2주차:
통계량의 시각화

서울시립대학교 통계데이터사이언스학과
전종준

빈도분석

범주형 (확률)변수와 빈도

- 범주형 데이터: 이산적인 값을 데이터. 관측값의 종류가 많아야 countable.
- 범주형 확률변수: 범주형 데이터를 이산적인 실수값으로 변환(코딩)해주는 함수.
- 수학적 정의: 확률변수 X 에 대해 $P(X = x) > 0$ 인 $x \in \mathbb{R}$ 를 모두 모은 집합을 A라고 할 때, $P(A) = 1$ 이 되면 X를 범주형 변수라 부르고, A 의 원소 x 들을 level 이라 부른다.

- 예시
  - 성별 (여:0, 남자1) 을 코딩한 확률변수 X 는 범주형 변수
  - 집주소에서 시군구,읍면동 을 코딩한 확률변수 X 는 범주형 변수

빈도분석

범주형 데이터의 코딩

- 자연수 코딩의 예
  - 혈액형 A, B, O, AB 형을 데이터의 값 (4개의 level 값을 가지고 있음)
  - 자연수 1,2,3,4 에 각 혈액형 데이터 값을 대응

    $X(A) = 1, X(B) = 2, X(AB) = 3, X(O) = 4$
- 확률분포
  - 원래 데이터의 확률분포가 $P(\{A\}) = 0.4, P(\{B\}) = 0.3, P(\{AB\}) = 0.2, P(\{O\}) = 0.1$ 라 하자
  - $P(X = 1), \ldots$
  - $p_X(x) = P(X = x)$ 로 정의하면 $p_X(x)$ 는 X 의 확률분포 정보를 모두 표현할 수 있음.

빈도분석

범주형 데이터의 코딩

- 벡터코딩의 예
  - 혈액형 A, B, O, AB 형을 데이터의 값 (4개의 level 값을 가지고 있음)
  - 4차원 벡터에 각 혈액형 데이터 값을 다음과 같이 대응

    $Y(A) = (1,0,0,0), Y(B) = (0,1,0,0), Y(AB) = (0,0,1,0), Y(O) = (0,0,0,1)$
- 확률분포
  - $P(Y = (1,0,0,0)) = 0.4.$, ($P(Y = 1)$ 은 정의 안됨)
  - $p_Y(y) = P(Y = y)$ (단, $y \in \mathbb{R}^4$) 로 정의하면 $p_Y(y)$ 는 Y 의 확률분포 정보를 모두 표현할 수 있음.

빈도분석

범주형 변수와 정보

- 범주형 변수의 확률분포
  - $P(X = x)$ 의 정보를 표현하는 것은 각 x 마다 $P(X = x)$ 를 나타내면 됨. -> 막대그래프
  - $P(X \leq x)$ 의 그래프를 그려도 됨 -> 누적분포 함수, 점프 점이 확률값을 나타냄

- 빈도표와 상대빈도의 시각화

빈도분석

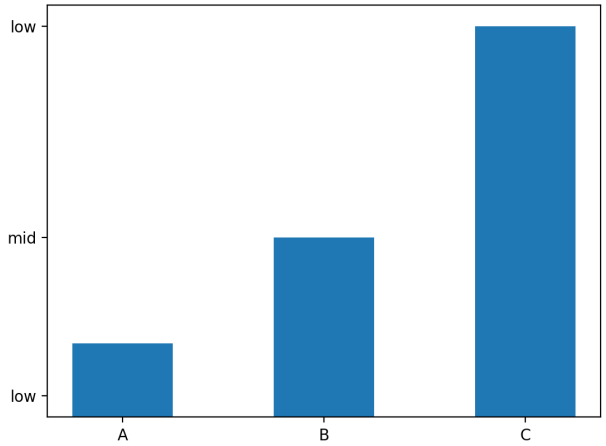
범주형 변수와 정보

- 막대그래프와 확률분포의 추정
  - $P(X = x)$ 이 알려진 경우 -> 확률분포를 알고 있는 경우
  - $P(X = x)$ 가 알려지지 않고, 자료만 주어진 경우

    $P(X = x)$ 의 추정값으로 상대빈도를 계산, 데이터가 $x_1, \dots, x_n$ 으로 주어져있다고 하면 값 x를 가지는 자료의 상대빈도 $n_x = \frac{\sum_{i=1}^{n} I(x_i=x)}{n}$ (대수의 법칙)
- 상대빈도의 시각화 -> 이산형 확률변수의 확률분포 추정에 대한 시각화

빈도분석

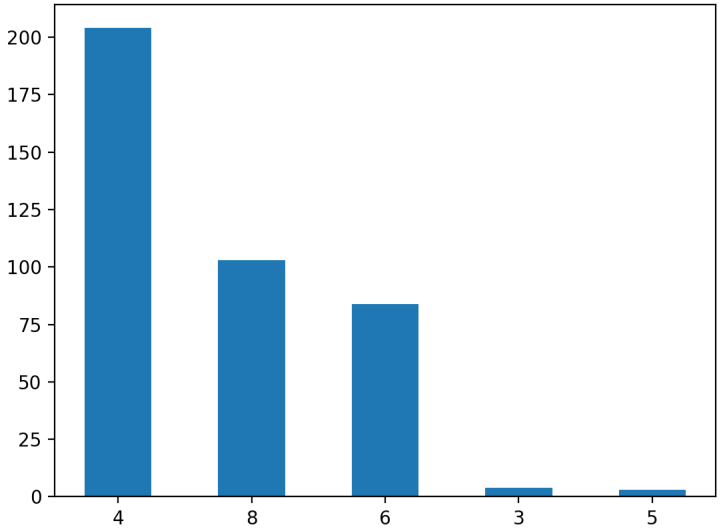
```
import matplotlib.pyplot as plt
import numpy as np
# %%
# x축, height, width
x = np.array([1.0, 2.0, 4.0])
plt.bar(range(len(x)), x, width = 0.5, bottom= 0)
plt.xticks(range(len(x)) , ['A', 'B', 'C'])
plt.yticks(np.array([0.5,2,4]) , ['low', 'mid', 'low'])
plt.ylim((0.3))
```



빈도분석

MTCARS dataset

```
import pandas as pd
import matplotlib.pyplot as plt
# mtcars 데이터셋 로드
url =
"https://raw.githubusercontent.com/mwaskom/seabor
n-data/master/mpg.csv"
df = pd.read_csv(url)
freq_counts = df['cylinders'].value_counts()
x = freq_counts.values
plt.bar(range(len(x)), x, width = 0.5, bottom= 0)
plt.xticks(range(len(x)), freq_counts.index)
plt.show()
```



빈도분석

범주형 변수와 빈도

- 성별은 범주형 변수인가?
- 나이는 범주형 변수인가?
- 몸무게는 범주형 변수인가?
- 범주형 변수가 아닌 자료는 무엇인가?

- 분석을 쉽게 하기 위해서 연속형 확률변수를 도입함.

빈도분석

연속형 (확률)변수와 빈도

- 개념: 확률변수 X 의 값이 연속적인 속성을 가져서 모든 값 x 에 대해 $P(X = x) = 0$ 이고 X가 구간에서만 확률을 같는 경우
- 수학적 정의: 확률변수 X 에 대해 $P(X \in B) > 0$ 인 B 가 있다면 항상 B 의 원소 개수가 셀수없는 경우 (uncountable) 이 확률변수를 연속형이라고 함.
- 일변량 자료:

연속형분포와 히스토그램

연속형 확률변수의 시각화

- 확률밀도함수 그래프

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm, chi2

x = np.linspace(-8, 8, 100)
y = norm.pdf(x, loc = 0, scale = 1)
plt.plot(x,y, alpha = 0.7)
plt.xlim((-8,8))
plt.xlabel('x : support')
plt.ylabel('f(x) : density')
plt.show()
```

연속형분포와 히스토그램

연속형 확률변수의 시각화

- 히스토그램

```
np.random.seed(1)
x = np.random.normal(0, 1, 1000)
plt.hist(x,bins=20, density=True)
```

연속형분포와 히스토그램

연속형 확률변수의 시각화

- 히스토그램의 수리적 표현
  - 편의상 위해 0과 1사이에 정의된 분포를 가정하고 같은 길이의 M개의 bin 을 다음과 같이 표시

$$B_1 = \left[0, \frac{1}{M}\right), B_2 = \left[\frac{1}{M}, \frac{2}{M}\right), \cdots, B_{M-1} = \left[\frac{M-2}{M}, \frac{M-1}{M}\right), B_M = \left[\frac{M-1}{M}, 1\right].$$

  - 히스토그램

$$\widehat{p}_n(x) = \frac{\text{number of observations within } B_\ell}{n} \times \frac{1}{\text{length of the bin}} = \frac{M}{n} \sum_{i=1}^{n} I(X_i \in B_\ell).$$

연속형분포와 히스토그램

연속형 확률변수의 시각화

- 히스토그램의 수학적표현
  - x 에서 히스토그램의 높이 (함수값) = x가 속한 bin에 들어간 관측치 개수/n
    (모든 데이터에 대해서 x 속한 bin 에 해당 데이터가 있으면 1점, 없으면 0점해서 총합계)/n
    -> 거리 개념이 있음. bin 에 속하지 않으면 x에서 가깝든 멀든 모두 0점

  - 1/n점, 0/n점으로 된 이산적 점수를 연속적으로 만들고, 거리에 따라 역가중할 수 없나?
    커널함수의 도입

연속형분포와 히스토그램

연속형 확률변수의 시각화

- 히스토그램의 점근적 성질 [참고]
  - 만약 참 밀도함수 $\mathrm{p}(x)$ 의 미분값이 균등 유계 (uniformly bounded) 면 [0,1] 위의 M개의 bin 으로 이루어진 히스토그램함수 $\widehat{p_M}(x)$ 는 다음과 같이 참밀도함수를 근사함.

$$\sup_x |\widehat{p}_M(x) - p(x)| = O\left(\frac{1}{M}\right) + O_P\left(\sqrt{\frac{M^2 \log M}{n}}\right)$$

연속형분포와 히스토그램

연속형 확률변수의 시각화

- 커널밀도추정 (Kernel Density Estimation)
  - 주어진 데이터로부터 확률 밀도 함수를 비모수적으로 추정하는 방법

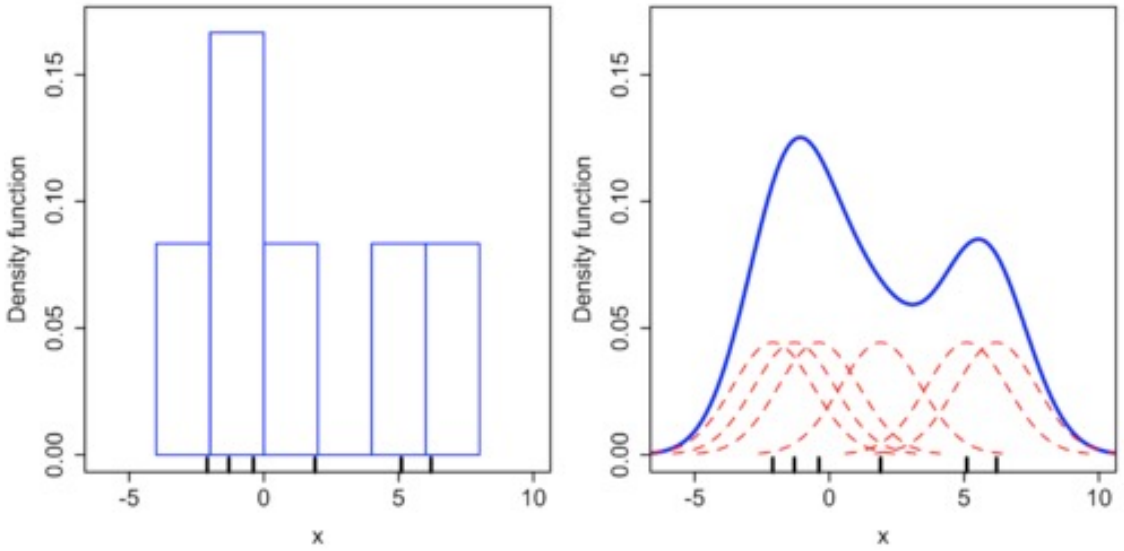
$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right)$$

  - K: 커널함수, h>0 bandwidth parameter

1. $\int K(x)dx = 1$ (정규화 조건)
2. $K(x) \geq 0$ (비음수 조건)
3. $\int xK(x)dx = 0$ (대칭성)

연속형분포와 히스토그램

히스토그램과 커널밀도 추정의 비교



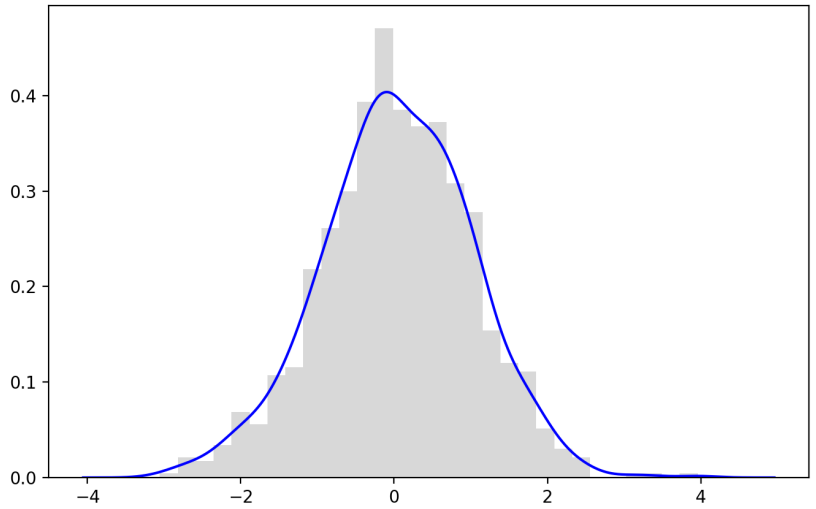
연속형분포와 히스토그램

히스토그램과 커널밀도 추정의 비교

```
from scipy.stats import gaussian_kde
kde = gaussian_kde(x)
x_vals = np.linspace(min(x)-1, max(x)+1, 1000) # 평가할 x 값들
y_vals = kde(x_vals)

# 그래프 시각화
plt.figure(figsize=(8, 5))
plt.hist(x, bins=30, density=True, alpha=0.3, color='gray', label='Histogram')
plt.plot(x_vals, y_vals, color='blue', label='KDE (gaussian_kde)')
```

연속형분포와 히스토그램

히스토그램과 커널밀도 추정의 비교



다변량자료의 시각화

결합확률

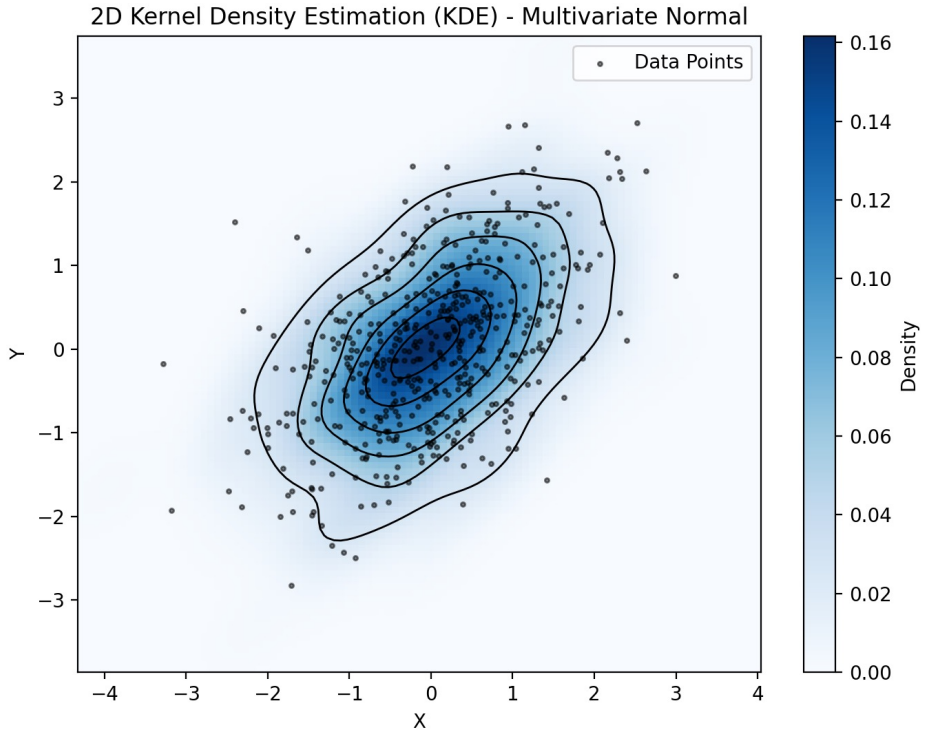
- 연속형 자료의 결합확률
  - 2차원 자료의 경우 2차원 히스토그램
  - 커널밀도함수 추정 -> 3차원 그림, 혹은 heat map
    (적절한 bandwidth 설정이 중요함)
- 범주형 자료의 결합확률
  - 2차원 범주형 자료 -> 분할표 (결합확률, 주변확률, 조건부 확률 등을 표현가능함)

다변량자료의 시각화

```
mean = [0, 0] # 평균 벡터
cov = [[1, 0.5], [0.5, 1]] # 공분산 행렬
data = np.random.multivariate_normal(mean, cov,
size=500)
x = data[:,0]
y = data[:,1]

kde = gaussian_kde(data.T)
x_vals = np.linspace(min(x)-1, max(x)+1, 100)
y_vals = np.linspace(min(y)-1, max(y)+1, 100)
X, Y = np.meshgrid(x_vals, y_vals)
z_vals = kde(np.vstack([X.ravel(),
Y.ravel()])).reshape(X.shape)
```

```
# 그래프 시각화
plt.figure(figsize=(8, 6))
plt.pcolormesh(X, Y, z_vals, shading='auto',
cmap='Blues')
plt.colorbar(label='Density')
plt.contour(X, Y, z_vals, colors='black', linewidths=1)
# 등고선 추가
plt.scatter(x, y, s=5, color='black', alpha=0.5,
label='Data Points')
plt.xlabel('X')
plt.ylabel('Y')
plt.title('2D Kernel Density Estimation (KDE) -
Multivariate Normal')
plt.legend()
plt.show()
```

다변량자료의 시각화


2D Kernel Density Estimation (KDE) - Multivariate Normal
Data Points
X
Y
Density
−4
−3
−2
−1
0
1
2
3
4
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00

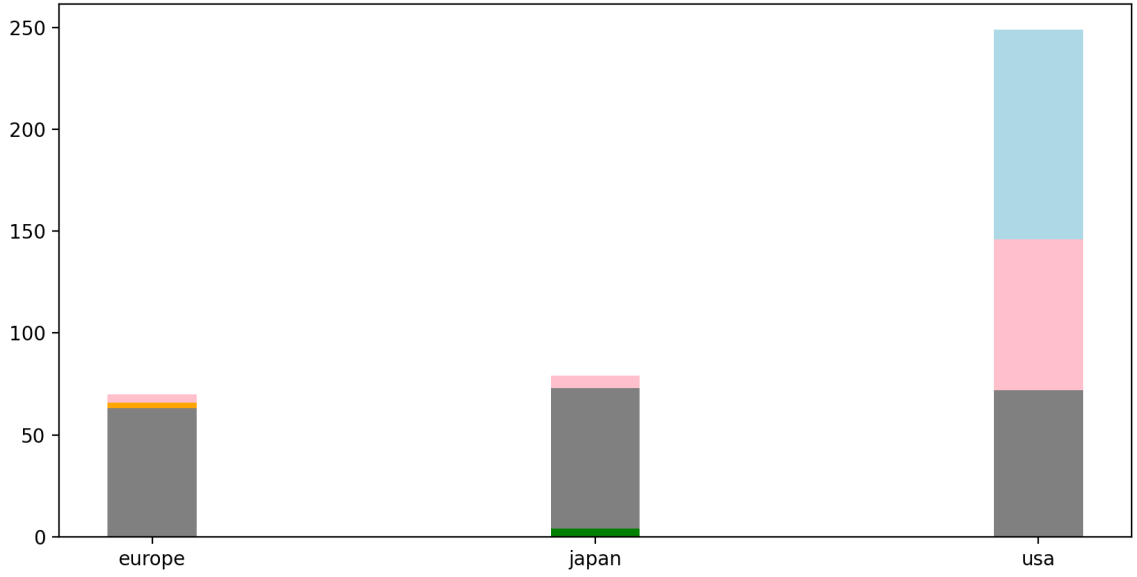
다변량자료의 시각화

범주형자료의 결합빈도

```
import pandas as pd
import matplotlib.pyplot as plt
# mtcars 데이터셋 로드
url =
"https://raw.githubusercontent.com/mwaskom/seaborn-
data/master/mpg.csv"
df = pd.read_csv(url)
freq_df = df[['cylinders',
'origin']].groupby(['cylinders',
'origin']).size().unstack(fill_value=0)
Y = np.array(freq_df)
x = np.arange(3)
fig, ax = plt.subplots(figsize = (10,5))
h1 = Y[0,:]; h2 = Y[1,:]; h3 = Y[2,:]; h4 = Y[3,:]; h5
= Y[4,:]
w = 0.2
fig, ax = plt.subplots(figsize = (10,5))
ax.bar(x, h1, width=w, color ='green')
ax.bar(x, h2, width=w, bottom = h1, color ='gray')
ax.bar(x, h3, width=w, bottom = h1+h2, color ='orange')
ax.bar(x, h4, width=w, bottom = h1+h2+h3, color ='pink')
ax.bar(x, h5, width=w, bottom = h1+h2+h3+h4, color
='lightblue')
ax.set_xticks(x, freq_df.columns.values)
```

다변량자료의 시각화

범주형자료의 결합빈도



다변량자료의 시각화

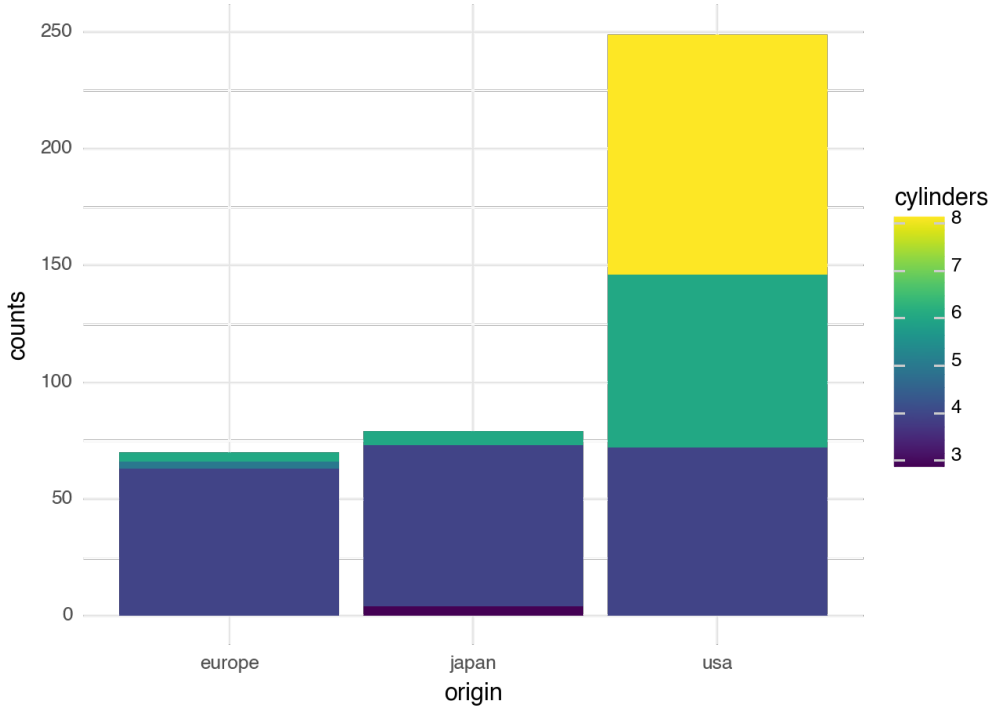
범주형자료의 결합빈도

```
import numpy as np
import pandas as pd
from plotnine import ggplot, aes, geom_bar,
theme_minimal, scale_fill_brewer

# 데이터 생성
freq_df = df[['cylinders',
'origin']].groupby(['cylinders', 'origin']).size()
freq_df = freq_df.reset_index()
freq_df.rename(columns={0:'counts'}, inplace = True)
```

```
# ggplot으로 Stacked Bar Chart 그리기
p = (
ggplot(freq_df, aes(x='origin', y='counts',
fill='cylinders')) +
geom_bar(stat='identity') +
theme_minimal()
)

print(p)
```

다변량자료의 시각화

범주형자료의 결합빈도



다변량자료의 시각화

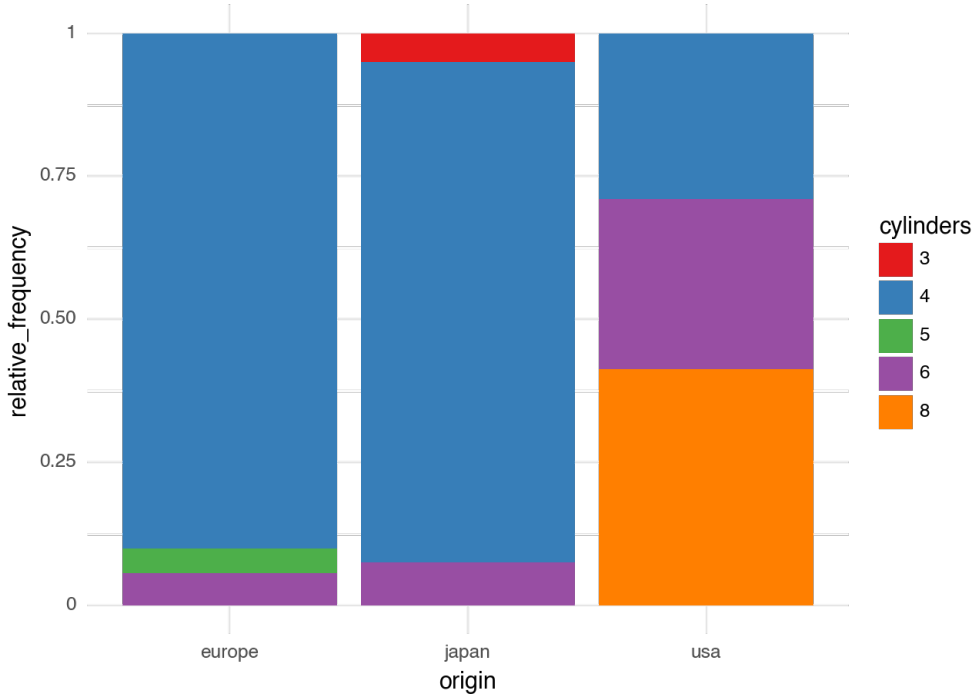
범주형자료의 결합빈도

```
freq_df = df[['cylinders',
'origin']].groupby(['cylinders', 'origin']).size()
# Convert counts to relative frequencies within each
'origin' group
freq_df = freq_df.groupby(level='origin').apply(lambda
x: x / x.sum())
freq_df = freq_df.reset_index(name='relative_frequency')
freq_df['cylinders'] =
freq_df['cylinders'].astype('category')
```

```
# ggplot으로 Stacked Bar Chart 그리기
p = (
ggplot(freq_df, aes(x='origin', y='relative_frequency',
fill='cylinders')) +
geom_bar(stat='identity') +
theme_minimal() +
scale_fill_brewer(type='qual', palette='Set1')
)

print(p)
```

다변량자료의 시각화

범주형자료의 결합빈도



자료분포의 비교

MTCARS 데이터

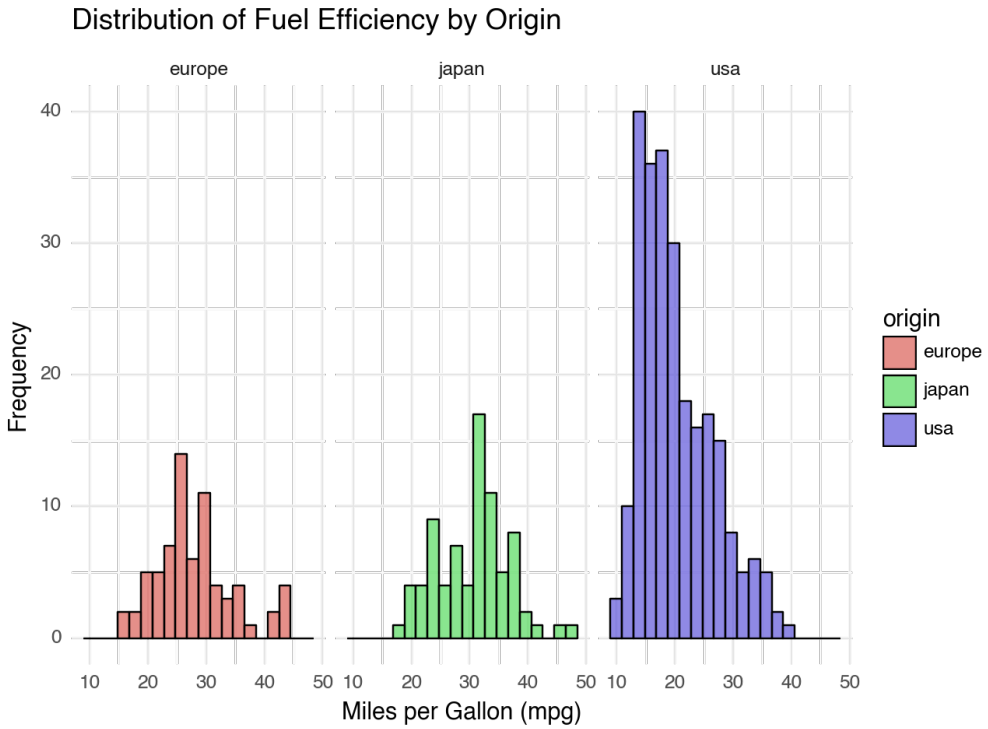
- 원산지별 차량의 연비 분포의 비교

```
from plotnine import ggplot, aes, geom_histogram,
facet_wrap, theme_minimal, labs
import pandas as pd

# Create faceted histograms for each 'origin'
p = (
ggplot(df, aes(x='mpg', fill='origin')) +
geom_histogram(bins=20, alpha=0.7, color='black') +
facet_wrap('~origin') +
theme_minimal() +
labs(title='Distribution of Fuel Efficiency by Origin',
x='Miles per Gallon (mpg)',
y='Frequency')
)

print(p)
```

자료분포의 비교

MTCARS 데이터

- 원산지별 차량의 연비 분포의 비교



자료분포의 비교

MTCARS 데이터

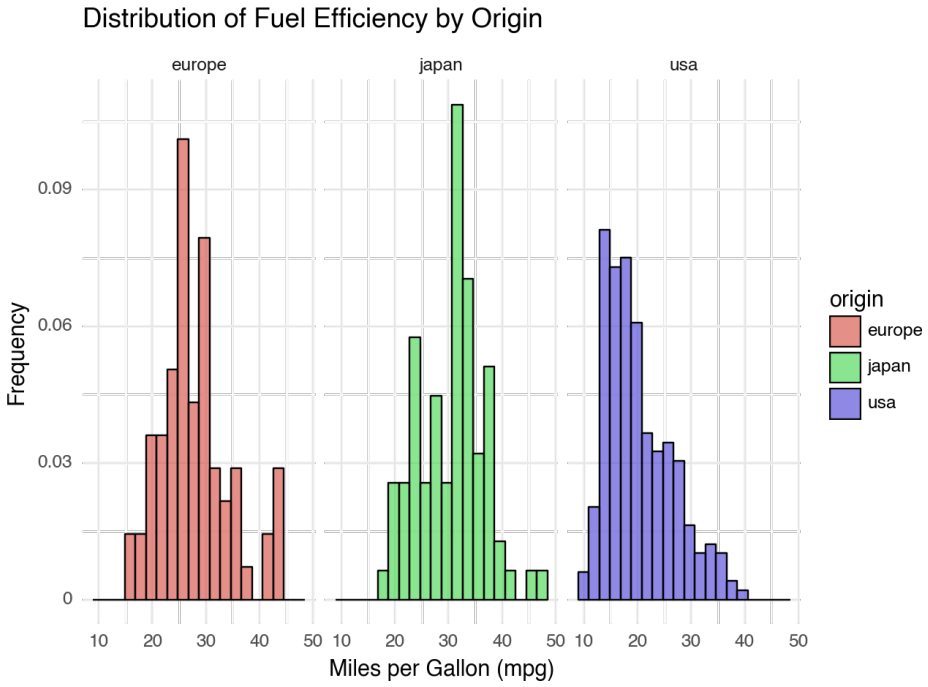
- 원산지별 차량의 연비 분포의 비교

```
p = (
ggplot(df, aes(x='mpg',fill='origin')) +                print(p)
geom_histogram(aes(y = '..density..'), bins=20,
alpha=0.7, color='black') +
facet_wrap('~origin') +
theme_minimal() +
labs(title='Distribution of Fuel Efficiency by Origin',
x='Miles per Gallon (mpg)',
y='Frequency')
)
```

자료분포의 비교

MTCARS 데이터

- 원산지별 차량의 연비 분포의 비교
  - 평균, 분산의 비교
  - pdf 의 비교

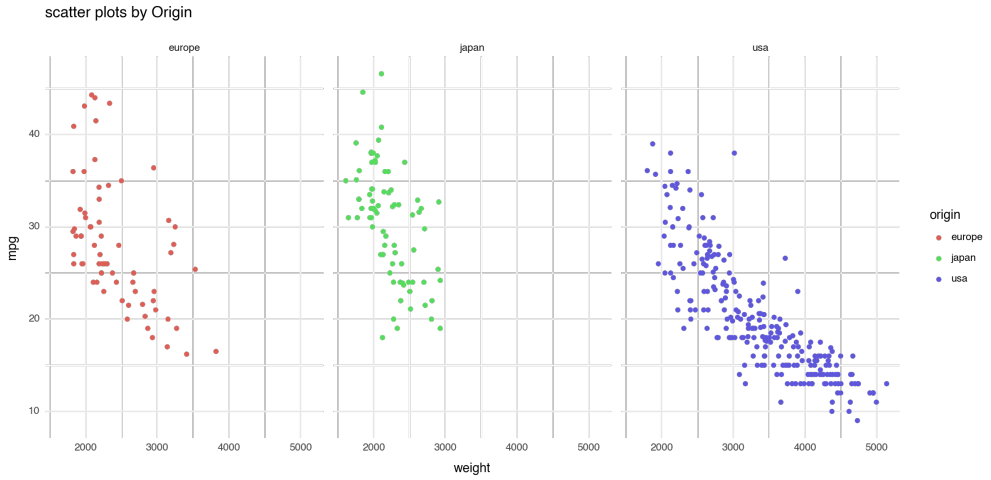


자료분포의 비교

MTCARS 데이터

- 원산지별 차량 무게와 연비의 관계 비교

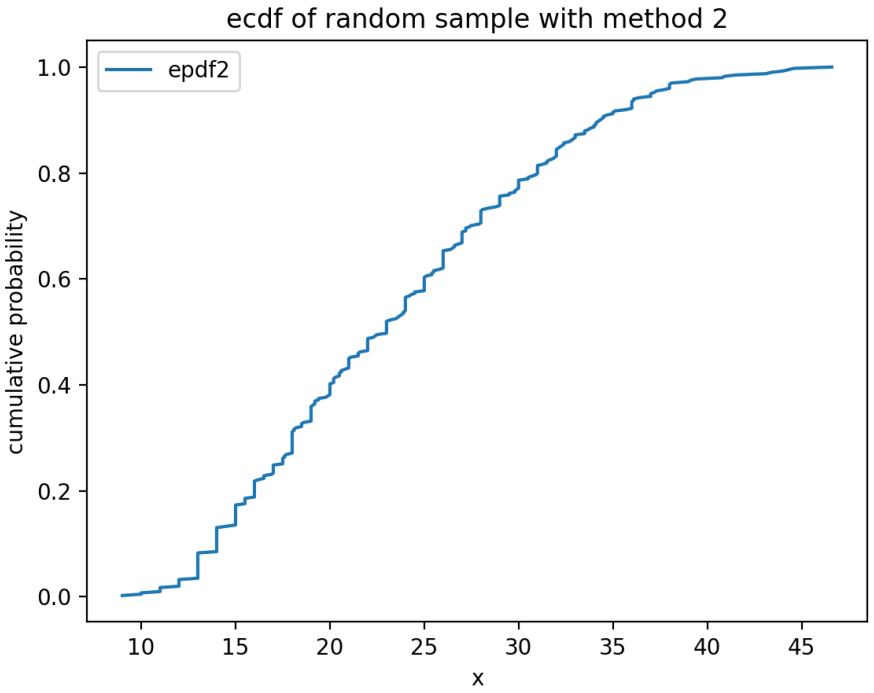
```
from plotnine import geom_point, theme
p = (
ggplot(df, aes(x='weight', color='origin')) +
geom_point(aes(y = 'mpg')) +
facet_wrap('~origin') +
theme_minimal() +
theme(figure_size=(12, 6)) +
labs(title='scatter plots by Origin',
x='weight',
y='mpg')
)
print(p)
```



자료분포의 비교

경험적 확률분포 (empirical cdf)

```
from
statsmodels.distributions.empirical_distribution
import ECDF
x = df['mpg'].to_numpy()
# ECDF 함수를 이용한 경험적 누적분포함수 시각화
ecdf = ECDF(x)
plt.plot(ecdf.x, ecdf.y, label='epdf2')
plt.xlabel('x')
plt.ylabel('cumulative probability')
plt.title('ecdf of random sample with method 2')
plt.legend()
plt.show()
```



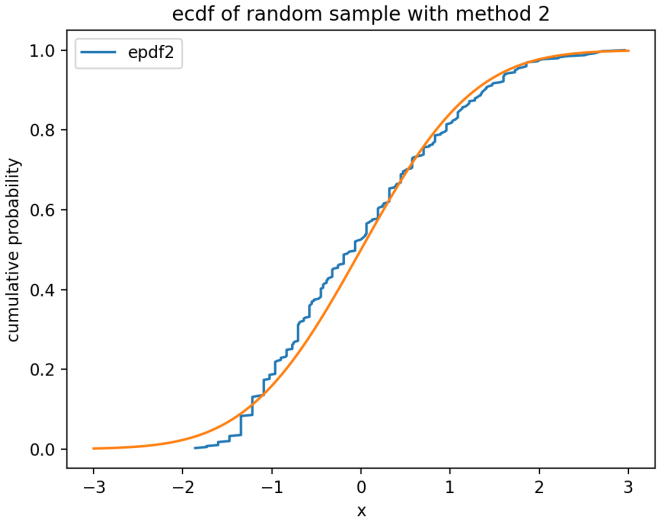
자료분포의 비교

Kolmogorov-Smirnov test

- 분포간의 거리를 재는 통계량 F_n 을 empirical cdf, F를 cdf 라고 할때 두 분포의 거리를 다음과 같이 정의

$$D = \sup_x |F_n(x) - F(x)|$$

- 검정통계량: 귀무가설은 자료가 분포 F를 따른다. 그러면 n이 클 때 $\sqrt{n}\sup_x |F_n(x) - F(x)|$ 는 특별한 패턴을 가짐. 그 패턴을 이용해서 가설검정을 함.



자료분포의 비교

Kolmogorov-Smirnov test

- 분포간의 거리가 가지는 현실적인 의미는 무엇인가?
  - X, Y 를 각각 다른 자산의 가격을 나타내는 확률변수라고 하자.
  - 두 확률변수에 대한 누적분포함수를 각각 F, G 라고 놓자.
  - 각 자산 가격이 x이하일 확률은 $F(x) = P(X \leq x), G(x) = P(Y \leq x)$ 고 두 자산가격의 분포 (가격 패턴)이 같다면 모든 x에 대해서 $F(x) - G(x) = 0$ 것임.
  - $\sup_x |F(x) - G(x)|$ 는 두 cdf 의 값의 차이의 최대값을 나타냄

자료분포의 비교

Cramer-von Mises

- 분포간의 거리를 재는 다른 방법
  - $C = \int \left(F_n(x) - F(x)\right)^2 f(x)dx$

자료분포의 비교

분포의 거리

- Hellinger Distance
  - 두 개의 pdf f, g 가 주어져 있음
  - $h(f,g)^2 = \int (\sqrt{f(x)} - \sqrt{g(x)})^2 \, dx$
- KL divergence of f from g
  - $\mathrm{KL}(f||g) = \int \log \frac{f(x)}{g(x)} f(x) dx$
  - **[예시] 두 정규분포의 KL-divergence 를 구해보시오**

자료분포의 비교

분포의 거리

- KL divergence of f from g
  - $\mathrm{KL}(f||g) \geq 0$
  - $\mathrm{KL}(f||g) = 0$ if and only if $f = g$
  - $\mathrm{KL}(f||g) \neq \mathrm{KL}(g||f)$
  - **(참고)** $\{x: f(x) > 0, g(x) = 0\}$ **인 구간이 있으면,** $\mathrm{KL}(f||g) = \infty$
    두 분포의 support 가 다를때 분포의 발산정도를 정량화하는데 문제가 생김
    -> Wasserstein Distance 가 주목받음

자료분포의 비교

분포 학습

- $d(f_\theta, f_{data})$: pseudo distance of f_θ and c
  - f_θ: 학습모형, f_{data}: 데이터의 분포 (이미지 혹은 텍스트)
  - $d(f_\theta, f_{data})$ 를 목적함수로 하여 이를 줄이는 방향으로 θ 를 선택하는것 -> 이상적으로 데이터의 분포와 동일한 분포인 f_θ를 얻게 됨
  - f_θ 로 부터 랜덤 샘플을 얻게 되면, 새로운 데이터를 생성하는 것