이변량_숫자 vs 범주

1.환경준비

(1) 라이브러리

In [1]:
```
import pandas as pd
import numpy as np
# import random as rd

import matplotlib.pyplot as plt
import seaborn as sns
from statsmodels.graphics.mosaicplot import mosaic      #mosaic plot!

import scipy.stats as spst
```

(2) 데이터 불러오기

- 다음의 예제 데이터를 사용합니다.

> 타이타닉 생존자

In [2]:
```
# 타이타닉 데이터
titanic = pd.read_csv('https://raw.githubusercontent.com/DA4BAM/dataset/master/titanic.1.csv')
titanic.head()
```

Out[2]:

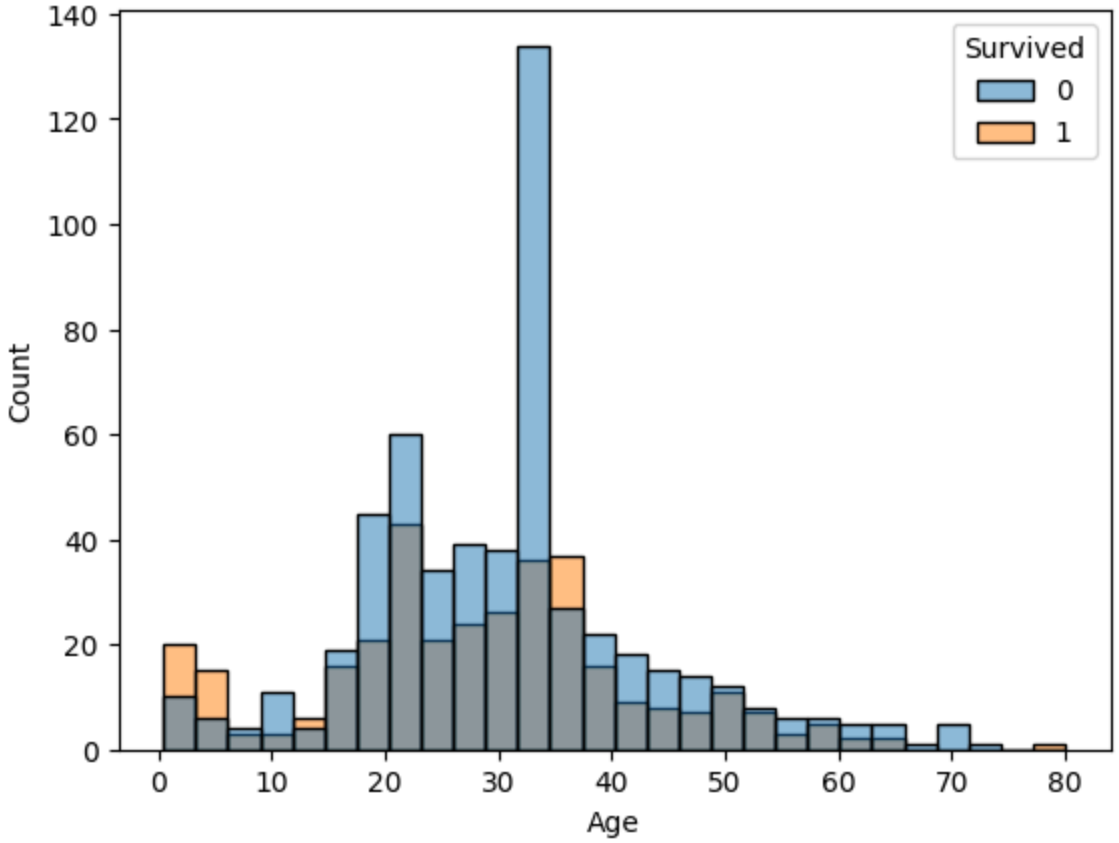
	PassengerId	Survived	Pclass	Title	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarl
0	1	0	3	Mr	male	22.0	1	0	A/5 21171	7.2500	NaN	
1	2	1	1	Mrs	female	38.0	1	0	PC 17599	71.2833	C85	
2	3	1	3	Miss	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	
3	4	1	1	Mrs	female	35.0	1	0	113803	53.1000	C123	
4	5	0	3	Mr	male	35.0	0	0	373450	8.0500	NaN	

2.숫자 --> 범주

(1) 시각화

- 히스토그램을 Survived로 나눠서 그려봅시다.

In [3]:
```
sns.histplot(x='Age', data = titanic, hue = 'Survived')
plt.show()
```

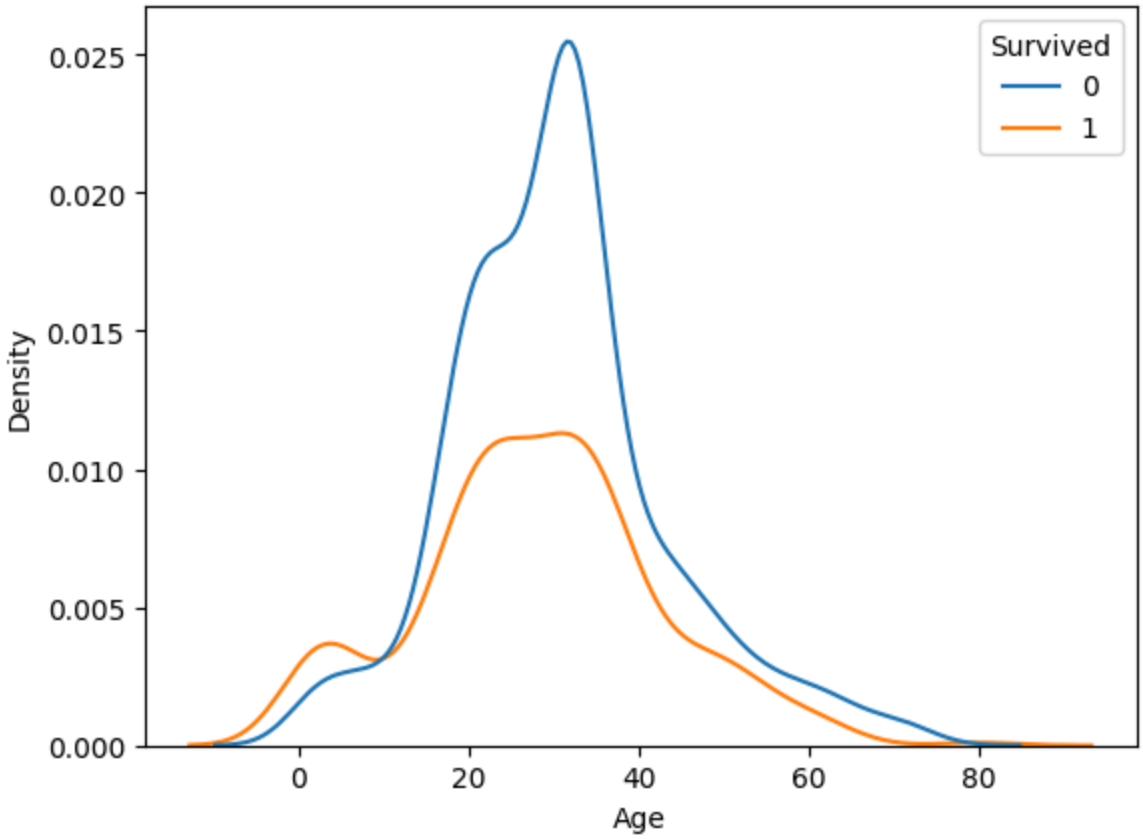


- kdeplot을 그려봅시다.
- 두가지 방법이 있습니다.
  - ① kdeplot(, hue = 'Survived')
    - 생존여부의 비율이 유지된 채로 그려짐
    - 두 그래프의 아래 면적의 합이 1
  - ② kdeplot(, hue = 'Survived', common_norm = False)
    - 생존여부 각각 아래 면적의 합이 1인 그래프
  - ③ kdeplot(, hue = 'Survived', multiple = 'fill')
    - 나이에 따라 생존여부 **비율**을 비교해볼 수 있음. (양의 비교가 아닌 비율!)

① kdeplot(, hue = 'Survived)

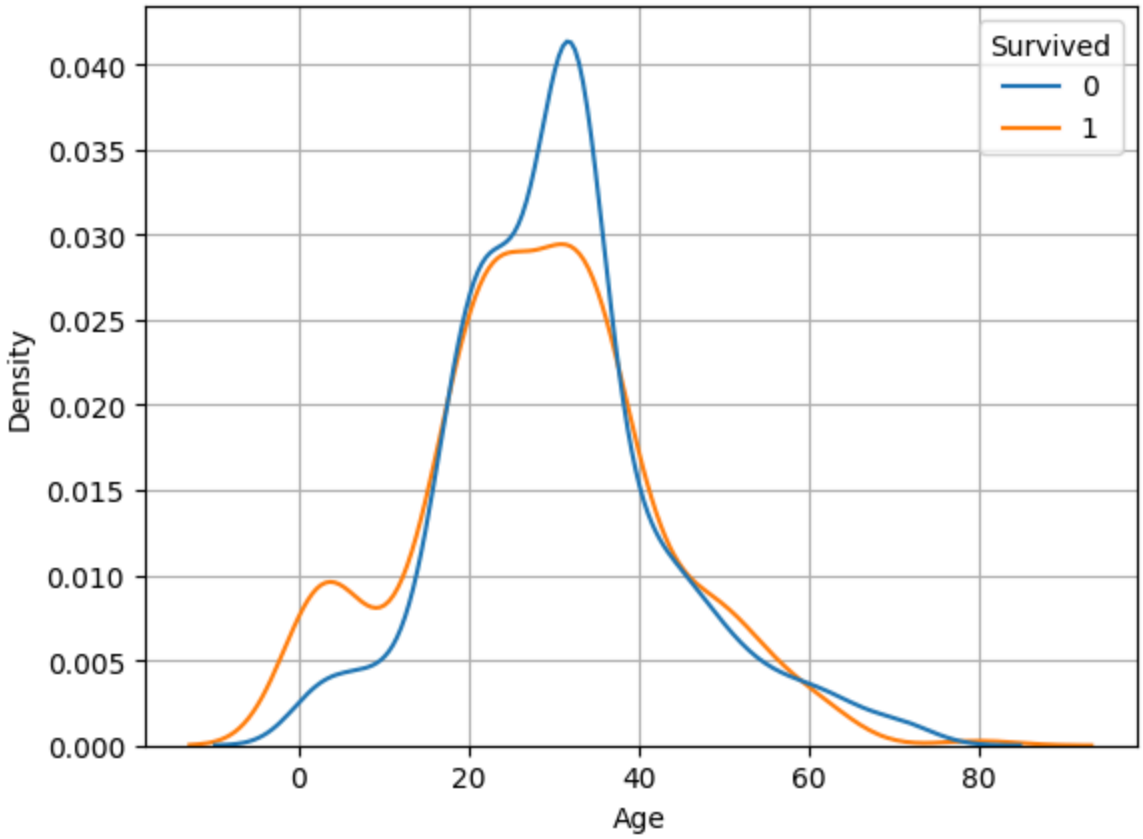
- common_norm = True (기본값)

In [4]:
```
sns.kdeplot(x='Age', data = titanic, hue ='Survived')
plt.show()
```



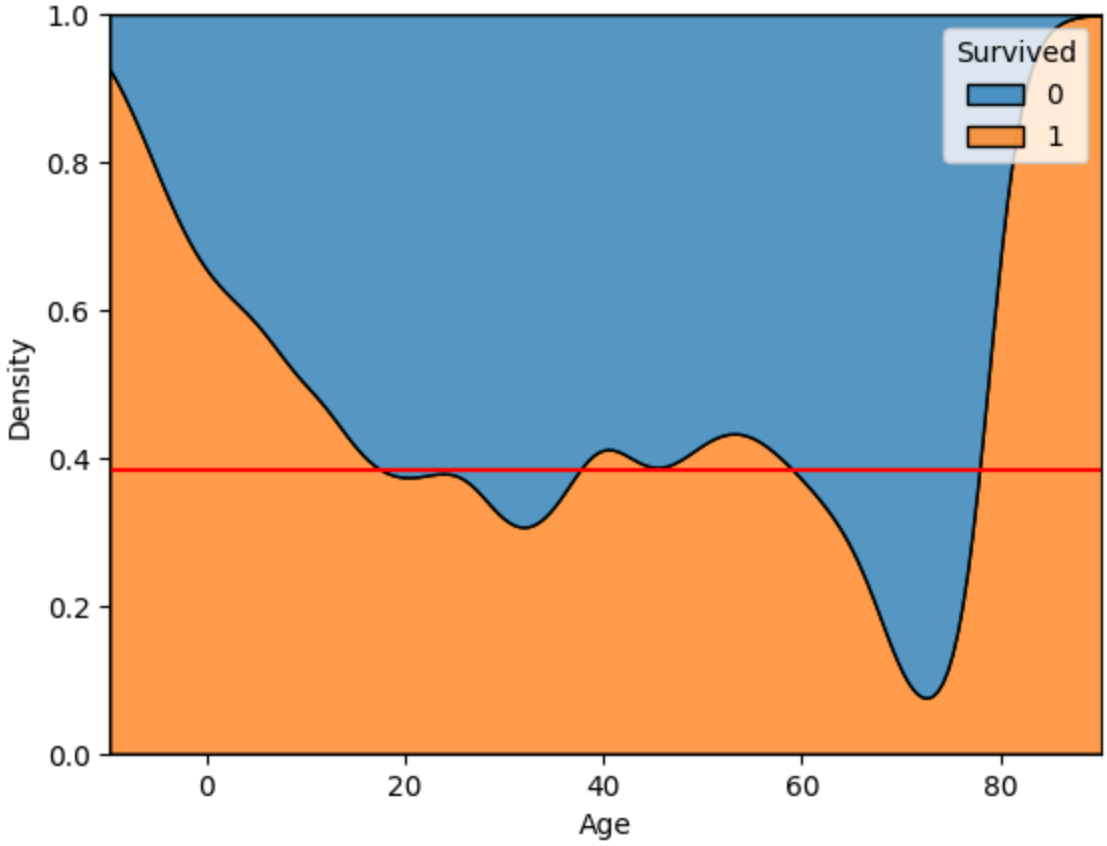
② kdeplot(, hue = 'Survived, common_norm = False)

In [5]:
```
sns.kdeplot(x='Age', data = titanic, hue ='Survived',
            common_norm = False)
plt.grid()
plt.show()
```

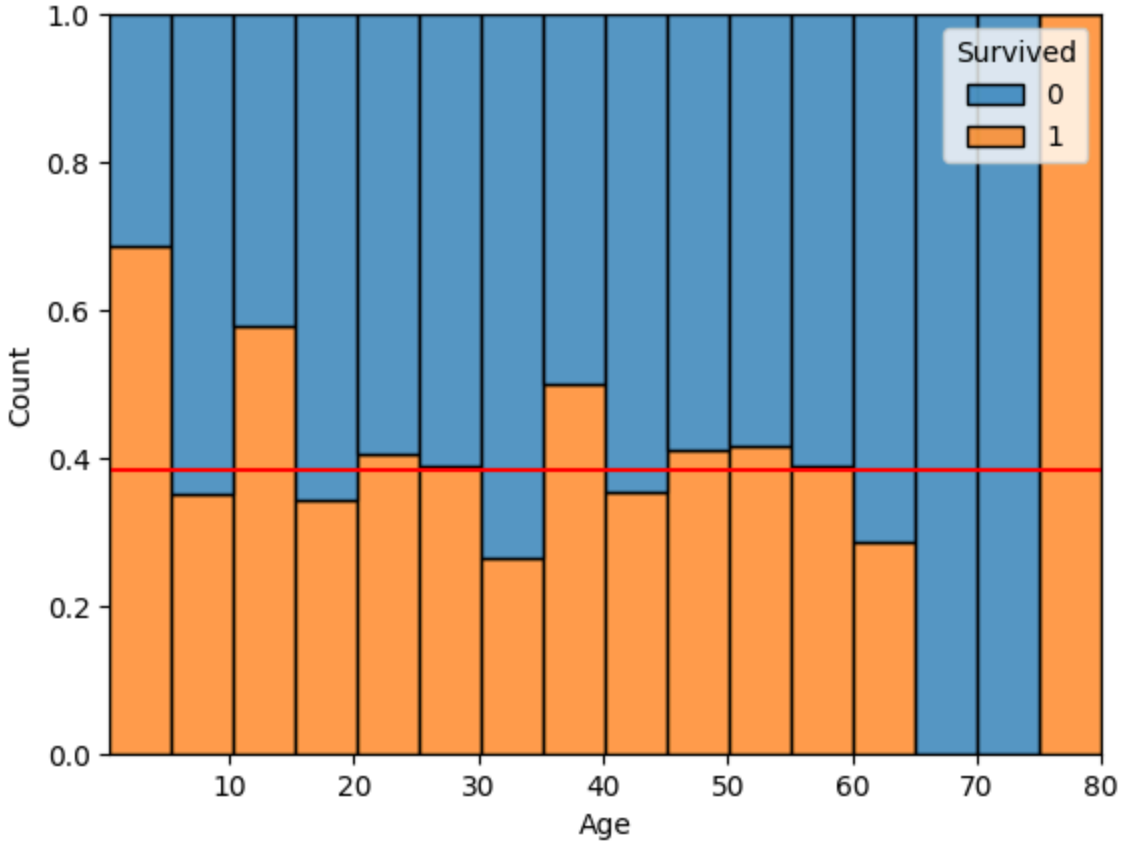


③ multiple = 'fill' 옵션

In [6]:
```
sns.kdeplot(x='Age', data = titanic, hue ='Survived'
            , multiple = 'fill')
plt.axhline(titanic['Survived'].mean(), color = 'r')
plt.show()
```



In [7]:
```
sns.histplot(x='Age', data = titanic, bins = 16
             , hue ='Survived', multiple = 'fill')
plt.axhline(titanic['Survived'].mean(), color = 'r')
plt.show()
```



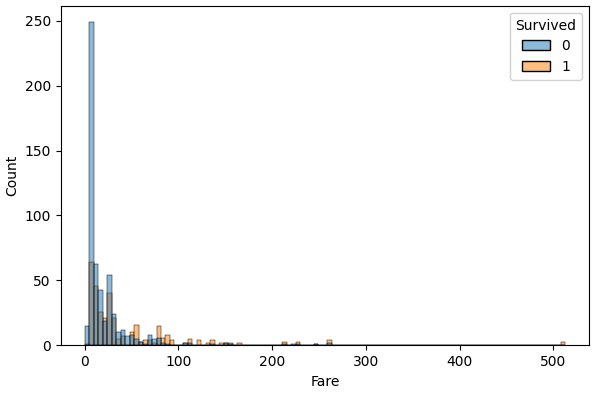
-연습문제-

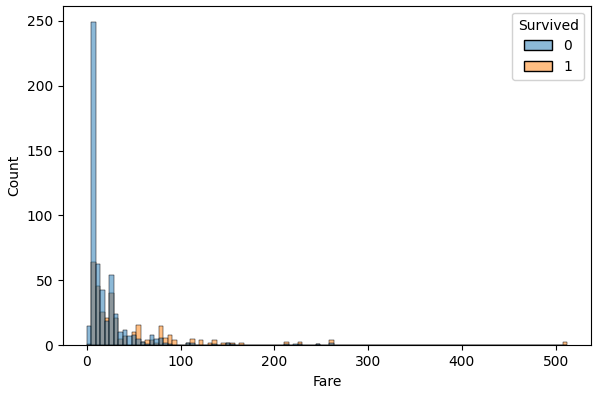
다음의 관계에 대해 시각화 해 봅시다.

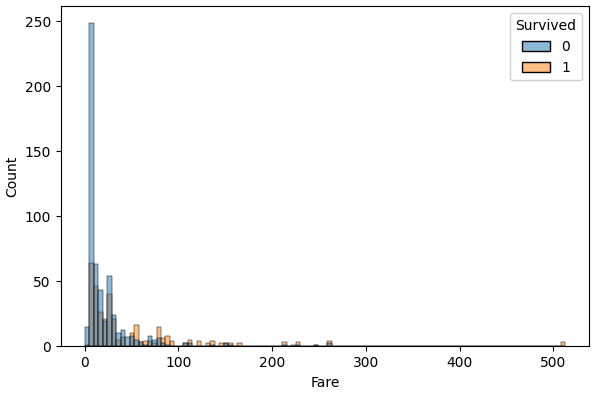
- [문1] Fare(운임) --> Survived

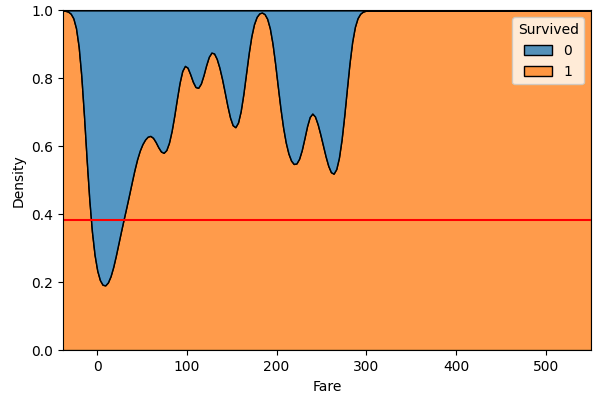
In [17]:
```
def ed_1(var, target, data):
    plt.figure(figsize=(15, 15))

    plt.subplot(3, 2, 1)
    sns.histplot(x=var, data=data, hue=target)

    plt.subplot(3, 2, 2)
    sns.histplot(x=var, data=data, hue=target)
    plt.subplot(3, 2, 3)
    sns.histplot(x=var, data=data, hue=target, common_norm = False)

    plt.subplot(3, 2, 4)
    sns.kdeplot(x=var, data=data, hue=target
            , multiple = 'fill')
    plt.axhline(titanic[target].mean(), color = 'r')

    plt.subplot(3, 2, 5)
    sns.histplot(x=var, data = titanic, bins = 16
             , hue =target, multiple = 'fill')
    plt.axhline(titanic[target].mean(), color = 'r')

    plt.show()
```

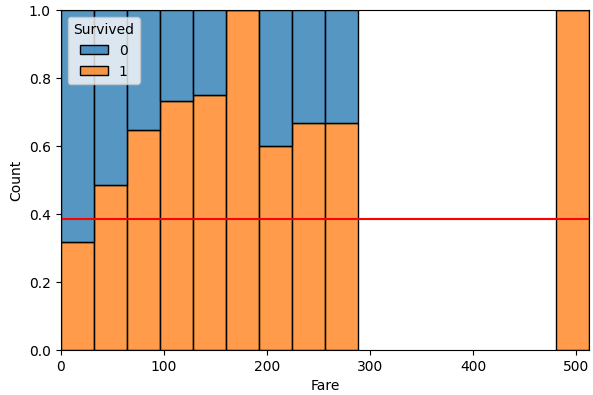
In [18]:
```
ed_1('Fare', 'Survived', titanic)
```











3.복습문제

- 항공기 탑승객의 만족도와 관련 있는 요인을 분석해 봅시다.
- 약 5천명의 탑승객에 대해서 탑승 경험을 바탕으로 데이터셋이 구성되어 있습니다.
  - Target
    - 탑승 만족도(satisfaction) : 만족 = 1, 불만 = 0
  - Feature
    - 성별, 나이, 여행타입, 객실등급, 비행거리, 객실등급, 비행거리, 식음료 만족도, 출발 지연시간

In [26]:
```
path = 'https://raw.githubusercontent.com/DA4BAM/dataset/master/Air_Satisfaction.csv'
cols = ['Gender', 'Age', 'Type of Travel', 'Class', 'Flight Distance', 'Food and drink',
        'Departure Delay in Minutes', 'satisfaction']
```

```
data = pd.read_csv(path, usecols = cols)
data['satisfaction'] = np.where(data['satisfaction'] =='satisfied', 1 ,0)
data.head()
```

Out[26]:

	Gender	Age	Type of Travel	Class	Flight Distance	Food and drink	Departure Delay in Minutes	satisfaction
0	Male	13	Personal Travel	Eco Plus	460	5	25	0
1	Male	25	Business travel	Business	235	1	1	0
2	Female	26	Business travel	Business	1142	5	0	1
3	Female	25	Business travel	Business	562	2	11	0
4	Male	61	Business travel	Business	214	4	0	1

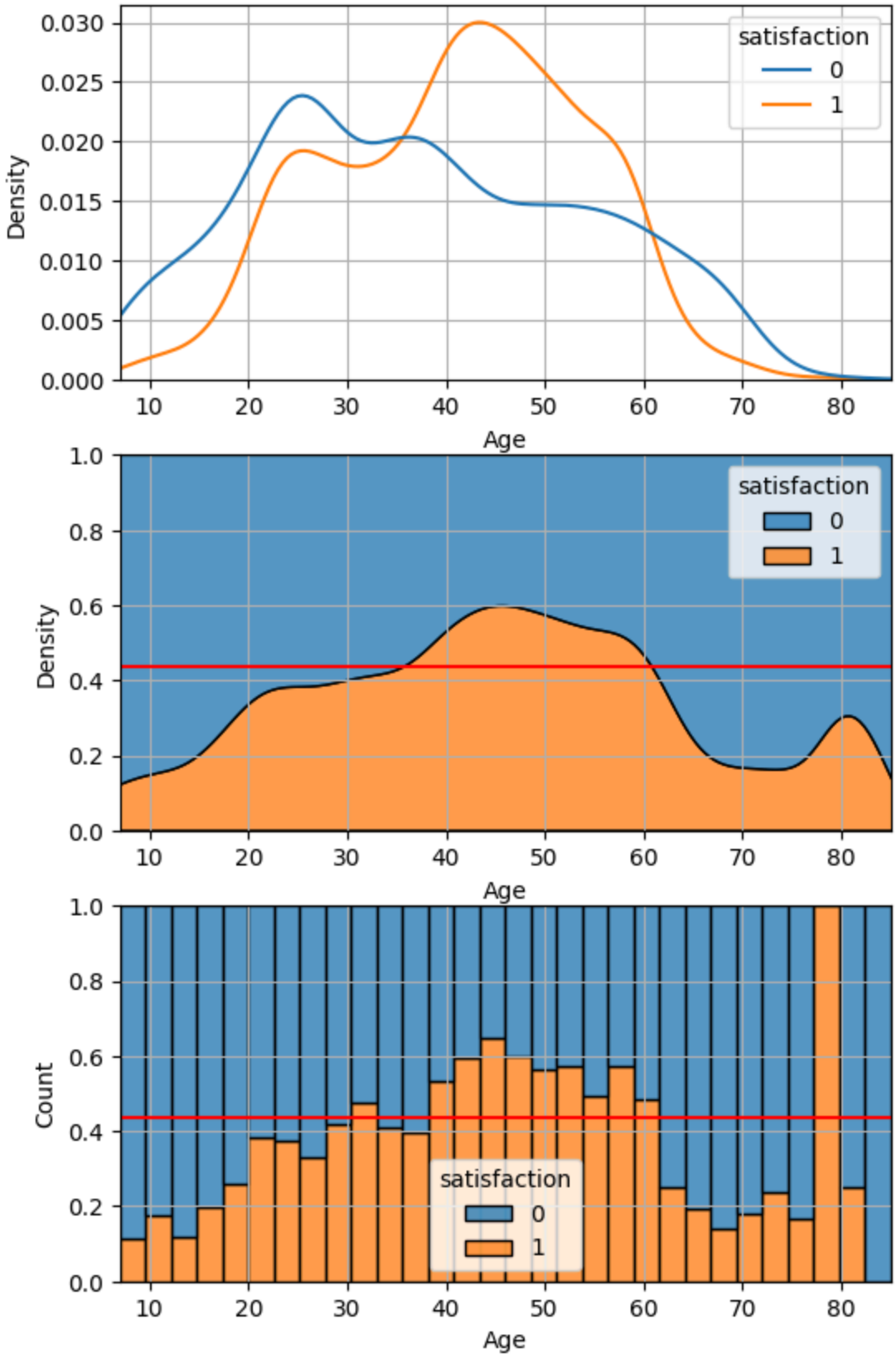
In [27]:
```
target = 'satisfaction'
```

In [32]:
```
def eda_2_nc(feature, target, data) :

    plt.figure(figsize = (6, 10))
    plt.subplot(3,1,1)
    sns.kdeplot(x = feature, data = data, hue = target, common_norm = False)
    plt.xlim(data[feature].min(), data[feature].max())
    plt.grid()

    plt.subplot(3,1,2)
    sns.kdeplot(x = feature, data = data, hue = target, multiple = 'fill')
    plt.axhline(data[target].mean(), color = 'r')
    plt.xlim(data[feature].min(), data[feature].max())
    plt.grid()

    plt.subplot(3,1,3)
    sns.histplot(x = feature, data = data, bins = 30, hue = target, multiple = 'fill')
    plt.axhline(data[target].mean(), color = 'r')
    plt.xlim(data[feature].min(), data[feature].max())
    plt.grid()

plt.show()
```

(1) Age --> Satisfaction

In [33]:
```
feature = 'Age'
```

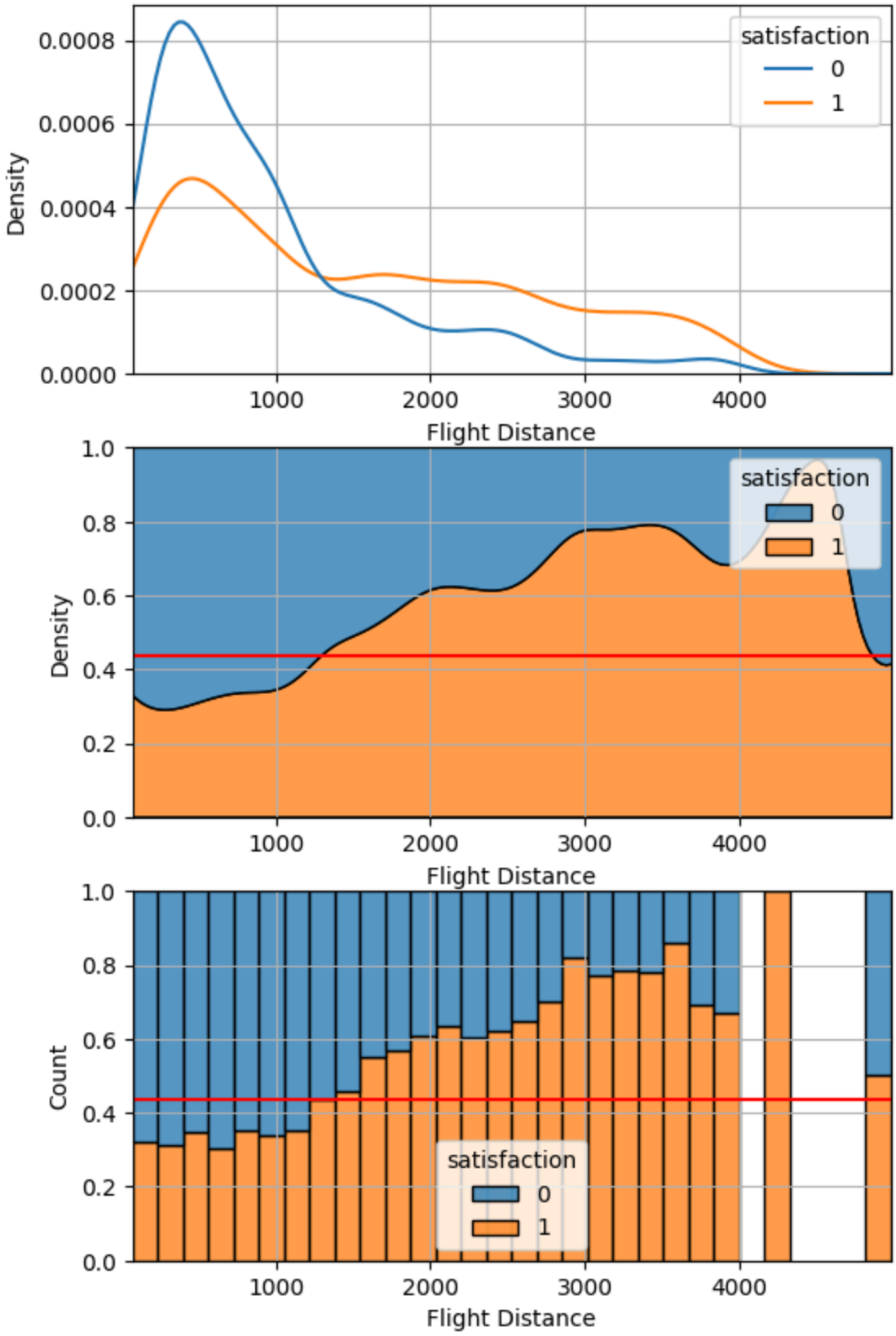
In [34]:
```
eda_2_nc(feature, target, data)
```



(2) Flight Distance --> Satisfaction

In [36]:
```
feature = 'Flight Distance' # 비행거리 # 만족도
```

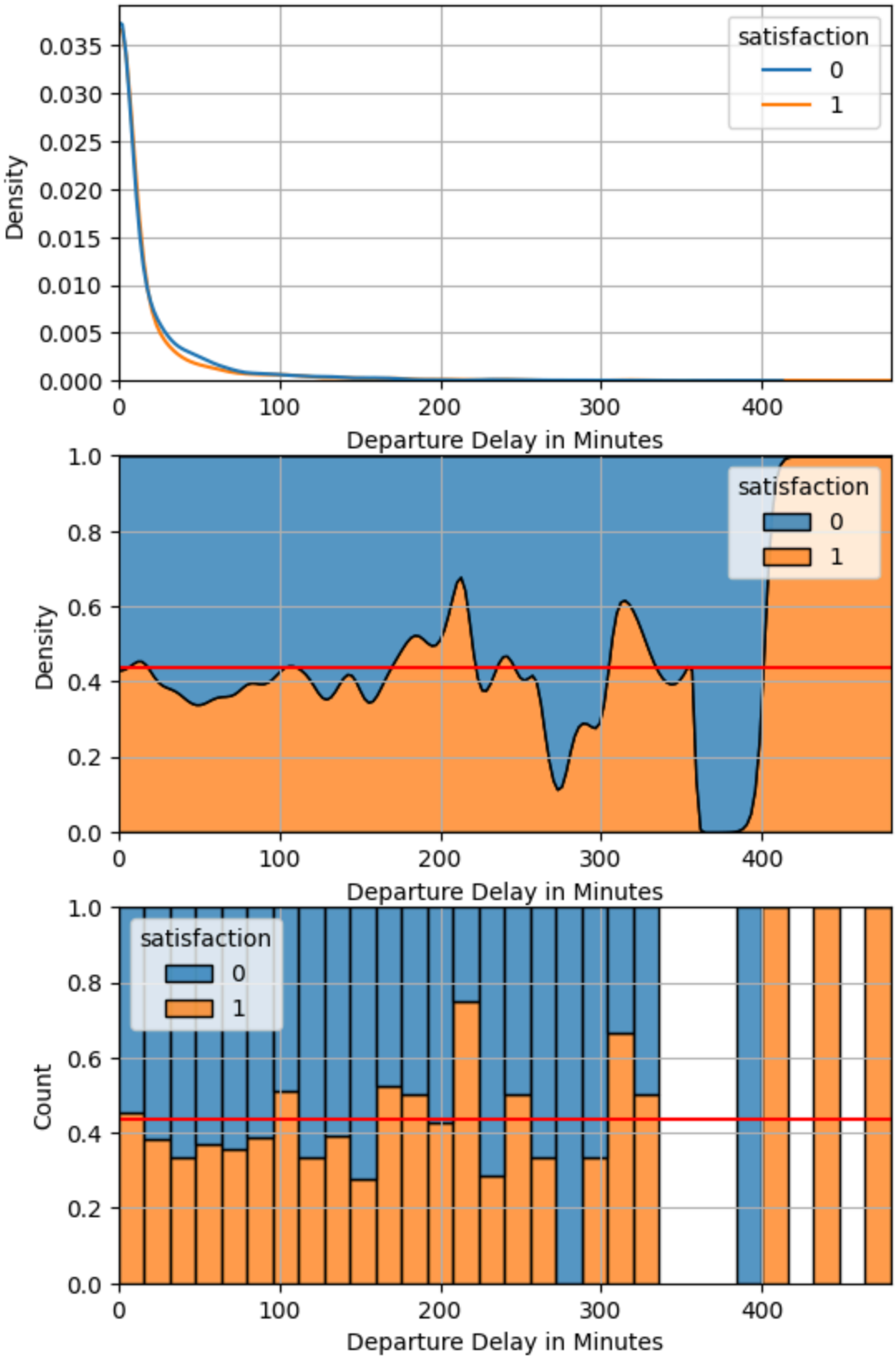
In [37]:
```
eda_2_nc(feature, target, data)
```



(3) Departure Delay in Minutes --> Satisfaction

In [38]:
```
feature = 'Departure Delay in Minutes' # 만족도
```

In [39]:
```
eda_2_nc(feature, target, data)
```



In [40]:
```
# 1) 먼저 교차표 집계- normalize 하면 안 됨
table = pd.crosstab(data[feature], data[target])
print(table)
print('-' * 50)

# 2) 카이제곱검정
k_statistic, pvalue, dof, expected_freq = spst.chi2_contingency(table)
```

```
print(f'카이제곱 통계량 : {k_statistic}')
print(f'P_value : {pvalue}')
print(f'자유도도 : {dof}')        # 적절한 자유도를 가진 모델을 선택하는 것이 중요
print(f'기대빈도 : {expected_freq}') # 기대 빈도가 높을수록, 관측된 데이터와 기대되는 데이터가
```

```
satisfaction                   0     1
Departure Delay in Minutes
0                           1505  1263
1                             88    59
2                             62    53
3                             39    44
4                             54    42
...                          ...   ...
324                            1     0
391                            1     0
412                            0     1
435                            0     1
480                            0     1

[210 rows x 2 columns]
--------------------------------------------------
카이제곱 통계량 : 226.76825794803818
P_value : 0.18997471129223364
자유도도 : 209
기대빈도 : [[1.56392e+03 1.20408e+03]
 [8.30550e+01 6.39450e+01]
 [6.49750e+01 5.00250e+01]
 [4.68950e+01 3.61050e+01]
 [5.42400e+01 4.17600e+01]
 [4.46350e+01 3.43650e+01]
 [4.12450e+01 3.17550e+01]
 [3.61600e+01 2.78400e+01]
 [3.39000e+01 2.61000e+01]
 [3.72900e+01 2.87100e+01]
 [3.27700e+01 2.52300e+01]
 [3.39000e+01 2.61000e+01]
 [2.14700e+01 1.65300e+01]
 [2.31650e+01 1.78350e+01]
 [2.88150e+01 2.21850e+01]
 [2.37300e+01 1.82700e+01]
 [2.31650e+01 1.78350e+01]
 [1.86450e+01 1.43550e+01]
 [2.26000e+01 1.74000e+01]
 [1.86450e+01 1.43550e+01]
 [2.03400e+01 1.56600e+01]
 [1.52550e+01 1.17450e+01]
 [1.63850e+01 1.26150e+01]
 [1.69500e+01 1.30500e+01]
 [1.41250e+01 1.08750e+01]
 [1.63850e+01 1.26150e+01]
 [1.35600e+01 1.04400e+01]
 [1.69500e+01 1.30500e+01]
 [1.97750e+01 1.52250e+01]
 [1.29950e+01 1.00050e+01]
 [1.01700e+01 7.83000e+00]
 [1.07350e+01 8.26500e+00]
 [7.91000e+00 6.09000e+00]
 [6.21500e+00 4.78500e+00]
 [6.21500e+00 4.78500e+00]
 [1.01700e+01 7.83000e+00]
 [7.34500e+00 5.65500e+00]
 [9.04000e+00 6.96000e+00]
 [6.78000e+00 5.22000e+00]
 [1.07350e+01 8.26500e+00]
 [6.78000e+00 5.22000e+00]
```

```
[5.65000e+00 4.35000e+00]
[9.60500e+00 7.39500e+00]
[7.91000e+00 6.09000e+00]
[8.47500e+00 6.52500e+00]
[6.21500e+00 4.78500e+00]
[1.01700e+01 7.83000e+00]
[2.26000e+00 1.74000e+00]
[7.91000e+00 6.09000e+00]
[5.08500e+00 3.91500e+00]
[3.39000e+00 2.61000e+00]
[6.78000e+00 5.22000e+00]
[5.65000e+00 4.35000e+00]
[7.34500e+00 5.65500e+00]
[5.08500e+00 3.91500e+00]
[6.78000e+00 5.22000e+00]
[5.65000e+00 4.35000e+00]
[3.39000e+00 2.61000e+00]
[1.69500e+00 1.30500e+00]
[4.52000e+00 3.48000e+00]
[6.78000e+00 5.22000e+00]
[2.82500e+00 2.17500e+00]
[4.52000e+00 3.48000e+00]
[3.39000e+00 2.61000e+00]
[4.52000e+00 3.48000e+00]
[2.82500e+00 2.17500e+00]
[3.95500e+00 3.04500e+00]
[5.65000e+00 4.35000e+00]
[5.65000e-01 4.35000e-01]
[2.26000e+00 1.74000e+00]
[2.82500e+00 2.17500e+00]
[2.82500e+00 2.17500e+00]
[3.95500e+00 3.04500e+00]
[2.82500e+00 2.17500e+00]
[2.26000e+00 1.74000e+00]
[5.65000e-01 4.35000e-01]
[2.82500e+00 2.17500e+00]
[1.69500e+00 1.30500e+00]
[1.13000e+00 8.70000e-01]
[5.65000e-01 4.35000e-01]
[4.52000e+00 3.48000e+00]
[1.69500e+00 1.30500e+00]
[1.13000e+00 8.70000e-01]
[1.69500e+00 1.30500e+00]
[4.52000e+00 3.48000e+00]
[2.82500e+00 2.17500e+00]
[5.65000e-01 4.35000e-01]
[1.13000e+00 8.70000e-01]
[1.13000e+00 8.70000e-01]
[2.26000e+00 1.74000e+00]
[1.69500e+00 1.30500e+00]
[2.82500e+00 2.17500e+00]
[1.13000e+00 8.70000e-01]
[2.26000e+00 1.74000e+00]
[2.82500e+00 2.17500e+00]
[2.82500e+00 2.17500e+00]
[1.13000e+00 8.70000e-01]
[1.13000e+00 8.70000e-01]
[1.13000e+00 8.70000e-01]
[3.39000e+00 2.61000e+00]
[1.69500e+00 1.30500e+00]
```

```
[1.13000e+00 8.70000e-01]
[1.13000e+00 8.70000e-01]
[5.65000e-01 4.35000e-01]
[1.69500e+00 1.30500e+00]
[3.39000e+00 2.61000e+00]
[2.82500e+00 2.17500e+00]
[1.13000e+00 8.70000e-01]
[5.65000e-01 4.35000e-01]
[1.13000e+00 8.70000e-01]
[1.13000e+00 8.70000e-01]
[1.13000e+00 8.70000e-01]
[2.26000e+00 1.74000e+00]
[3.39000e+00 2.61000e+00]
[1.13000e+00 8.70000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[1.69500e+00 1.30500e+00]
[1.69500e+00 1.30500e+00]
[1.13000e+00 8.70000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[1.13000e+00 8.70000e-01]
[1.13000e+00 8.70000e-01]
[2.82500e+00 2.17500e+00]
[1.13000e+00 8.70000e-01]
[1.13000e+00 8.70000e-01]
[1.13000e+00 8.70000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[1.13000e+00 8.70000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[1.69500e+00 1.30500e+00]
[1.13000e+00 8.70000e-01]
[1.13000e+00 8.70000e-01]
[1.13000e+00 8.70000e-01]
[1.13000e+00 8.70000e-01]
[5.65000e-01 4.35000e-01]
[1.69500e+00 1.30500e+00]
[1.13000e+00 8.70000e-01]
[5.65000e-01 4.35000e-01]
[1.13000e+00 8.70000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[1.13000e+00 8.70000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[1.69500e+00 1.30500e+00]
[1.69500e+00 1.30500e+00]
[1.13000e+00 8.70000e-01]
[1.13000e+00 8.70000e-01]
[5.65000e-01 4.35000e-01]
[1.13000e+00 8.70000e-01]
[5.65000e-01 4.35000e-01]
[1.69500e+00 1.30500e+00]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
```

```
[1.69500e+00 1.30500e+00]
[5.65000e-01 4.35000e-01]
[1.13000e+00 8.70000e-01]
[1.13000e+00 8.70000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[1.13000e+00 8.70000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[1.13000e+00 8.70000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[1.13000e+00 8.70000e-01]
[1.13000e+00 8.70000e-01]
[1.13000e+00 8.70000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]
[5.65000e-01 4.35000e-01]]
```

In [ ]: