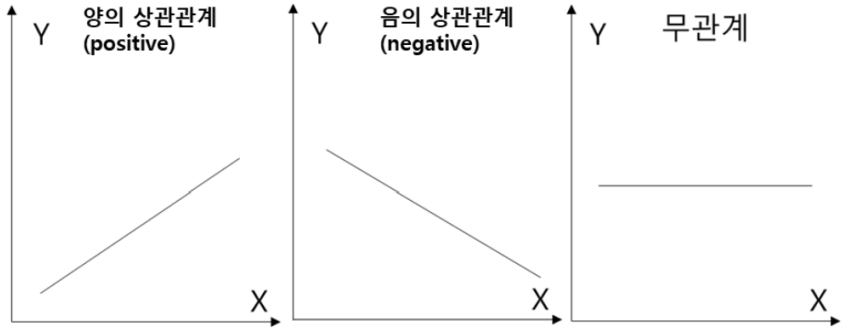
# Housing Price Data

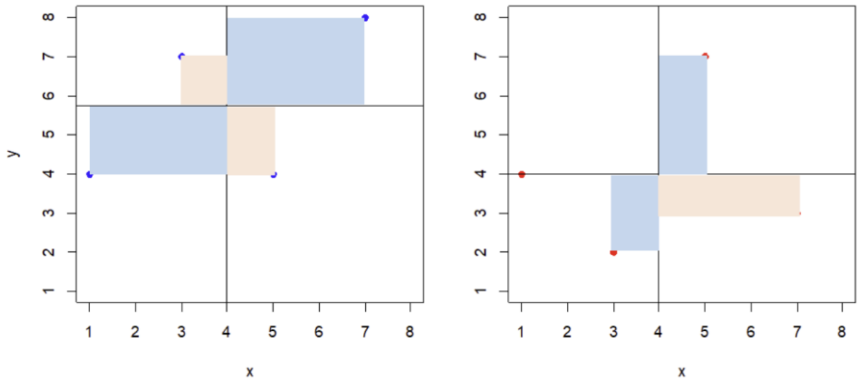
# Correlation

## 연속형 변수의 상관관계를 알 수 있다.

- 관계의 방향 (그래프)
- 관계의 강도 (공분산 값) → 데이터 단위에 따라 달라짐 → 피어슨 상관계수




[그림 1] 관계의 방향성




[그림 6] 공분산 계산

$$r_{x,y} = \frac{Cov(x,y)}{\sigma_x \sigma_y}$$

## 피어슨 상관계수 요구 조건

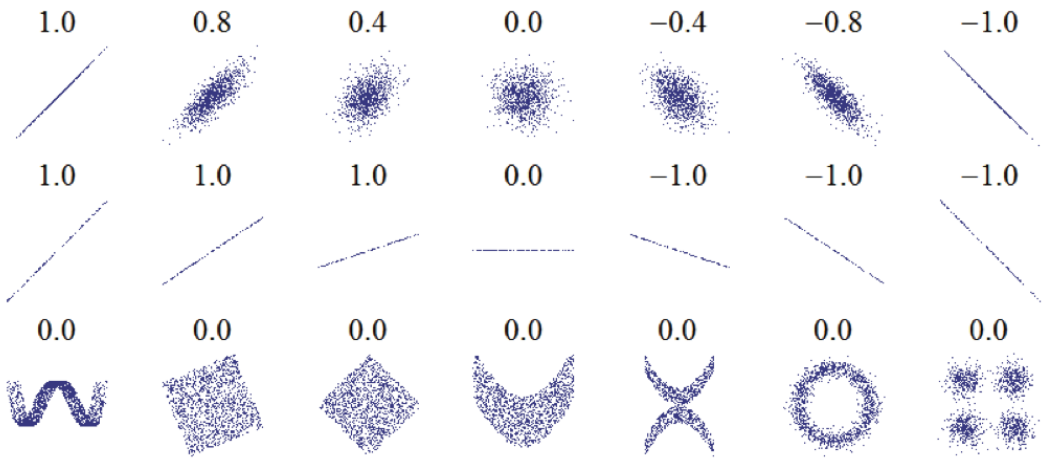
- 연속형 변수
- 정규분포

<그림 참조>
https://diseny.tistory.com/entry/%EC%83%81%EA%B4%80%EA%B4%80%EA%B3%84%EC%99%80-%EC%83%81%EA%B4%80%EA%B3%84%EC%88%98

# Mutual Information

## Correlation의 한계

- 데이터 분포의 경사 반영 X
- 비선형 관계성 반영 X




## Mutual Information

- 두 데이터의 dependence를 측정
- 두 변수의 관계성을 포착(비선형 포함)
- Ex) 독립일 경우

  p(x,y) = p(x)p(y)

  ∴ log(p(x,y)/p(x)p(y)) = log(1) = 0

Mutual Information은 joint distribution p(X, Y)가 p(X)p(Y)와 얼마나 비슷한지를 측정하는 척도로, 아래와 같이 정의할 수 있다.

$$\mathbb{I}(X;Y) \triangleq \mathbb{KL}\left(p(x,y) \| p(x)p(y)\right) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \frac{p(x,y)}{p(x)\,p(y)}$$

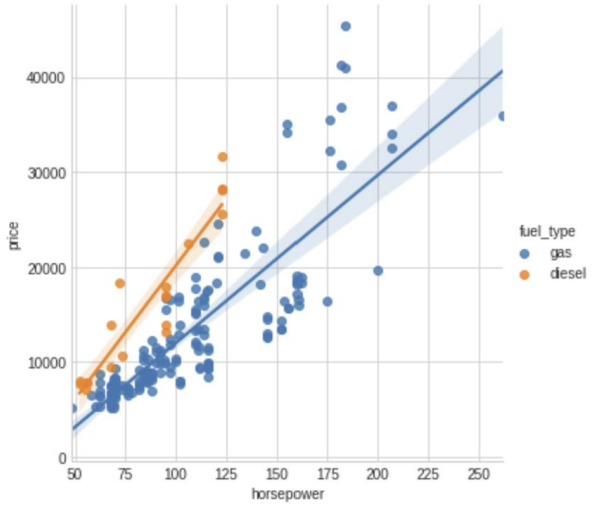
MI의 정의

<그림 참조>
https://process-mining.tistory.com/141

# Mutual Information

## Mutual information의 한계

- 두 데이터의 직접적인 관계성만 파악
- => 모델 선택에 영향을 미침




MI만 사용할 경우, 단순 관계성 유무에 대해서만 나타나기 때문에

정리하여, mutual information은 모델이

1. 사용하고 해석하기 쉽거나
2. 계산 과정에서 효율성이 보이거나
3. 이론적으로 잘 만들어졌거나
4. 과적합을 예방하거나
5. 모든 종류의 관계를 감지할 수 있다면

* horsepower의 MI는 높으며, fuel_type의 MI는 낮은 경우 *

=> fuel_type에 따라 horsepower와 price의 관계성을 더 구체적으로 표현 가능

=> MI가 낮다고 feature의 의미가 없는 것은 아님

<그림 참조>
https://wakaranaiyo.tistory.com/214

# 참조 링크

1) 상관관계에 대해서
https://diseny.tistory.com/entry/%EC%83%81%EA%B4%80%EA%B4%80%EA%B3%84%EC%99%80-%EC%83%81%EA%B4%80%EA%B3%84%EC%88%98

2) 피어슨 상관관계 사용 조건
https://eigenvector.tistory.com/36

3) Mutual Information
https://wakaranaiyo.tistory.com/214

4) Mutual Information과 correlation
https://process-mining.tistory.com/141

# Code w. Housing Price Data

## Correlation

```
# 수치형 변수 간 상관관계
import seaborn as sns
import matplotlib.pyplot as plt

num_cols = df.select_dtypes(include=['number']).columns
corr = df[num_cols].corr()
corr = corr.dropna(axis=1)

# 집값(SalePrice)과의 상관관계 내림차순 정렬
corr_target = corr['SalePrice'].sort_values(ascending=False)

print(corr_target[1:].head(10))
```

29] ✓ 0.0s

```
OverallQual     0.790982
GrLivArea       0.708624
GarageCars      0.640409
GarageArea      0.623431
TotalBsmtSF     0.613581
1stFlrSF        0.605852
FullBath        0.560664
TotRmsAbvGrd    0.533723
YearBuilt       0.522897
YearRemodAdd    0.507101
Name: SalePrice, dtype: float64
```

## Mutual information

```
from sklearn.feature_selection import mutual_info_regression

X = df.select_dtypes(include=['number']).drop('SalePrice', axis=1)
# X = X.fillna(X.median())  # 또는 fillna(0), dropna(axis=1)
X = X.dropna(axis=1)
mi = mutual_info_regression(X, SalePrice)
mi_series = pd.Series(mi, index=X.columns).sort_values(ascending=False)
print(mi_series.head(10))
```

✓ 0.3s

```
OverallQual     0.566789
GrLivArea       0.480381
TotalBsmtSF     0.370926
GarageArea      0.366466
GarageCars      0.364490
YearBuilt       0.354266
1stFlrSF        0.308283
MSSubClass      0.275609
FullBath        0.256406
YearRemodAdd    0.242115
dtype: float64
```

# To Do list

## 1) p-value
: 정규분포를 따를 때, 신뢰성이 높음 -> 피어슨 상관관계 사용 여부 확인

## 2) Entropy
: Mutual Information 기본 지식

## 3) Joint distribution
: Mutual Information 기본 지식

## 4) 여러가지 데이터 상관관계 분석
: https://boksup.tistory.com/59