

# **비즈니스를 위한 데이터마이닝**

가톨릭대학교 경영학과

이홍주

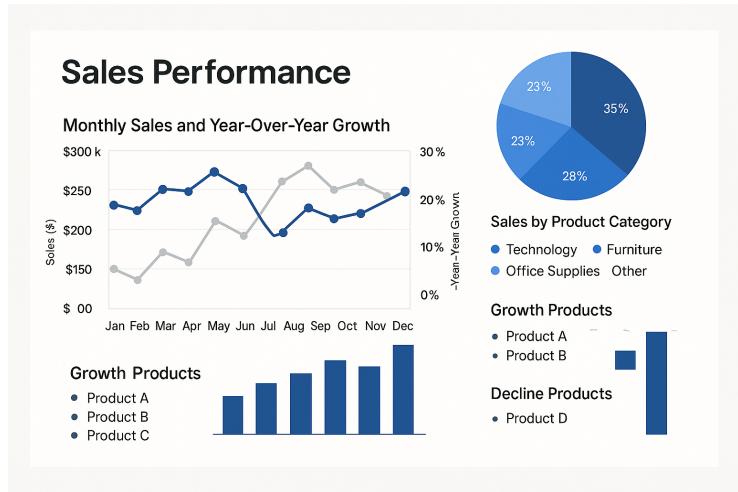
# 시각화의 목적

## 1. 프리젠테이션 (보고 및 스토리텔링)

- 분석 결과를 명확하고 설득력 있게 전달
- 예) 경영진 보고용, 마케팅 캠페인 효과 분석 발표

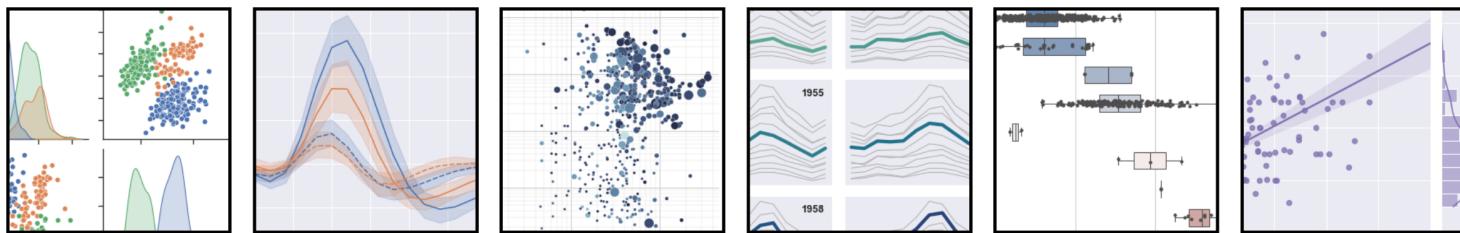
## 2. 데이터 탐색 및 사전 분석

- 예측 분석 관련 데이터 시각화
- 데이터 구조 이해, 데이터 수정(예상치 못한 격차 또는 오류 값 식별), 이상치 식별, 기본 패턴 발견(변수 간의 상관관계, 클러스터), 흥미로운 질문을 도출하기 위한 자유 탐색



# 시각화 라이브러리

- Matplotlib 가장 기본! 수학적 시각화
- Pandas 기본
- Seaborn 조금 더 예쁘



# 보스턴 주택 데이터

지역 내 주택들의 중앙값!  
개개인 ×

**표 3-1** 보스턴 주택 데이터셋의 변수 설명

CRIM	범죄율
ZN	25,000제곱피트 이상의 부지에 대해 구획된 주거용 토지의 비율
INDUS	비소매업이 차지하는 토지 비율
CHAS	찰스강 인접 여부(1=인접, 0=비인접)
nox	10ppm당 일산화질소
RM	주택의 평균 방 개수
AGE	1940년 이전에 건축된 주택에 사는 비율
DIS	보스턴 5대 상업 지구와의 거리
RAD	고속도로 진입 용이성 정도
TAX	재산세율(10,000달러당)
PTRATIO	시town별 학생 대 교사 비율
LSTAT	저소득층 비율
MEDV	주택 가격의 중앙값(단위: 1,000달러)
CAT.MEDV	주택 가격의 중앙값이 3만 달러 이상인지 여부(1=이상, 0=미만)

# Amtrak 기차 이용 데이터

시각화된 데이터

	Month	Ridership	Date
0	01/01/1991	1708.917	1991-01-01
1	01/02/1991	1620.586	1991-02-01
2	01/03/1991	1972.715	1991-03-01
3	01/04/1991	1811.665	1991-04-01
4	01/05/1991	1974.964	1991-05-01
...	...	...	...
154	01/11/2003	2076.054	2003-11-01
155	01/12/2003	2140.677	2003-12-01
156	01/01/2004	1831.508	2004-01-01
157	01/02/2004	1838.006	2004-02-01
158	01/03/2004	2132.446	2004-03-01

# 기본 차트: 막대 그래프, 선 그래프, 산점도

- 산점도 Scatterplot



- 수치형 변수 간의 관계를 보여주는 데 사용
- 비지도 학습에서 두 가지 수치형 변수 간의 정보 중복이나 군집 발견과 같은 연관성을 밝히는 데 도움이 됨

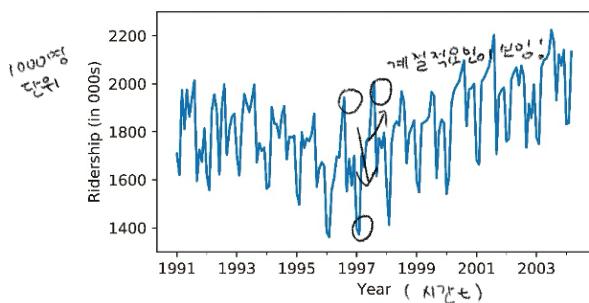
- 막대 그래프 bar chart

- 평균, 개수, 비율과 같은 단일 통계값의 그룹별 비교에 유용
- 막대 높이는 통계값을 나타내고, 각 막대는 그룹을 표시하며, 각 막대의 높이(수평 막대의 경우에는 길이)는 변수 값을 나타냄

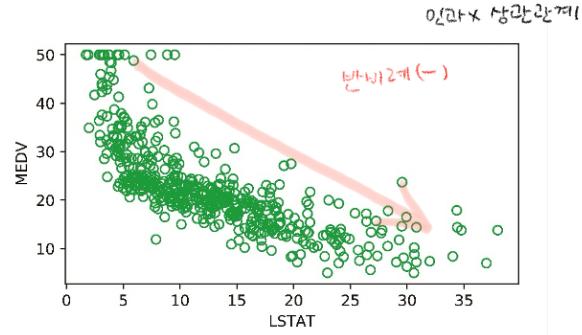
- 선 그래프 line plot

- 주로 시계열을 보여주는 데 사용
- 그래프를 그리기 위한 시간 프레임의 크기는 시간 척도와 마찬가지로 예측 작업의 규모와 데이터의 속성에 따라 달라짐

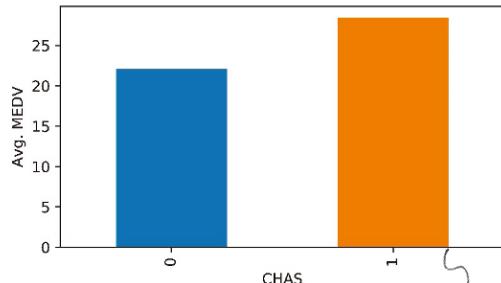
# 기본 차트: 막대 그래프, 선 그래프, 산점도



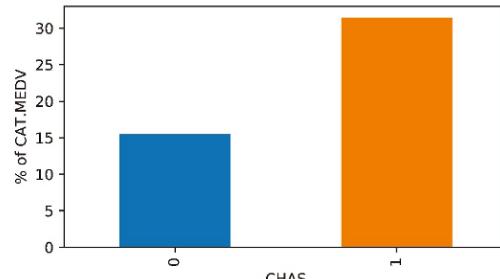
(a) 선그래프



(b) 산점도



(c) 수치형 변수에 대한 막대그래프



(d) 범주형 변수에 대한 막대그래프

그림 3-1 기본 차트

# 데이터 불러오기

```
## Load, convert Amtrak data for time series analysis  
Amtrak_df = pd.read_csv('../data/Amtrak.csv')  
Amtrak_df['Date'] = pd.to_datetime(Amtrak_df.Month,  
format='%d/%m/%Y')  
ridership_ts = pd.Series(Amtrak_df.Ridership.values,  
index=Amtrak_df.Date)
```

이전에 만든 날짜가 column에 있으므로

## ## Boston housing data

```
housing_df = pd.read_csv('../data/BostonHousing.csv')  
housing_df = housing_df.rename(columns={'CAT_MEDV':  
'CAT_MEDV'})
```

column 이름에 .(점)이 있어서

ex. Amtrak.a 같은 식으로

부른다면!

Amtrak.a

Amtrak['a']

이런식으로 불러올 때!

# 산점도

**Using pandas:**

```
## scatter plot with axes names  
housing_df.plot.scatter(x='LSTAT', y='MEDV', legend=False)
```

**Using matplotlib:**

```
## Set the color of points and draw as open circles.  
plt.scatter(housing_df.LSTAT, housing_df.MEDV, color='C2',  
            facecolor='none')  
plt.xlabel('LSTAT')  
plt.ylabel('MEDV')
```

# 막대 그래프

Using pandas:

```
# compute mean MEDV per CHAS = (0, 1)
ax = housing_df.groupby('CHAS').mean().MEDV.plot(kind='bar')
ax.set_ylabel('Avg. MEDV')
```

Using matplotlib:

```
# compute mean MEDV per CHAS = (0, 1)
dataForPlot = housing_df.groupby('CHAS').mean().MEDV
fig, ax = plt.subplots()
ax.bar(dataForPlot.index, dataForPlot, color=['C5', 'C1'])
ax.set_xticks([0, 1], False)
ax.set_xlabel('CHAS')
ax.set_ylabel('Avg. MEDV')
```

# 선 그래프

## Using pandas:

```
ridership_ts.plot(ylim=[1300, 2300], legend=False)  
plt.xlabel('Year') # set x-axis label  
plt.ylabel('Ridership (in 000s)') # set y-axis label
```

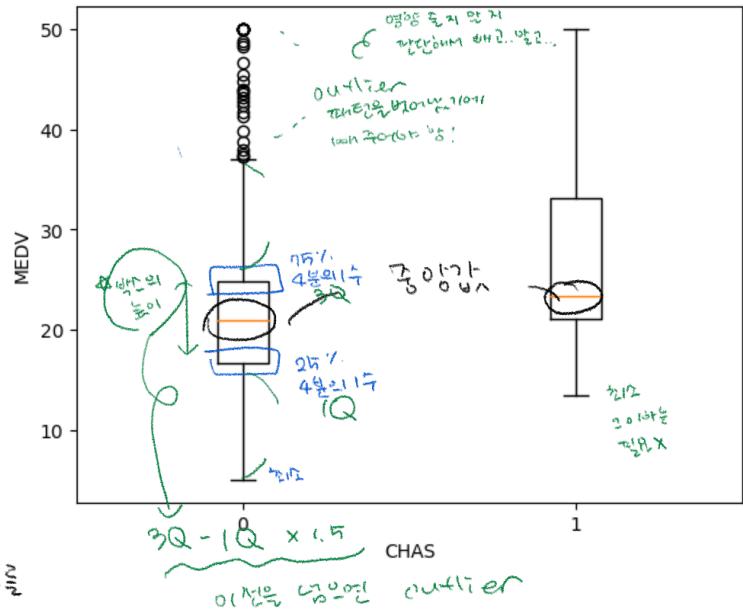
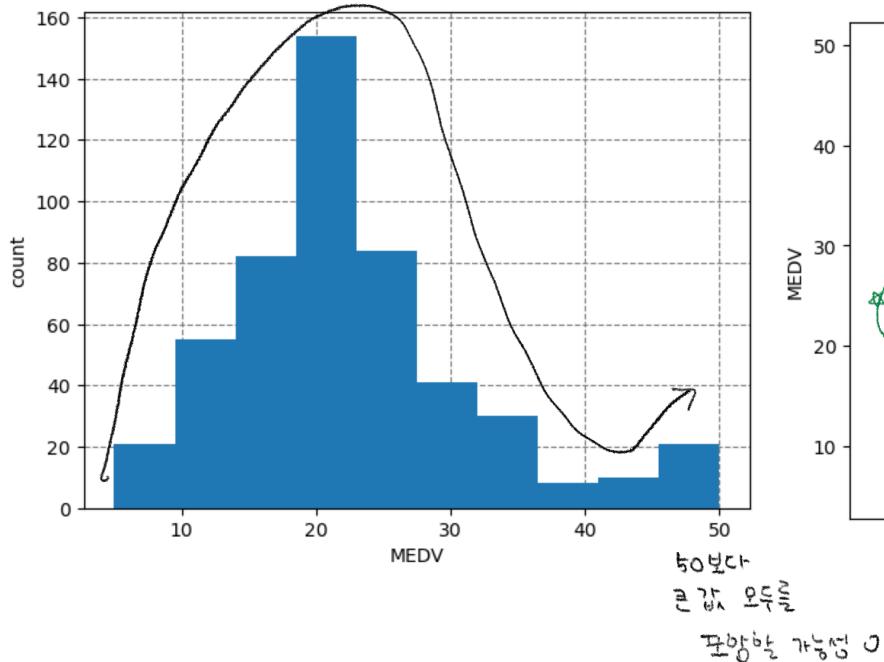
## Using matplotlib:

```
plt.plot(ridership_ts.index, ridership_ts)  
plt.xlabel('Year') # set x-axis label  
plt.ylabel('Ridership (in 000s)') # set y-axis label
```

# 분포도: 박스 플롯과 히스토그램

- 분포도
  - 수치형 변수의 전반적인 분포를 표시하며, 범위(or bins)내에 값이 얼마나 분포하는지 보여줌
  - 범주형의 경우 각 범주가 얼마나 등장하는지를 보여줌
- 박스 플롯
  - 나란히 생성해 하위 그룹끼리 비교하거나, 여러 개의 박스 플롯을 시간별로 생성함으로써 시간 변화에 따른 분포를 관찰할 수 있음
- 히스토그램
  - 일련의 수직 연결된 막대로 모든 x값의 빈도를 나타냄

# 분포도: 박스 플롯과 히스토그램



# 히스토그램

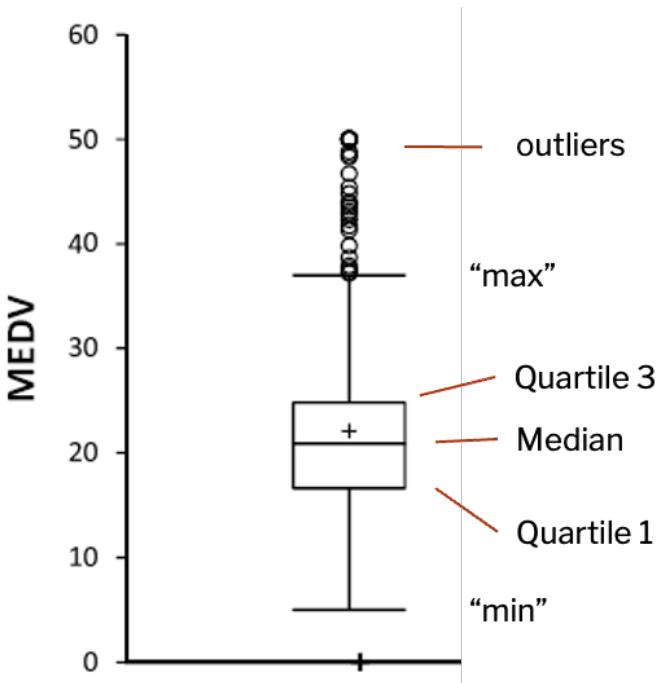
Using pandas:

```
ax = housing_df.MEDV.hist()  
ax.set_xlabel('MEDV')  
ax.set_ylabel('count')
```

Using matplotlib:

```
fig, ax = plt.subplots()  
ax.hist(housing_df.MEDV)  
ax.set_axisbelow(True) # Show the grid lines behind the histogram  
ax.grid(which='major', color='grey', linestyle='--')  
ax.set_xlabel('MEDV')  
ax.set_ylabel('count')
```

# 박스 플롯

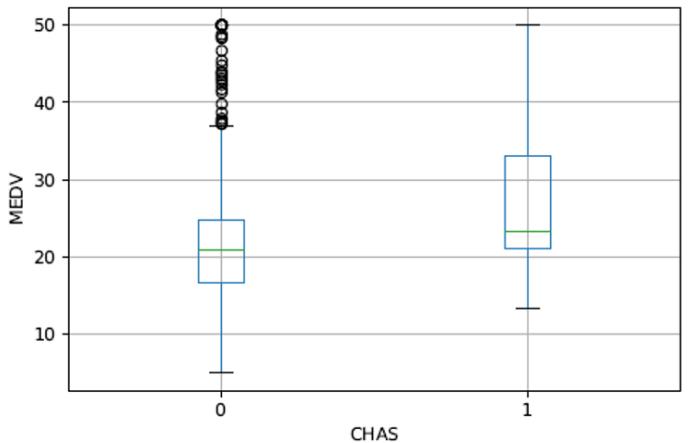


- 상위 이상치는  $Q3 + 1.5(Q3 - Q1)$ 보다 큰 값으로 정의됨
- “max” = maximum of non-outliers
- “min” = minimum of non-outliers
- 하위 이상치는  $Q1 - 1.5(Q3 - Q1)$ 보다 작은 값으로 정의됨
- 사용하는 시각화 라이브러리에 따라 다를 수 있음

# 박스 플롯

- 나란히 배치한 박스플롯은 하위 그룹들을 비교하는 데 유용함
- 찰스강(Charles river) 인근(1)에 위치한 주택은 그렇지 않은 곳(0)에 위치한 주택보다 가치가 더 높다.

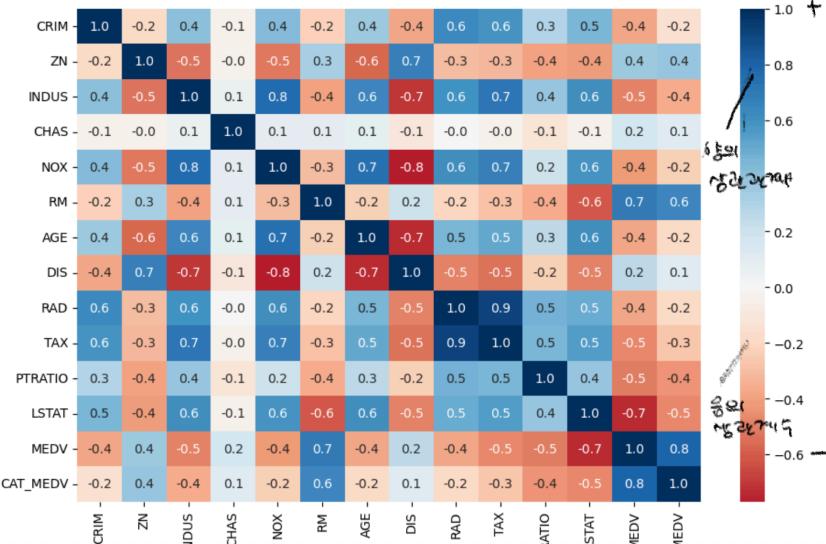
ax =  
housing\_df.boxplot(column='MEDV', by='CHAS')  
ax.set\_ylabel('MEDV')  
plt.suptitle('') # Suppress the  
titles  
plt.title('')  
  
하나의 두자를 가지고도  
△ 2~3개씩을  
그리는 경우 매우  
쓰임▲



# 히트맵 Heatmap

열기

- 히트맵의 셀들은 색이 정보를 포함하고 있음  
여기서는... 예전에 근데 줄이 그리지 않았던 것
- 다양한 영역에 사용될 수 있으나 데이터분석에서는 상관관계와 결측치 비율을 보여주는 경우에 많이 활용함  
-1 음수 | 양수
- 진하고 파랄수록 더 강한 양의 상관관계
- 진하고 빨갈수록 더 강한 음의 상관관계



모든 상관관계가 1일 때 다 넣을 필요가 있음을!

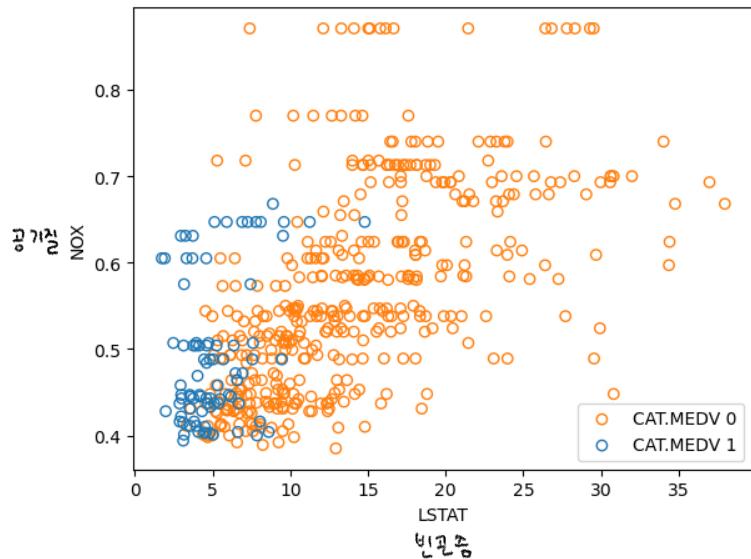
# 다차원 시각화 Multidimensional Visualization

- 한 변수 또는 두 변수간의 관계를 넘어 여러 변수의 관계를 시각화하는 것  
    → 항목이 어려워짐  
    사실 선정도 잘 그리지 × 다차원시각화 ↓
- 속성 변수 추가
  - 색상, 크기, 모양, 다중 패널, 애니메이션
- 차트 조절
  - 스케일링, 집계와 계층 구조, 확대/축소, 필터링
  - 추세선과 인-플롯 레이블
    - 데이터 옆에 레이블 표기
  - 인터랙티브 시각화

# 산점도에 색상 추가

```
fig, ax = plt.subplots()
for catValue, color in (0, 'C1'), (1,
    'C0'):
    subset_df =
        housing_df[housing_df.CAT_ME
        DV == catValue]
    ax.scatter(subset_df.LSTAT,
        subset_df.NOX, color='none',
        edgecolor=color)
ax.set_xlabel('LSTAT')
ax.set_ylabel('NOX')
ax.legend(["CAT.MEDV 0",
    "CAT.MEDV 1"])
plt.show()
```

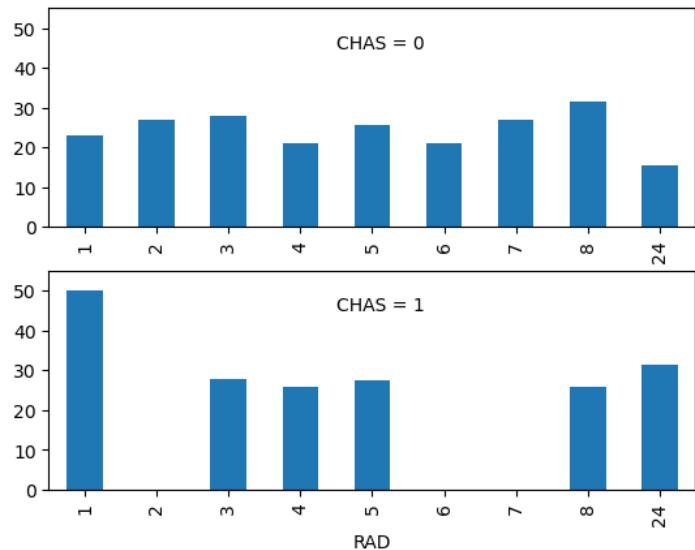
- NOX vs. LSTAT
  - low median value
  - high median value



# 막대 그래프 – 세가지 변수 사용

- y축 MEDV 주택 가격
- x축 RAD 고속도로 접근 면적
- 패널 분할: CHAS

다자원 ... 대체 3가지 변수...



# 행렬 산점도

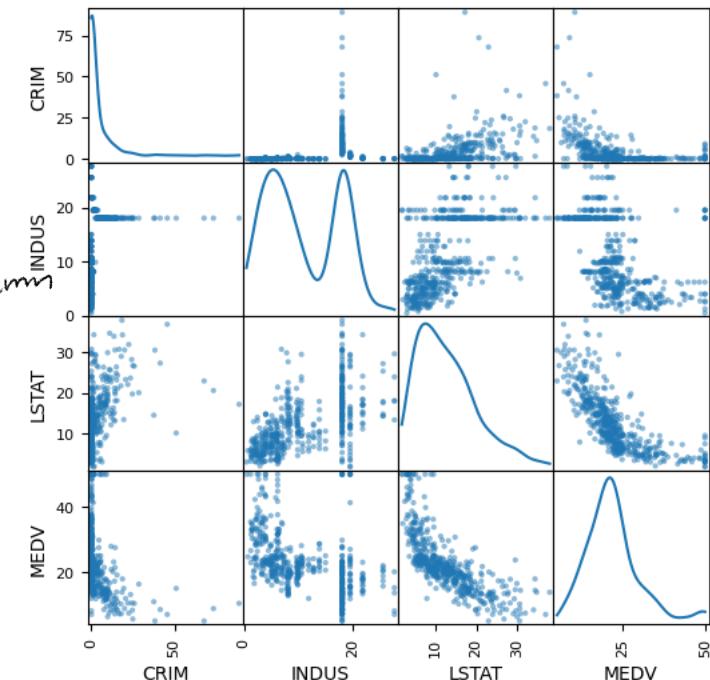
- scatter\_matrix()

산점도  
Scatter

- =

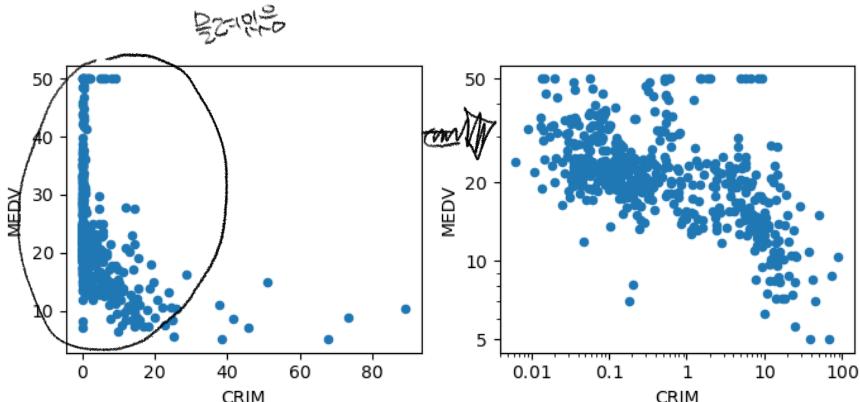
```
scatter_matrix(housing_df[  
    ['CRIM', 'INDUS', 'LSTAT',  
     'MEDV']], figsize=(6, 6),  
    diagonal='kde')
```

한국어로 번역한 텍스트

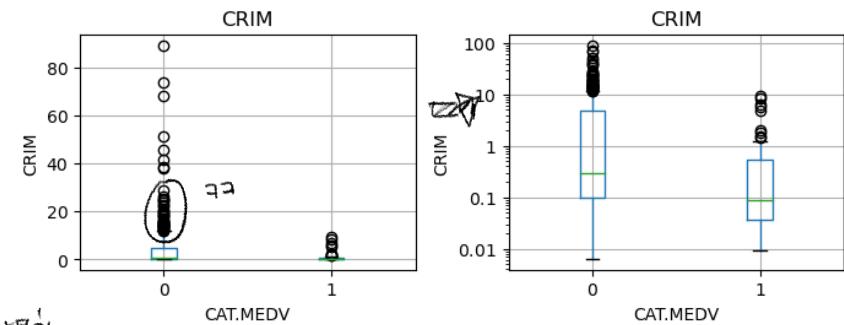


# 차트 조절: 스케일링 Scaling

- 디스플레이의 축척을 변경하면 변수 간의 관계를 부각시킬 수 있음  
*ex. log*
- 밀집 현상을 해소하고, 2개의 로그 스케일 변수 간 선형 관계(로그-로그 관계)를 드러내줌



```
# Regular scale
housing_df.plot.scatter(x='CRIM',
                         y='MEDV', ax=axes[0])
# log scale
ax =
    housing_df.plot.scatter(x='CRIM',
                            y='MEDV', logx=True,
                            logy=True, ax=axes[1])
```



# 차트 조절: 집계와 계층구조

- 집계의 수준을 변경하는 스케일링
  - 설계 막상
  - 6
- 시계열 데이터에서 흔히 사용하며 년/월/주/등의 수준에 따라 시각화함
  - 전방적인 대안 가능
  - 일별보다 더 낮은 것임  
단위(일수로) 소비력을 단순화
- 비시간성 변수는 지리적 위치(보스턴 주택 예제의 우편번호별 주택), 조직도(부서 또는 부문의 인력) 등 의미 있는 계층이 있다면 집계될 수 있음

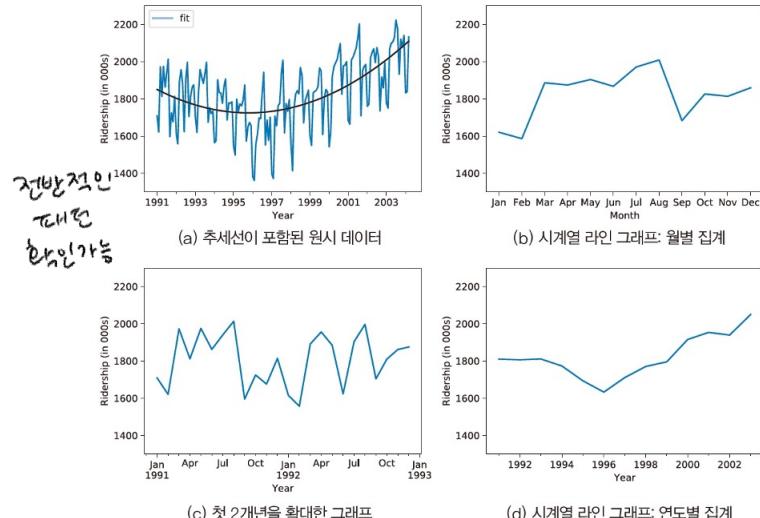
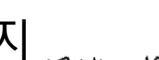
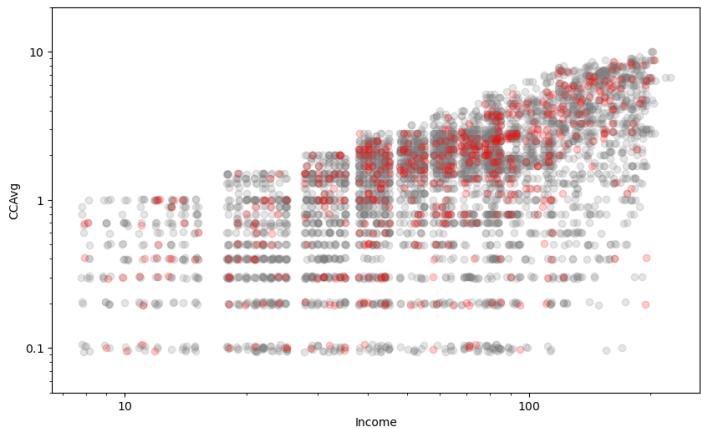


그림 3-9 철도 승객 시계열 데이터 집계

# 대용량 데이터 시각화

- 샘플링
- 표시 크기를 줄임  

- 표시 색의 투명도를 사용
- 다중 패널 사용
- 집계 사용
- 지터링 사용  
    -   
    - 
- 작은 양의 노이즈를 추가해  
    데이터가 겹치는 것을 방지  
    -   
    - 

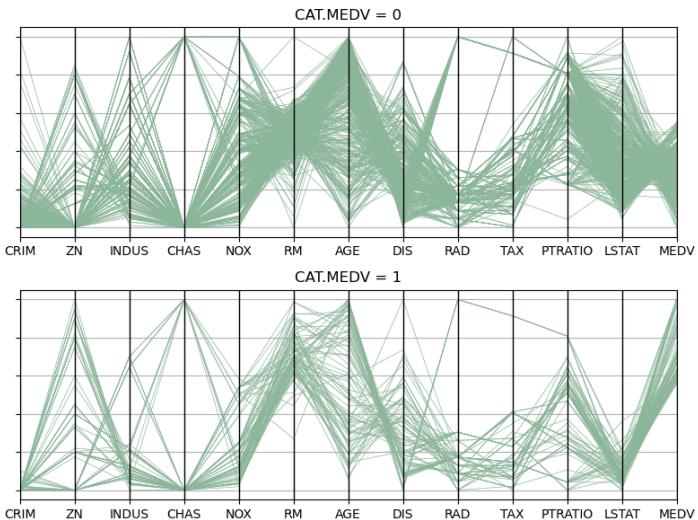


지터링  
겹침 방지  
색 투명도  
다른 것은 O일 수도

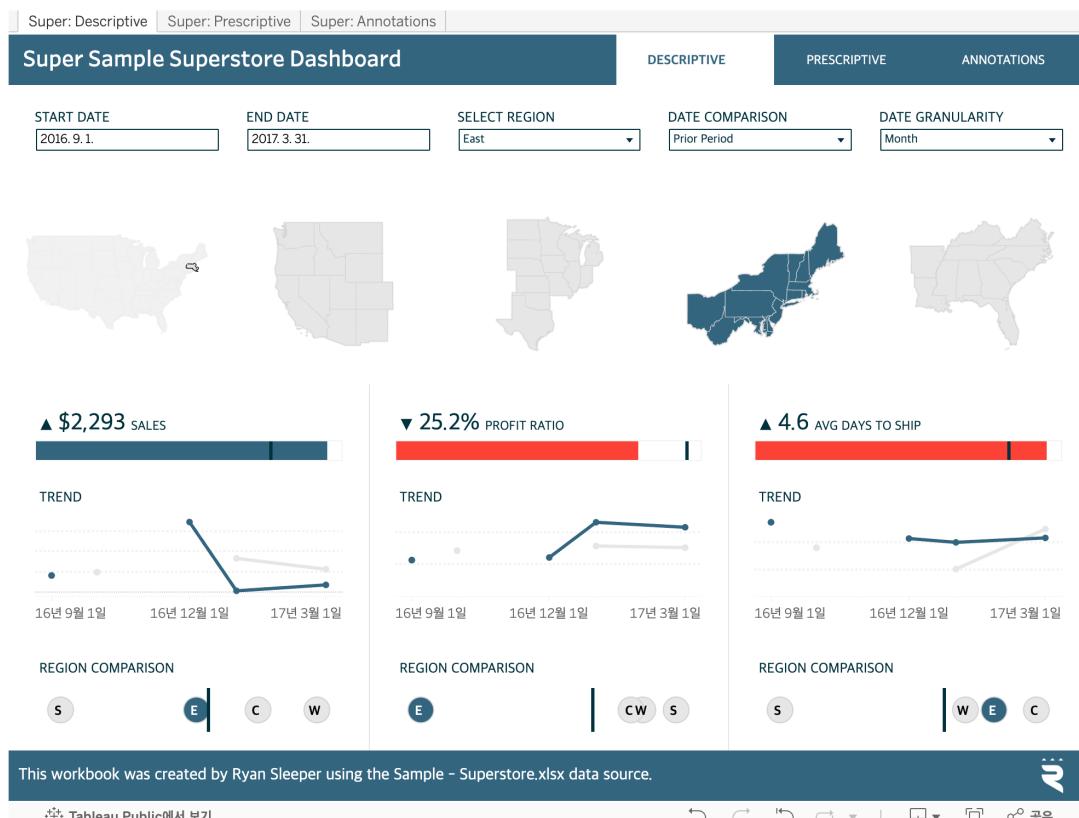
# 다변량 플롯

pf 그려보았지

- y축은 표준화
- x축은 여러 변수
- 다중 패널과 함께 사용시 두 집합의 차이 시각화



# 인터랙티브 시각화



<https://www.tableau.com/data-insights/dashboard-showcase/superstore>  
pw 28.41%