 개발되어온 주요 개념과 방법들은 여전히 이러한 시스템들의 기반이 된다.

주요 개념

- ***육각형 구간이나 등고선 도표는 데이터의 방대한 양에 압도당하지 않으면서, 한 번에 두 수치형 변수를 시각적으로 검토하기 위해 유용한 도구이다.***
- ***분할표는 두 범주형 변수의 도수를 확인하기 위한 표준 방법이다.***
- ***상자그림과 바이올린 도표는 범주형 변수와 수치형 변수 간의 관계를 도식화하기 위한 도구이다.***

1.9 마치며

탐색적 데이터 분석(EDA)의 개발과 함계, 통계학은 데이터 과학이라는 분야로 가는 초석을 놓았다. EDA의 핵심은 바로, 데이터를 다루는 모든 프로젝트에서 가장 우선적이며 가장 중요한 과정이 데이터를 들여다보는 것에 있다. 데이터를 요약하고 시각화하는 것을 통해, 프로젝트에 대한 가치 있는 통잘과 이해를 얻게 된다. 이 장에서는 위치와 변이 추정 같은 간단한 계측에서부터 조건화 변수 그래프 시각화와 같이 다변량 간의 관계를 살펴보기 위한 다양한 시각화 기법까지 살펴봤다.

오픈소스 커뮤니티에서 개발된 다양ㅅ한 방법과 기술들이 R과 파이썬이라는 언어가 갖는 확장성과 결합되어 데이터 분석을 위한 셀 수 없이 많은 방법들이 만들어지고 있다. 탐색적 분석은 모든 데이터 과학 프로젝트의 초석이 되어야 한다.